

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

кафедра Компьютерных технологий

«Допущен к защите»
Заведующий кафедрой _____

(Ф.И.О., ученая степень, звание)

« _____ » _____ 20__ г.
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Сайт дистанционного обучения
университета АУЭС

Специальность Вычислительная техника и программное обеспечение

Выполнил (а) Мухтаров С.Д.
(Фамилия и инициалы) группа _____

Научный руководитель Ержан А.А. д.ф.н. (Р.Ф.)
(Фамилия и инициалы, ученая степень, звание)

Консультанты:

по экономической части:

Ермеева З.Д. с.т.н. преподаватель
(Фамилия и инициалы, ученая степень, звание)
Ермеева « 15 » 05 20/14 г.
(подпись)

по безопасности жизнедеятельности:

Бригидова А.Г. д.ф.н. преподаватель
(Фамилия и инициалы, ученая степень, звание)
Бригидова « 04 » 05 20/14 г.
(подпись)

по применению вычислительной техники:

Ержан А.А. д.ф.н. (Р.Ф.)
(Фамилия и инициалы, ученая степень, звание)
Ержан « 27 » 05 20/14 г.
(подпись)

(Фамилия и инициалы, ученая степень, звание)

« _____ » _____ 20__ г.

(подпись)

Нормоконтролер: Тусупов Д.М.
(Фамилия и инициалы, ученая степень, звание)

Тусупов « 28 » мая 20/14 г.
(подпись)

Рецензент:

(Фамилия и инициалы, ученая степень, звание)

« _____ » _____ 20__ г.

(подпись)

Алматы 2014 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Факультет Информационных технологий
Специальность Вычислительная техника и программное обеспечение
Кафедра Компьютерных технологий

ЗАДАНИЕ

на выполнение дипломного проекта

Студент Мухтаров Султан Дулатович
(фамилия, имя, отчество)

Тема проекта Сайт дистанционного обучения
университета АУЭС

утверждена приказом ректора № 115 от «24» сентября 2013 г.

Срок сдачи законченной работы «29» мая 2014 г.

Исходные данные к проекту требуемые параметры результатов проектирования (исследования) и исходные данные объекта

Разработка сайта дистанционного обучения
для университета АУЭС

Перечень подлежащих разработке дипломного проекта вопросов или краткое содержание дипломного проекта:

Современные технологии разработки сайтов, последо-
вательность создания сайта дистанционного обучения,
история PHP, язык гипертекстовой разметки HTML,
CSS-каскадное таблицы стилей, объект-ориентированной
язык JavaScript, Основы XML, Система управления
базой данных MySQL

Перечень графического материала (с точным указанием обязательных чертежей)

Экономическое приложение, указание уточненного объема и группа сложности, дополнительные коэффициенты сложности, поправочные коэффициенты, диаграмма себестоимости

Рекомендуемая основная литература

- 1 Рэбин Никсон. Создаем динамические веб-сайты с помощью PHP, MySQL и JavaScript
- 2 Линн Бейли, Майкл Моррисон. Изучаем PHP и MySQL
- 3 Веллинг Л., Томсон Л. Разработка веб-приложений с помощью PHP и MySQL
- 4 Суэринг С., Конверс С., Парк Д. PHP и MySQL. Библия программиста.

Консультанты по проекту с указанием относящихся к ним разделов

Раздел	Консультант	Сроки	Подпись
Ассессора	Брессева З. Д.	11.04 - 13.05.14	Брессева
БЖД	Трихачев И. Г.	11.04 - 04.05.14	Трихачев
Основная часть	Сретенск Л. А.	11.04 - 20.05.14	Сретенск
Нормоконтроль	Тусуров Д. М.	28.05.14 г.	Тусуров

Г Р А Ф И К
подготовки дипломного проекта

№ п/п	Наименование разделов, перечень разрабатываемых вопросов	Сроки представления руководителю	Примечание
1	Сбор материала и обработка информации	1.04.2014	
2	Формирование задач и разработка структуры СДО	10.04.2014	
3	Разработка СДО	13.04.2014	
4	Корректировка и отладка СДО	30.04.2014	
5	Расчет экономической части	10.05.2014	
6	Работа над разделом БЖД	16.05.2014	
7	Подготовка графического материала	18.05.2014	
8	Работа над пояснительной запиской	20.05.2014	

Дата выдачи задания « 3 » марта 2014 г.

Заведующий кафедрой _____
(подпись) (Фамилия и инициалы)

Руководитель _____
(подпись) (Фамилия и инициалы)

Задание принял к исполнению студент _____
(подпись) (Фамилия и инициалы)

Аннотация

Тема дипломной работы "Сайт дистанционного обучения университета АУЭС". "Алматинский университет энергетики и связи" послужил как объект исследования в дипломной работе, а именно процесс обучения. Предметом исследования является введение дистанционного обучения, как альтернатива заочного образования.

Выпускная работа состоит из пяти разделов. Во введении раскрыта главная цель моей работы, а также актуальность дистанционного обучения в Казахстане. Разделы описывают от структуры разработки до созданного полнофункционального сайта. В заключении представлены основные выводы и предложения по данной проделанной работе.

Аңдатпа

Дипломдық жұмыстың тақырыбы "АЭЖБУ университетінің дистанциондық оқу сайты". Алматы энергетика және байланыс университетінің оқу үдірісі жобаның нысаны болып табылады. Зерттеу пәні болып, дистанциондық оқуды осы кафедраға еңгізу.

Дипломдық жоба бес тараудан тұрады. Кіріспе сайтың жобаланған негізгі ойын білдіреді және оның осы аұқыттағы маңыздылығы. Бес тарауда жобаның бас тапқы кезеңінен толық сайттың жасалғанына дейін көрсетілген. Ең соңында жұмыстың қортындысы көрсетілген.

Annotation

Thesis "Site of distance learning of the university AUPET ". "Almaty University of Power Engineering and Telecommunication" has served as an object of study in the thesis work, precisely the learning process. The subject of research is the introduction of distance learning as an alternative distance education.

Final work consists of five sections. In the introduction, the main purpose is revealed my work , as well as the relevance of distance education in Kazakhstan. Sections describe from the structure of the design to created a full website. In conclusion presented main suggestions and conclusions on the done work.

Содержание

Введение.....	14
1 Аналитическая часть.....	15
1.1 Современные технологии разработки сайтов.....	15
1.1.1 Особенности современных Web–редакторов.....	16
1.1.2 Классификация интернет–систем.....	18
1.1.3 Интернет сегодня.....	21
1.2 Последовательности создания сайта бесплатного дистанционного обучения.....	26
1.2.1 Постановка задачи.....	26
1.2.2 Назначение и область применения сайта.....	27
1.2.3 Анализ предметной области.....	28
1.2.4 Основные типы организационных структур дистанционного образования.....	31
1.2.5 Электронный учебник как средство дистанционного обучения.....	32
1.2.6 Дистанционное обучение в Казахстане.....	35
2 Программы используемые при разработке данного сайта.....	34
2.1 Краткая история PHP.....	34
2.2 Язык гипертекстовой разметки HTML.....	36
2.3 CSS – каскадные таблицы стилей.....	39
2.4 Объектно–ориентированный язык JavaScript.....	40
2.5 Основы XML.....	41
2.6 Система управления базами данных MySQL.....	43
3 Создание базы данных и приложения к ней.....	47
3.1 СУБД.....	47
3.2 Разграничение доступа.....	48
3.3 Создание базы данных на Wampserver.....	52
3.4 Описание программы.....	54
4 Техничко–экономическое обоснование.....	65
4.1. Расчет затрат на разработку информационных технологий.....	65
4.2 Расчет цены программного продукта.....	74
4.3 Расчет эксплуатационных затрат при использовании информационных технологий.....	75
4.4 Расчет основных показателей экономической эффективности.....	77
Вывод.....	78
5 Безопасность жизнедеятельности.....	79
5.1 Анализ условий труда.....	79
5.2 Анализ оборудования.....	81
5.3 Расчет естественного освещения.....	81
5.4 Расчет системы кондиционирования офиса.....	85
Вывод.....	89
Заключение.....	90

Список литературы	91
Приложение А	92
Приложение Б	140
Приложение В.....	142
Приложение Г	143
Приложение Д.....	143
Приложение Ж.....	143
Приложение З	144

Введение

В современном мире дистанционное обучение представляет получения образования нового поколения. Это организация процесса образования таким образом, при котором преподаватель предоставляет полную информацию программы обучения курса, для самостоятельного изучения студентам. Данная технология обучения на первый взгляд отделяет преподавателя от студента, но новые технология позволят в любое время осуществлять диалог между собой в онлайн режиме. Дистанционно могут обучаться не только выпускники средних школ и колледжей, но и любой другой желающий. Например жители отдаленных регионов, не имеющих доступ к профессиональной подготовке.

Во многих странах с середины 70–х годов данная форма обучения стала развиваться, в нынешнее время дистанционное обучение имеет оригинальное организацию процесса получения образования. В Казахстане дистанционное обучение на стадии развития, только последние 10 лет данный метод начали частично использовать.

Дистанционное обучения это не только новый метод образования, но финансово экономическая выгода для университета, ведь нет прямых затрат для содержания студентов в общежитии, предоставлении аудиторий и прочие затраты. Затраты необходимы только на оснащения сайта, а также оплата труда преподавателей.

Также необходимо принять во внимание, что система дистанционного образования, должна быть максимально приближена к реальной системе образования. Процесс обучения на данном сайте состоит из двух частей: изучение лекций и прохождение тестов по пройденным разделам.

Данную тему выбрал в качестве дипломного проекта, так как дистанционное образование актуально в современном обществе.

1 Аналитическая часть

1.1 Современные технологии разработки сайтов

Современные интернет–разработки предусматривает использование таких технологий как: PHP, JavaScript, Ajax, «Web 2.0». Данные технологии прежде всего дают возможность, удобство работы для пользователя. Это означает, что сайт, разработанный с данной концепцией, должен делать все , чтобы конечному пользователю было понятно, комфортно, интересно находиться на таком сайте.

Веб–сервис – предназначены для удобного доступа данных с помощью обычного интернет–браузера, пользователю не нужно задумываться ни о каких дополнительных программах на компьютере, в данной работе применялся сервер WampServer.

Ajax – это методика применения технологий JavaScript и XML, что позволяет загружать нужные данные не перегружая страницу, что значительно ускоряет удобство и скорость серфинга по сайту. RSS – это технология, основанная на XML, что позволяет пользователям с помощью специальной программы–агрегатора просматривать новости с различных сайтов как единый поток – строка новостей.

Теги – это метки, использование которых позволяет более удобно идентифицировать и тематически сортировать контент (статьи, рисунки, мультимедиа, файлы и т.п.).

На сегодня широко используются такие разновидности (CMS):

- Web content management Systems для управления веб–сайтами (например, энциклопедиями, подобными Википедии, онлайн–изданиями, блогами, форумами, корпоративными или личными веб–страницами и др);
- транзакционные СКО для обеспечения транзакций в электронной торговле;
- интегрированные СКО для работы с документацией на предприятиях;
- электронные библиотеки (Digital Asset Management) для обеспечения цикла жизни файлов электронных медиа (видео, графические, презентации и т.д.);
- системы для обеспечения цикла жизни документации (руководства, справочники, описания);
- образовательные СКО – системы для организации Интернет–курсов и соответствующего цикла жизни документации;
- корпоративные СКВ (Enterprise content management systems) с разноплановым приспособлением для нужд предпринимательской деятельности. Поддерживают цикл жизни внутренней и внешней документации;

– платформенные СКВ (Platform Content Management Systems) поддерживают автоматизацию работы с компьютерными файлами, папками, приложениями в определенной программной среде.

1.1.1 Особенности современных Web-редакторов

Для создания сайта самое первое что необходимо сделать это выбрать web-редактора. Выбор зависит от разработчика, а также от поставленных целей по созданию данного сайта. За все время создания своего сайта я выделил два современных и достаточно удобных web-редактора: Dreamweaver, Sublime Text.

Dreamweaver – WYSIWYG HTML-редактор компании Adobe. Изначально разработан и поддерживался компанией Macromedia, вплоть до 8-й версии (2005 год). Следующие версии, начиная с Dreamweaver CS3(2007), выпускает Adobe. К основным достоинствам относится:

- Цветовая подсветка кода, которая позволяет легко отделить HTML от PHP и Java Script, сразу же найти область комментариев, увидеть ошибку в написании кода представлена на рисунке 1.1.

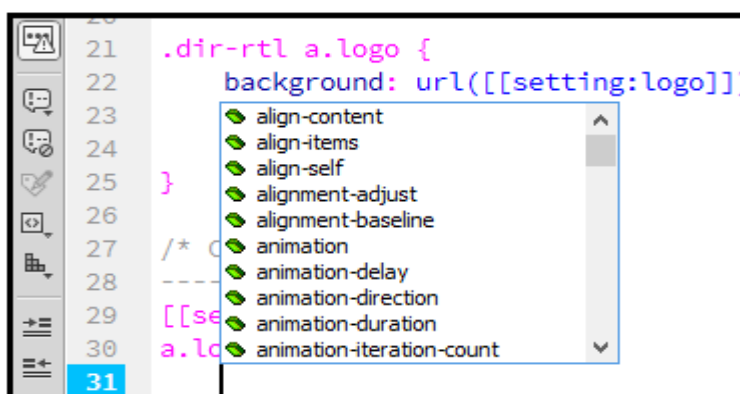


Рисунок 1.1 – Цветовая подсветка

- Автоматическая подстановка кода, что позволяет не тратить время на полное написание каких либо тегов или стандартных функций – стоит просто ввести первые несколько букв этого тега или функции и Dreamweaver сам догадается, что Вы имели ввиду – все что остается – просто нажать Enter. При работе над большими проектами с более 1000 строчек кода в странице это существенно и в разы экономит время. К тому же Вы уверены, что не ошибетесь при написании кода, и не придется долго выискивать – где же допущена ошибка. Для новичков – это прекрасный учебник HTML, CSS, PHP и других популярных языков программирования.

- Автоматическая подстановка классов CSS из подгружаемого файла стилей. Эта функция Dreamweaver позволяет быстро и эффективно работать с классами подгружаемого файла стилей. Даже если над файлом стилей работает несколько человек (к примеру через систему контроля версий CVS или SVN),

Вы всегда будете видеть все доступные и вновь созданные классы – что не даст запутаться или забыть о том, что уже написано, а что еще нет.

- На текущий момент рендеринг визуального редактора Adobe Dreamweaver CS3 еще далек до идеала. Сомневаюсь, что это изменится в ближайших версиях, но главное не в рендеринге. На мой взгляд, изюминкой Dreamweaver является подсветка участка кода при выделении в визуальном редакторе какого либо элемента. Для того, чтобы найти, к примеру, ячейку таблицы со сложной вложенностью Вам понадобится кликнуть по этой ячейке в визуальном редакторе, и необходимый Вам код тут же подсветится в текстовом редакторе представлено на рисунке 1.2. Работая в режиме двух окон (текстовое визуальный редактор) у Вас не возникнет трудностей чтения кода и поиска в нем нужного участка. Таким образом, документ любой сложности не вызывает проблем с редактированием и повышает производительность в разы.

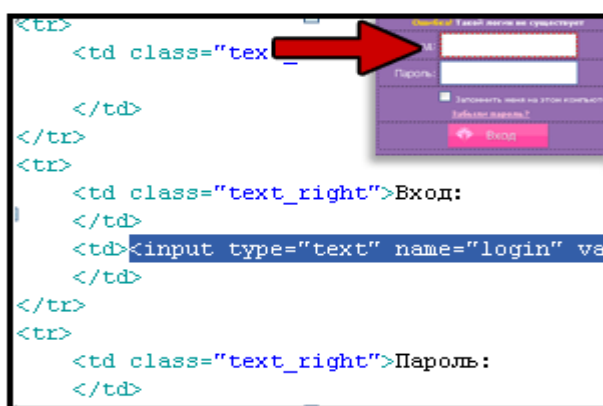


Рисунок 1.2 – Автовод

- У Dreamweaver есть множество мелких, но приятных функций для облегчения работы. Он полностью настраиваемый. Есть возможность создавать свои собственные объекты и команды, изменять "горячие" клавиши. При желании можно расширять возможности Dreamweaver новыми свойствами с помощью языка JavaScript представлено на рисунке 1.3. К примеру, одной из интересных и редко используемых но крайне полезных функций Dreamweaver является форматирование кода. Эта функция находится в разделе command и называется – apply source formatting. С помощью нее Вы можете привести в красивый и удобочитаемый вид даже самый запутанный код

Можно еще долго описывать все прелести работы с Dreamweaver, а также особенности его настройки и борьбы с глюками (к сожалению, ничто не идеально), но, на мой взгляд, более удобного редактора для веб-верстки найти крайне сложно.

```
477 <!-- авторизация --><div class="f_block" style="width: 100%; text-align: center;">
478 </div>
479 <div class="block_top"><img alt="top bar" /></div>
480 <div class="block_bottom"><img alt="bottom bar" /></div>
481 <div class="block_in">
482 <table width="100%" cellpadding="5" cellspacing="0">
483 <tr>
484 <td style="width: 50%; vertical-align: top;">
485 <input type="text" name="login" value="" class="input_blue" style="width: 95%; height: 20px; margin-bottom: 5px;" />
486 <input type="password" value="" class="input_blue" style="width: 95%; height: 20px; margin-bottom: 5px;" />
487 <input type="checkbox" name="remember" value="1" /> Запомнить пароль?
488 <input type="button" value="Вход" class="button" style="margin-top: 5px; width: 40%; float: right;" />
489 </td>
490 <td style="width: 50%; vertical-align: top; text-align: center;">
491 <div style="margin-bottom: 10px;">
492 <input type="button" value="Регистрация" class="button" style="width: 40%; margin: 0 auto;" />
493 </div>
494 <div style="margin-bottom: 10px;">
495 <input type="button" value="Забыли пароль?" class="button" style="width: 40%; margin: 0 auto;" />
496 </div>
497 </td>
498 </tr>
499 </table>
500 </div>
501 </div>
```

Рисунок 1.3 – Создание горячих клавиш на Javascript

Sublime Text – быстрый кроссплатформенный редактор исходных текстов программ. Поддерживает плагины на языке программирования Python. Sublime Text не является свободным или открытым программным обеспечением, однако, некоторые его плагины распространяются по свободной лицензии, а также разрабатываются и поддерживаются сообществом разработчиков. Дополнительно в Sublime Text реализована функция автосохранения, помогающая пользователям не потерять проделанную работу. Настраиваемые комбинации клавиш и инструмент навигации позволяют пользователям назначать свои комбинации клавиш для меню и панелей инструментов. Возможность поиска по мере набора используется для поиска в документе и осуществляет поиск по мере набора поискового запроса. Функция проверки синтаксиса работает подобным же образом, проверяя корректность прямо во время ввода. Sublime Text предлагает в дополнение к вышеперечисленному возможности автоматизации с помощью макросов и повтора последних действий.

1.1.2 Классификация интернет–систем

Цель любой информационной системы – обработка данных об объектах реального мира.

Интернет продукты можно классифицировать по ниже следующим пунктам.

По правам на доступность сервисов:

- Открытые – все сервисы полностью доступны для любых посетителей и пользователей.
- Полуоткрытые – для доступа необходимо зарегистрироваться (обычно бесплатно).
- Закрытые – полностью закрытые служебные сайты организаций (в том числе корпоративные сайты), личные сайты частных лиц. Такие сайты доступны для узкого круга пользователей. Доступ новым пользователям обычно даётся через приглашения (инвайты).

По физическому расположению:

- Общедоступные сайты сети Интернет.

– Локальные сайты – доступны только в пределах локальной сети.

Информационные ресурсы:

– Тематический сайт – сайт, предоставляющий специфическую узкоспециализированную информацию по какой-либо теме.

– Тематический портал – это большой веб-ресурс, который предоставляет исчерпывающую информацию по определённой тематике. Порталы похожи на тематические сайты, но дополнительно содержат средства взаимодействия с пользователями и позволяют пользователям общаться в рамках портала (форумы, чаты) – это среда существования пользователя.

Интернет-представительства владельцев бизнеса (торговля и услуги, не всегда связанные напрямую с Интернетом):

– Сайт-визитка – содержит самые общие данные о владельце сайта (организация или индивидуальный предприниматель). Вид деятельности, история, прайс-лист, контактные данные, реквизиты, схема проезда. Специалисты размещают своё резюме, то есть подробная визитная карточка.

– Представительский сайт – так иногда называют сайт-визитку с расширенной функциональностью: подробное описание услуг, портфолио, отзывы, форма обратной связи и т.д.

– Корпоративный сайт – содержит полную информацию о компании/владельце, услугах/продукции, событиях в жизни компании. Отличается от сайта-визитки и представительского сайта полнотой представленной информации, зачастую содержит различные функциональные инструменты для работы с содержимым (поиск и фильтры, календари событий, фотогалереи, блоги, форумы). Может быть интегрирован с внутренними информационными системами компании-владельца (КИС, CRM, бухгалтерскими системами). Может содержать закрытые разделы для тех или иных групп пользователей – сотрудников, дилеров, контрагентов и пр.

– Каталог продукции – в каталоге присутствует подробное описание товаров/услуг, сертификаты, технические и потребительские данные, отзывы экспертов и т.д. На таких сайтах размещается информация о товарах/услугах, которую невозможно поместить в прайс-лист.

– Интернет-магазин – сайт с каталогом продукции, с помощью которого клиент может заказать нужные ему товары. Используются различные системы расчётов: от пересылки товаров наложенным платежом или автоматической пересылки счета по факсу до расчётов с помощью пластиковых карт.

– Промо-сайт – сайт о конкретной торговой марке или продукте, на таких сайтах размещается исчерпывающая информация о бренде, различных рекламных акциях (конкурсы, викторины, игры и т. п.).

– Сайт-квест – Интернет-ресурс, на котором организовано соревнование по разгадыванию последовательности взаимосвязанных логических загадок.

– Веб-сервис – сайт, созданный для выполнения каких либо задач либо предоставления услуг в рамках глобальной сети.

- Доска объявлений.
- Каталог сайтов– например, Open Directory Project.
- Поисковые сервисы – например, Yahoo, Google.
- Почтовый сервис.
- Форумы.
- Блоговый сервис.
- Файлообменный пиринговый сервис – например, Bittorrent.
- Датахостинговый (хранение данных) сервис – например, Skydrive.
- Датаэдиторинговый (редактирование данных) сервис, например, GoogleDocs.
- Фотохостинг– например, Picnik, ImageShack, Panoramio, Photobucket.
- Хранение видео – например, YouTube, Dailymotion.
- Социальные Медиа– например, Buzz.
- Комбинированные веб–сервисы (Социальные сети) – например, Facebook, Twitter, ВКонтакте.
- Комбинированные веб–сервисы (Специализированные социальные сети) – например, MySpace, Flickr, Spring Me.

Немного подробнее о досках объявлений. Доска объявлений – сайт предоставляющий площадку для размещения объявления. Это может быть сайт с конкретной узконаправленной тематикой, многопрофильный, платным/бесплатным, отдельным сайтом некоторого конкретного города либо региона, может быть модерлируемый либо администрируемый и т.д.

По технологии отображения:

- Статические – состоящие из статичных html (htm) страниц составляющих единое целое. Пользователю выдаются файлы в том виде, в котором они хранятся на сервере.
- Динамические – состоящие из динамичных html (htm, dhtml) страниц–шаблонов, информации, скриптов и прочего в виде отдельных файлов. Содержимое генерируется по запросу специальными скриптами (программами) на основе других данных из любого источника

По типам макетов:

- Фиксированной ширины (англ.rigidfixed) – размеры элементов страницы имеют фиксированное значение, независящее от разрешения, размера, соотношения сторон экрана монитора и размеров окна обозревателя, задаётся в абсолютных значениях– PX (пиксели).
- Резиновый макет (англ.adaptablefluid) – размеры несущих элементов, значения ширины, задаются относительным значением–% (проценты), страницы отображаются во весь экран монитора по ширине . По сути данный тип макета является частным случаем фиксированного типа макета.

Динамично эластичный (англ.dynamicallyexpandableelastic) – размеры большинства элементов, задаются относительными значениями– EM и% . Все относительные пропорции размеров элементов всегда остаются неизменными, независимо от разрешения, размера, соотношения сторон экрана монитора,

размеров окна и масштаба окна обозревателя, и всегда постоянны относительно окна обозревателя.

1.1.3 Интернет сегодня

В настоящее время сеть Интернет объединяет несколько миллионов компьютеров пользователей по всему миру. Все эти компьютеры можно условно разделить на два основных класса: клиенты и серверы.

Клиенты – это компьютеры, подключенные к сети через поставщика услуг Интернета (Internet Service Provider), используют ее, как правило, для поиска и получения информации, работы с электронной почтой. Программное обеспечение клиентов может включать:

- Браузер или универсальный клиент – это программа, которая обеспечивает загрузку и отображение Web–страниц. Кроме того, современные версии этого ПО позволяют выполнять небольшие клиентские приложения (скрипты и апплеты), расширяющие возможности обычных Web–страниц, а также осуществлять путешествия в виртуальных мирах.

- Почтовые приложения, работающие с электронной почтой. Другие мультимедийные средства, как например, воспроизведение звуковых и видеороликов. Клиентские модули систем групповой работы, которые обеспечивают создание единого информационного поля над распределенными клиентами и информационными системами.

Серверы – это мощные компьютеры, которые имеют высокопропускной канал для связи с Интернетом и соответствующее программное обеспечение, поддерживающее работу тех или иных служб, и, как правило, предоставляют информацию клиенту. В зависимости от типа сервисов, которые предоставляют серверы, они делятся на:

- Web–серверы – работают с протоколом высокого уровня HTTP, и в ответ на запрос клиента формируют по заранее определенному алгоритму Web–страницу. Причем, эта страница может быть либо создана заранее, и тогда ее называют статической, либо генерироваться непосредственно в ответ на запрос – это динамически созданная страница.

- Почтовые серверы – как правило, поддерживают протокол POP (Post Office Protocol) и SMTP (Simple Mail Transfer Protocol). Служат, как следует из названия, для работы с электронной почтой и имеют различное назначение. Посредством протокола POP обеспечивается загрузка писем с ящика клиента на почтовом сервере, которым он пользуется, в его программу–клиент, а также некоторые другие возможности для клиент–серверного взаимодействия (см. приложение 1). Для отправки почтовых сообщений от клиента на сервер, а также передачи их между серверами, на пути их следования от отправителя к серверу получателя используется протокол SMTP.

- FTP–серверы – работают с протоколом FTP и служат для организации доступа к файловым архивам сервера.

– Gopher–серверы – работают по протоколу Gopher, разработанному в Университете Миннесоты (University of Minnesota), в настоящее время их можно считать устаревшими.

Это серверы, с которыми можно взаимодействовать, используя соответствующее программное обеспечение (например, программу почтового клиента или браузер). Однако есть специализированные серверы, которые тоже могут быть доступны посредством Интернета, но, они, как правило, предназначены для обслуживания других серверов (например, некоторые серверы, обслуживающие базы данных или обеспечивающие проверку кодов кредитных карт) или для работы со специализированным программным обеспечением (RealAudio или NetShow). Многие фирмы предлагают специализированное серверное программное обеспечение, также называемое сервером (или, правильнее, информационным сервером). Это ПО является дополнением к одному из вышеперечисленных серверов и выполняет функции обслуживания электронных конференций, торговли, рассылки новостей и т.д.

Информационные серверы Информационные серверы устанавливаются на специализированные операционные системы. Как правило, это различные версии Unix, Windows NT Server или Novel NetWare. Поскольку основная цель книги – описать создание информационных систем с помощью среды Delphi, а код, созданный в Delphi, гарантированно работает корректно только на Windows–платформах, то в дальнейшем мы будем рассматривать лишь эту операционную систему. Существуют программы–эмуляторы, позволяющие запускать Win32–приложения на различных не Windows платформах, однако эти "переходники" могут не соответствовать современным жестким критериям сетевой безопасности и поэтому запускать Web–серверы на них не рекомендуется. Может показаться, что ограничение возможности выбора операционной системы сервера приведет к потере гибкости всей информационной системы, но это не так. Поскольку предполагается использование Web–сервера из сети интернет, то любой клиент должен иметь возможность пользоваться сервером, независимо от его платформы – это основное требование совместимости платформ. И оно выполняется. Поскольку обмен информацией между клиентом и сервером в среде Интернета осуществляется с использованием кроссплатформенных стандартов, а конкретно – протоколов низкого (TCP/IP) и высокого (HTTP) уровней, то в общем случае нет особой необходимости знать, какой именно клиент пользуется ресурсом. Запрос клиента должен содержать адрес (имя) Web–страницы, которая затем будет отправлена сервером клиенту, или имя выполняемого модуля, который будет выполнен сервером для того, чтобы динамически сформировать и послать клиенту страницу. Адреса запрашиваемых документов содержатся в их URL (Universal Resources Locator) или в терминах спецификации языка HTML 4.0 – URI (Universal Resource Identifier). Для того чтобы подготовить статические страницы, необходимо владеть языком HTML или визуальными средствами подготовки Web–страниц (например, Microsoft Frontpage или Macromedia DreamWeaver). Динамические

страницы формируются программой, называемой CGI-модулем (CGI-скриптом) в соответствии с дополнительными данными, содержащимися в запросе или присланными клиентом, и на основании правил, заложенных при создании CGI-модуля. При создании динамической страницы модуль может использовать любые данные, доступные ему в настоящий момент времени, как, например, результаты поиска в базе данных. Благодаря этому свойству, можно заранее обеспечить гибкие интерактивные возможности Web-сервера. Именно для обеспечения этих свойств были созданы следующие языки и технологии.

ASP (Active Server Pages) Активные серверные страницы – технология, разработанная фирмой Microsoft, для написания сценариев создания Web-страниц на языке VBScript или JavaScript. Достоинством данной технологии является интеграция с другими продуктами этой фирмы, и как следствие, автоматизация многих операций.

Perl – язык программирования, "перекочевавший" на платформу Win32 с Unix-систем. Достоинством является простота и большое количество исходных текстов, бесплатно доступных в Интернете – следствие его популярности при написании CGI-модулей на Unix-платформе. Недостатком является узкая область применения и необходимость установки интерпретатора языка на Web-сервер (дело в том, что изначально язык Perl не имел компилятора, а был интерпретируемым). Кроме того, функциональные возможности этого языка очень ограничены.

Возможности Borland Delphi. Среда Delphi предоставляет богатые возможности для создания сетевых проектов. Программисту доступны практически все средства, которые используются при создании настольных приложений: Object Inspector (Инспектор объектов), Component palette (Палитра компонентов), Code editor (Редактор кода) и т. д. Создание CGI-модулей максимально автоматизировано и может проходить двумя путями: с помощью стандартного набора невизуальных компонентов или непосредственного кодирования, в соответствии со спецификацией CGI. Кроме того, в поставке Delphi Client Server Suite есть библиотеки, в которых инкапсулированы объекты, существенно ускоряющие процесс написания программ. Короче, есть все, для быстрого и качественного создания различных интернет-приложений: начиная от браузера и заканчивая Web-сервером. Мощь Object Pascal и высокоразвитые средства для ускорения работы превращают разработку интернет-приложений в удовольствие.

Следует отметить, что особое внимание данному направлению разработки приложений уделено в 4 и 5 версиях Delphi. В последнее время грань различий между клиентом и сервером размывается. Например, на базе операционной системы Windows 95, традиционно считавшейся клиентской ОС, можно установить Web-сервер, а пакет MS Office 2000 поддерживает публикации баз данных в WWW.

IP-адресация и служба DNS компьютеров в сети очень много. Однако с момента создания интернета и до сих пор существует возможность каждому компьютеру соединиться с любым другим. Как это достигается? Практически

все компьютеры, находящиеся в сети, используют один набор протоколов – TCP/IP.

Каждый компьютер имеет свой уникальный IP–адрес. Этот адрес назначается провайдером из пространства доступных ему адресов на время сеанса (динамический адрес) или на все время пользования клиентом услугами данного провайдера (статический адрес). IP–адрес может иметь вид: 195.242.9.22. Провайдеры, у которых число пользователей больше пространства доступных адресов, пользуются динамической адресацией. Серверы всегда получают статический адрес. Для более удобной навигации в сети интернет создана система Domain Name Service (DNS), сопоставляющая символьные названия IP–адресам. Например, IP–адресу, приведенному выше, соответствует имя: www.bhv.ru. В этом случае пользователи избавлены от необходимости помнить четыре трехзначные цифры при подключении к серверам. Вместо этого они ассоциативно подключаются к нужному серверу по названию, которое обычно соответствует либо профилю предоставляемой информации, либо имени фирмы или частного лица, поддерживающих данный узел. Кроме того, адресация системы DNS организована таким образом, чтобы структурировать все имена по разделам. За основу взято две системы структуризации: территориальная и организационная. Согласно территориальной структуризации, всем государствам присваивается имя своего домена: пространства адресов. При организационной структуризации имя домена присваивается некоторому классу, объединяющему различные учреждения. В обоих случаях домен верхнего уровня может адресовать узлы более низкого уровня. Например, домен ru может включать узлы в следующем виде: имя сервера.имя домена.ru.

Таким образом, интернет–провайдер, получая пространство IP–адресов, как правило, приобретает права на использование множества имен, заканчивающихся некоторой символьной записью, внесенной в базу данных системы DNS или провайдера более высокого уровня. Например, интернет–провайдер "Ранг и ганг", купивший право на символьное имя randk.com, может предоставлять своим клиентам право использования доменных имен типа klient.randk.com или ww.klient.randk.com.

Следует помнить, что приставка www была введена для обозначения узлов среды World Wide Web и в настоящее время общепринята для индексации Web–серверов, однако адреса DNS, приведенные выше, не эквивалентны, и могут адресовать различные ресурсы. В таблице 1.1 приведены принятые обозначения некоторых доменных имен верхних уровней.

Т а б л и ц а 1.1 – Соответствие некоторых доменных имен организационным и территориальным структурам

Имя домена	Принадлежность
com	Коммерческие организации
edu	Образовательные учреждения
org	Некоммерческие организации

Имя домена	Принадлежность
gov	Правительственные институты
mil	Военные службы
net	Сетевые структуры
ru	Россия
by	Белоруссия
us	США
ua	Украина
uk	Великобритания
pl	Польша
cz	Чехия
ca	Канада
fi	Финляндия
de	Германия
fr	Франция

Один сервер – несколько имен. В соответствии с принятой схемой обработки запросов пользователя в системе DNS не может существовать два ресурса с одинаковыми именами, однако один ресурс может иметь несколько имен.

При этом, IP–адрес сайта останется таким же, как и klient.randk.com, однако, как видим, символьные (или name) адреса будут разными, и пользователь Интернета сможет загрузить этот сайт под любым из получившихся name–адресов. Данная операция присваивания нескольких name–адресов одному ресурсу может делаться по нескольким причинам, главными из которых являются:

- сокращение числа доменных имен в URL ресурса;
- изменение названия сайта и обеспечение притока аудитории старого сайта;
- обозначение как географической, так и организационной принадлежности ресурса.

Если сайт размещен у провайдера с длинным name–адресом, то URL ресурса автоматически будет содержать несколько доменных имен верхних уровней по образцу: www.имяклиента.имяпровайдера1.имяпровайдера2. ru. Разумеется, такая ситуация мало кого устроит, особенно если владелец ресурса рассчитывает на большую аудиторию. Вряд ли клиент будет вводить такой длинный URL в свой браузер, если только сайт не публикует исключительную информацию. Для исправления такой ситуации владелец сайта может приобрести имя в домене com или ru, например. При этом тот же самый ресурс будет доступен и с name–адреса www.klient.com заметно упростив загрузку ресурса. Настройка всех параметров name и IP–адресации, как правило, входит в компетенцию провайдера, однако, конфигурирование Web–сервера

осуществляется самим администратором ресурса. В случае, если сайт меняет название, например в связи со сменой имени фирмы–владельца, то требуется обеспечить попадание на новый сайт тех клиентов, которые ввели адрес старого ресурса. Для этого новому Web–сайту и присваивается дополнительный name–адрес старого. Кроме того, корпорации, работающие в нескольких странах, очень часто содержат один сайт, предлагающий информацию на разных языках. Для того чтобы пользователь в некоторой стране смог обратиться к такому сайту, регистрируется имя ресурса в домене этой страны, и затем используется переадресация на конкретный региональный раздел корпоративного сайта. Например, клиент провайдера "Ранг и ганг", имеющий name–адрес `www.klient.com` и работающий на территории Польши, может зарегистрироваться в верхнем домене этой страны, создать раздел на польском языке вида `www.klient.com/poland/` и присвоить дополнительный name адрес этому разделу `www.klient.pl`, облегчив доступ к своему Web–сайту пользователям из Польши.

1.2 Последовательности создания сайта бесплатного дистанционного обучения

1.2.1 Постановка задачи

Постановка задачи определяет следующие моменты:

- для каких лиц создается данный сайт;
- цель проекта;
- задача задача;

Подробная разработка структуры сайта, как именно будут составлены разделы сайта. В свою очередь из разделов создается пункты меню, из меню категории. Также для постановки задачи необходимо разработать функциональность и дизайн сайта.

От функциональности вашего сайта, зависит его популярность, то есть посещаемость пользователями. Сайт должен быть легким в пользовании, если же вы сделаете сложную и не понятную систему управления, все меньше количество посетителей у вас будет на сайте.

Дизайн сайта, в нем не существует ограничений в фантазии, нет строгих правил. Дизайн зависит от цели создания сайта и должна соответствовать данной теме. Например цвет логотипа АУЭС красного и синего цветов, и официальный сайт оформлен в данной цветовой гамме. Как бы отражая тем самым стиль университета, придерживаясь и не отходя от цели. Но хотел бы отметить успех сайта не зависит от красоты дизайна, а важна достигнутая цель, по которой разрабатывался сайт.

Для продвижения сайта необходимы ключевые короткие названия, для легкости поиска в поисковых системах, а также для легкого запоминания пользователями названия сайта.

1.2.2 Назначение и область применения сайта

Данный сайт предназначен для всех желающих получить дистанционное образование. В дипломном проекте будут реализованы стандарты веб-сайта с применением системы динамического управления и наполнением на базе веб-интерфейса.

Разработка интерактивного web-сайта включает в себя выполнение следующего комплекса задач:

- разработка контента сайта;
- разработка дизайна сайта, включающая в себя создание графических элементов макета сайта и элементов навигации;
- создание форм, обеспечивающих интерактивное взаимодействие с клиентами корпорации;
- разработка панели управления содержимым сайта, для обеспечения возможности оперативного изменения и добавления информации на сайт;
- создание групп пользователей для организации безопасности работы в системе, путем распределения уровней доступа, модулями авторизации;
- разработка базы клиентов корпорации с возможностью просмотра данных обо всех клиентах корпорации, данных по предварительным заказам и данных по заключенным договорам для каждого из клиентов.

Контент сайта – это его информационное наполнение. Цель помещения контента на сайт состоит в том, чтобы предоставить клиентам корпорации максимально полную информацию о продуктах, предоставляемых сайтом, не утомяв его при этом слишком большими текстами и ненужной информацией.

Дизайн сайта – определяет визуальное оформление сайта и является характеристикой его внешнего вида. Дизайн предназначен для повышения эстетической ценности сайта от его содержимого.

Формы интерактивного взаимодействия с пользователями – позволяют получать информацию от клиента и отправлять ее на сервер для обработки. Служат для установления обратной связи с клиентами корпорации.

Панель управления содержимым сайта – используется для редактирования, удаления и добавления контента сайта, изменения структуры меню и способа навигации по сайту. Позволяет сотрудникам корпорации оперативно изменять информацию на сайте без привлечения сторонних специалистов, что позволяет избежать дополнительных затрат на сопровождение сайта и поддерживать актуальность представленной информации.

Организация безопасности – обеспечение безопасности работы в системе. Модуль авторизации позволяет ограничить доступ к панели управления сайтом, а также разграничить права пользователей. В частности, будет организовано две группы пользователей с предоставлением им различных возможностей доступа. Группе администраторов предоставляется полный доступ к панели управления с возможностью полного изменения web-сайта, группе менеджеров

отдела продаж предоставляется частичный доступ с возможностью изменения контента сайта и возможностью работы с базой клиентов корпорации.

Профили пользователя – содержат информацию из базы данных клиентов корпорации. Предоставляют возможность просмотра информации о клиентах корпорации. Содержат формы для ввода данных о договорах заключенных с клиентами. Введенная информация заносится в базу данных клиентов, что обеспечивает возможность ее последующего просмотра.

1.2.3 Анализ предметной области

Важным интегрированным фактором типологии дистанционных университетов является совокупность используемых в учебном процессе педагогических методов и приемов. Выбрав в качестве критерия способ коммуникации преподавателей и обучаемых, эти методы (приемы) можно классифицировать следующим образом:

1 Методы обучения посредством взаимодействия обучаемого с образовательными ресурсами при минимальном участии преподавателя и других обучаемых (самообучение). Для развития этих методов характерен мультимедиа подход, когда при помощи разнообразных средств создаются образовательные ресурсы: печатные, аудио, видео–материалы, и что особенно важно для электронных университетов – учебные материалы, доставляемые по компьютерным сетям. Это, прежде всего:

- интерактивные базы данных;
- электронные журналы;
- компьютерные обучающие программы (электронные учебники).

В интерактивных базах данных систематизируются массивы данных, которые могут быть доступны посредством телекоммуникаций. Используя эти ресурсы, разработчики курсов, например, могут поддерживать локальные базы данных как для студентов, так и для преподавателей. Другим решением является предоставление доступа к внешним базам данных. Число баз данных, доступных через компьютерные сети быстро растет.

Так, в Murdoch University каталог библиотечных услуг, которые доступны через Internet, насчитывает более 70 страниц. Студенты и преподаватели Государственного университета Огайо имеют доступ по крайней мере к 9 главным библиотекам и целому ряду баз данных через Internet. Пользователи CompuServe имеют доступ к таким базам данных, как Academic American Encyclopedia, Dissertation Abstracts, ERIC, Magazine Database Plus, Peterson's College Database.

Электронные журналы представляют собой периодические издания, которые распространяются среди подписчиков через компьютерные сети. Они становятся все более важным источником получения информации и обучения. Как утверждалось в U.S.News & World Report (1994), более 2700 газет в 1994 году предпринимали ту или иную попытку издания электронных версий, в то время как в 1989 году таких газет было лишь 42. Strangelove составил в 1992

году справочник, который включал 35 электронных журналов и 90 информационных бюллетеней, доступных через Internet. Студенты подписываются на такие журналы с целью использования их как неотъемлемой части курса или как дополнения к работе.

Компьютерные обучающие программы представляют собой программное обеспечение, которое может использоваться на удаленном компьютере через компьютерную сеть. Сеанс связи с удаленным компьютером может осуществляться при помощи, например, модемной связи или Telnet услуг в Internet.

2 Методы индивидуализированного преподавания и обучения, для которых характерны взаимоотношения одного студента с одним преподавателем или одного студента с другим студентом (обучение "один к одному"). Эти методы реализуются в дистанционном образовании в основном посредством таких технологий, как телефон, голосовая почта, электронная почта. Развитие теленаставничества (система "тьюторов"), опосредованного компьютерными сетями, является важным компонентом учебного процесса в электронных университетах.

3 Методы, в основе которых лежит представление студентам учебного материала преподавателем или экспертом, при котором обучающиеся не играют активную роль в коммуникации (обучение "один ко многим").

Эти методы, свойственные традиционной образовательной системе, получают новое развитие на базе современных информационных технологий. Так, лекции, записанные на аудио или видеокассеты, читаемые по радио или телевидению, дополняются в современном дистанционном образовательном процессе так называемыми "э-лекциями" (электронными лекциями), т.е. лекционным материалом, распространяемым по компьютерным сетям с помощью систем досок объявлений (BBS). Э-лекция может представлять собой подборку статей или выдержек из них, а также учебных материалов, которые готовят обучающихся к будущим дискуссиям. На базе технологии электронной доски объявлений развивается также метод проведения учебных электронных симпозиумов, представляющих собой серию выступлений нескольких авторитетов ("первых спикеров").

4 Методы, для которых характерно активное взаимодействие между всеми участниками учебного процесса (обучение "многие ко многим"). Значение этих методов и интенсивность их использования существенно возрастает с развитием обучающих телекоммуникационных технологий. Иными словами, интерактивные взаимодействия между самими обучающимися, а не только между преподавателем и обучающимися, становятся важным источником получения знаний. Развитие этих методов связано с проведением учебных коллективных дискуссий и конференций. Технологии аудио, аудиографических и видеоконференций позволяют активно развивать такие методы в дистанционном образовании. Особую роль в учебном процессе дистанционных университетов играют компьютерные конференции, которые позволяют всем участникам дискуссии обмениваться письменными

сообщениями как в синхронном, так и в асинхронном режиме, что имеет большую дидактическую ценность.

Компьютерно–опосредованные коммуникации позволяют активнее использовать такие методы обучения, как дебаты, моделирование, ролевые игры, дискуссионные группы, мозговые атаки, методы Дельфи, методы номинальной группы, форумы, проектные группы.

Так, метод "мозговой атаки" представляет собой стратегию взаимодействия, позволяющую группам студентов эффективно генерировать идеи. Этот метод поощряет членов группы мыслить творчески и развивать идеи других членов группы. Основной целью метода мозговой атаки является создать фонд идей по определенной теме. При мозговой атаке исключается критицизм, поощряются свободные ассоциативные суждения.

Процедура Дельфи представляет собой метод для выработки надежного консенсуса номинальной группы студентов посредством серии анкетных опросов. Термин номинальная группа происходит от того, что студенты только номинально представляют собой группу на первоначальной стадии генерации идей. Первоначально каждого участника такой группы просят сформулировать и проранжировать идеи. Затем составляется общий список идей обычно путем выявления идей, которые получили самый высокий приоритет у отдельных участников, затем вторые по значимости и т.д. до тех пор, пока список у каждого участника не будет исчерпан. После этого все приглашаются к обсуждению идей. После дискуссии проводится голосование, в ходе которого членов группы просят проранжировать идеи, которые были генерированы в ходе дискуссии. В University of Auckland была разработана программная система для поддержки синхронных групповых занятий (groupware system), которая применялась в курсе по менеджменту.

С целью классификации дистанционных университетов по педагогическим принципам, лежащим в основе их учебной практики, целесообразно выделить следующие принципы телематических систем образования:

- интерактивность учебного процесса;
- обучение как диалог;
- адаптивность обучения;
- гибкость учебного материала;
- «передаваемость» материала в дистанционном образовании;
- активность обучаемого.

Дистанционные образовательные учреждения обычно основываются не на каком–то одном из этих принципов, а на их совокупности. Тем не менее, обычно выделяются доминирующие.

1.2.4 Основные типы организационных структур дистанционного образования

Основные типы организационных структур университетского дистанционного образования включают в себя:

- подразделения заочного (дистанционного) образования в традиционных университетах;
- консорциум университетов;
- открытые университеты;
- виртуальные университеты.

Характерной тенденцией дистанционного образования является объединение организационных структур университетов. Так, в последние годы стал развиваться новый тип организационной структуры дистанционного университетского образования – консорциум университетов. Дистанционные образовательные услуги оказывает специальная организация, объединяющая и координирующая деятельность нескольких университетов. Консорциум университетов предлагает набор курсов, разработанных в различных университетах – от курсов для абитуриентов до курсов на получение ученых степеней. В 70–х и 80–х годах во многих странах были учреждены национальные открытые университеты. Они использовали многие организационные принципы заочного обучения. Но в целом открытое образование привнесло много нового в образовательную систему. Принцип открытости образования означает свободу зачисления в число обучаемых и составления индивидуального учебного плана, а также свободу места, времени и темпов обучения. В основе открытого образования – богатая и детально разработанная образовательная среда, в которой обучаемый ориентируется вполне самостоятельно, стремясь к достижению стоящих перед ним образовательных целей.

В основе новой системы образования лежит принцип открытости, который применительно к высшему образованию означает:

- открытое поступление в высшее учебное заведение, т.е. отказ от любых условий и требований для зачисления, кроме достижения необходимого возраста (18 лет);
- открытое планирование обучения, т.е. свобода составления индивидуальной программы обучения путем выбора из системы курсов;
- свобода в выборе времени и темпов обучения, т.е. прием студентов в ВУЗ в течение всего года и отсутствие фиксированных сроков обучения;
- свобода в выборе места обучения: студенты физически отсутствуют в учебных аудиториях основную часть учебного времени и могут самостоятельно выбирать, где обучаться.

Проведение принципа открытости привело к значительным организационным новшествам, которые стали практически осуществимы именно благодаря внедрению новых технологий хранения, переработки и передачи информации. Так, например, в 90–х годах появилась новая модель

дистанционного образования на базе технологий проведения телеконференций. Эта модель называется телеобучением или телеобразованием. В этом случае проведение телеконференций, которые могут быть и в реальном времени, является главной формой взаимодействия между учителем и обучающимся, расширяя это взаимодействие, ранее осуществляемое, главным образом, по почте. При этом телеконференции могут проводиться как между преподавателем и учениками, так и между самими обучающимися. Это могут быть аудио, аудиографические, видео и компьютерные телеконференции.

Модель телеобразования появилась недавно, но она ведет к радикальным изменениям в организации современного образования. Это ярко проявляется в том, что на базе этой модели стала развиваться новая организационная форма современного образования – виртуальные университеты. Эта форма обучения рассматривается нами как новая, только что наметившаяся модель образования. В этой модели полностью реализуются те потенциальные возможности перестройки системы образования, которые имеют технологии телеконференций, используемые в учебных целях. Эти технологии позволяют группам учащихся и отдельным обучаемым встречаться с преподавателями и между собой, находясь на любом расстоянии друг от друга. Такие современные средства коммуникации дополняются компьютерными обучающими программами, которые замещают печатные тексты, аудио и видеопленки. Появление такой модели дистанционного образования ведет к тому, что образование осуществляется не только на расстоянии, но и независимо от какого-либо учреждения. Такая модель еще не реализована полностью. Она сталкивается с существенными трудностями, в частности, проблемой получения общественного признания и права выдавать дипломы и сертификаты, присваивать соответствующие степени (проблема аккредитации виртуального университета). Преодоление этих трудностей и полное развитие модели виртуального университета будет означать глубокие изменения в организационной структуре современного образования.

1.2.5 Электронный учебник как средство дистанционного обучения

Как говорилось выше, мы разберем одну из наиболее часто применяемых технологий в дистанционном обучении и в очном образовании также. А именно, это электронный учебник, наиболее часто встречающаяся форма представления нового материала. Кроме этого ЭУ может включать одновременно тренажеры, лабораторные работы, а также тесты; т.е. одновременно – это и ПО по предоставлению знаний и по их контролю.

Основные принципы дистанционного обучения (ДО) являются установление интерактивного общения между обучающимся и обучающим без обеспечения их непосредственной встречи и самостоятельное освоение определенного массива знаний и навыков по выбранному курсу и его программе при заданной информационной технологии.

Главной проблемой развития дистанционного обучения является создание новых методов и технологий обучения, отвечающих телекоммуникационной среде общения. В этой среде ярко проявляется то обстоятельство, что учащиеся не просто пассивные потребители информации, а в процессе обучения они создают собственное понимание предметного содержания обучения.

На смену прежней модели обучения должна прийти новая модель, основанная на следующих положениях: в центре технологии обучения – учащийся; суть технологии – развитие способности к самообучению; учащиеся играют активную роль в обучении; в основе учебной деятельности – сотрудничество.

В связи с этим требуют пересмотра методики обучения, модели деятельности и взаимодействия преподавателей и обучаемых. Я считаю ошибочным мнение многих российских педагогов–практиков, развивающих технологии дистанционного образования, что дистанционный учебный курс можно получить, просто переведя в компьютерную форму учебные материалы традиционного очного обучения.

Успешное создание и использование дистанционных учебных курсов должно начинаться с глубокого анализа целей обучения, дидактических возможностей новых технологий передачи учебной информации, требований к технологиям дистанционного обучения с точки зрения обучения конкретным дисциплинам, корректировки критериев образования.

Дидактические особенности курса ДО обуславливают новое понимание и коррекцию целей его внедрения, которые можно обозначить следующим образом:

- стимулирование интеллектуальной активности учащихся с помощью определения целей изучения и применения материала, а также вовлечения учащихся в отбор, проработку и организацию материала;
- усиление учебной мотивации, что достигается путем четкого определения ценностей и внутренних причин, побуждающих учиться;
- развитие способностей и навыков обучения и самообучения, что достигается расширением и углублением учебных технологий и приемов.

К числу дидактических принципов, затрагиваемых компьютерными технологиями передачи информации и общения, в первую очередь следует отнести:

- принцип активности;
- принцип самостоятельности;
- принцип сочетания коллективных и индивидуальных форм учебной работы;
- принцип мотивации;
- принцип связи теории с практикой;
- принцип эффективности.

В связи с этими принципами средства учебного назначения, которые используются в образовательном процессе ДО, должны обеспечивать возможность:

- индивидуализировать подход к ученику и дифференцировать процесс обучения;
- контролировать обучаемого с диагностикой ошибок и обратной связью;
- обеспечить самоконтроль и самокоррекцию учебно–познавательной деятельности учащегося;
- демонстрировать визуальную учебную информацию;
- моделировать и имитировать процессы и явления;
- проводить лабораторные работы, эксперименты и опыты в условиях виртуальной реальности;
- прививать умение в принятии оптимальных решений;
- повысить интерес к процессу обучения;
- передать культуру познания и др. Хотелось бы подчеркнуть особую важность определения целей курса.

Для построения четкого плана курса необходимо:

- определить основные цели, устанавливающие, что учащиеся должны изучить;
- конкретизировать поставленные цели, определив, что учащиеся должны уметь делать;
- спроектировать деятельность учащегося, которая позволит достичь целей.

Очень важно добиваться того, чтобы поставленные цели помогли определить, что ожидается от учащихся после изучения этого курса. Конкретизация целей позволяет дать представление о том, что учащийся в состоянии будет сделать в конце каждого урока. Фактически необходима постановка целей для каждого урока курса.

Цели помогают сконцентрироваться на развитии познавательной деятельности учащихся и определить, на какой стадии он находится.

Правильно сформулированные цели позволят учащимся:

- настроить мышление на тему обучения;
- сфокусировать внимание на наиболее важных проблемах;
- тщательно подготовиться к тестам, заданиям и другим средствам оценивания.

Деятельность должна быть спроектирована в соответствии со сформулированными целями.

При планировании и разработке дистанционных учебных курсов необходимо принимать во внимание, что основные три компоненты деятельности педагога, а именно изложение учебного материала, практика, обратная связь, сохраняют свое значение и в курсах ДО.

Разработанный и реализованный подход к дистанционному обучению заключается в следующем:

- перед началом дистанционного обучения производится психологическое тестирование учащегося с целью разработки индивидуального подхода к обучению;

- учебный материал представлен в структурированном виде, что позволяет учащемуся получить систематизированные знания по каждой теме.

Содержание предлагаемого к освоению курса дистанционного обучения педагогически отработано и систематизировано и состоит из комплекса психологических тестов, программы обучения и электронного учебника, который удовлетворяет вышеизложенным принципам.

Первоначально обучающемуся высылаются комплекс психологических тестов и пробный урок. Полученные результаты психологического тестирования обрабатываются и на основе этого строится психологический портрет учащегося, с помощью которого выбираются методы и индивидуальная стратегия обучения.

Программа обучения – один из наиболее важных видов раздаточных материалов для учащихся, обучающихся дистанционно. Учащиеся обращаются к ней для получения точной и ясной информации. Такое руководство включает в себя:

- информацию о системе дистанционного обучения, методах ДО;
- биографическую информацию о преподавателе;
- технологию построения учебного курса;
- цели курса;
- критерии окончания обучения;
- часы телефонных консультаций;
- описание экзаменов, проектов письменных работ;
- другие инструкции.

Электронный учебник, содержит собственно учебные материалы для дистанционного обучения, разделен на не зависимые темы–модули, каждая из которых дает целостное представление об определенной тематической области, и способствует индивидуализации процесса обучения, т.е. обучающийся может выбрать из вариантов обучения: изучение полного курса по предмету или изучение только конкретных тем.

1.2.6 Дистанционное обучение в Казахстане

С развитием дистанционного обучения со временем необходимость в заочной форме обучения в казахстанских ВУЗах может отпасть, считает вице–министр образования и науки республики Тахир Балыкбаев, сообщает ИА Новости–Казахстан. "Есть такое задание, чтобы мы побольше привлекали рабочую молодежь для получения высшего образования. Сегодня его (заочное) закрыть полностью нельзя, но использование новых технологий, в том числе дистанционных, это позволит нам повысить качество образования в этом направлении. Поэтому это вопрос серьезный. Если будет хорошо развита

система дистанционного обучения, может через какое-то время необходимость сама отпадет", – сказал Балыкбаев на брифинге в понедельник.

При этом он отметил, что пока заочная форма образования отменяться не будет. "Заочное образование нас очень сильно волнует, потому что там качество образования оставляет желать лучшего. Но с другой стороны полностью отменить заочное образование – объективно такой ситуации нет", – заверил вице-министр.

По его данным, сейчас порядка 150 тысяч казахстанских студентов из 520 тысяч обучается на заочном отделении. По заочному обучению выделяются места только на педагогические специальности, порядка 1 тысячи мест из 35 тысяч грантов, по госзаказу. Самостоятельно поступают на заочное около четверти абитуриентов.

"Поэтому полностью закрыть это направление пока такого не будет", – добавил Балыкбаев. В свою очередь глава Минобрнауки Аслан Саринжипов добавил, что основным критерием для министерства выступает не форма обучения в ВУЗах, а качество. "Здесь я хотел бы подчеркнуть, что мы не акцентируем внимание на форме обучения, мы будем акцентировать внимание на качестве обучения, будь то очное или заочное", – сказал министр. "Если качественное обучение – вопросов нет. Если есть вопросы некачественного образования, продажи дипломов и оценок, с этим мы мириться не будем. Ни в очной форме, ни в заочной. Другое дело, что отдельные вузы превращают это (заочное образование) в источник получения дохода. Здесь есть вопросы", – заключил Саринжипов.

На рубеже веков объективные потребности развития современного общества обусловили создание во многих странах мира системы дистанционного образования. Актуальность подобной формы обучения для Казахстана и стран СНГ очевидна: дистанционное образование призвано сыграть жизненно важную роль в сохранении и развитии единого образовательного пространства наших стран. Большой вклад в этот интеграционный процесс, на наш взгляд, может и должна внести Республика Казахстан. Во всем мире экспорт образовательных услуг несет не только социокультурную, но и экономическую нагрузку.

Опыт зарубежных стран может быть оперативно апробирован и применен для повышения эффективности управления качеством педагогических технологий, применяемых в казахстанской системе дистанционного обучения. Казахстан стал первым центрально-азиатским государством, которое вошло в европейское образовательное пространство. «Великую хартию университетов» подписали уже 60 казахстанских вузов. В сфере высшего образования произошли значительные качественные изменения по развитию академической мобильности, внедрению трехступенчатой модели подготовки кадров, а также по созданию условий для предоставления автономии вузам страны. Наряду с этим, проведена реструктуризация высшего образования: введена трехступенчатая модель подготовки кадров: (бакалавриат, магистратура, докторантура PhD).

Кроме того, заложены новые принципы финансирования высшего образования, адекватные рыночным условиям. В 38 вузах республики реализуется двудипломное образование, в 131 вузе ведется обучение по кредитной технологии, в 42 – дистанционное обучение. Вместе с тем, уникальным образовательным проектом в системе высшего образования Казахстана является вуз международного класса «Назарбаев Университет». Данный университет работает в партнерстве с университетами, входящими в 30-ку лучших вузов мира. Студенты первого набора уже приступили к обучению по программе бакалавриата в трех школах – инженерии, наук и технологий, социальных наук. Положительным моментом развития дистанционного образования в Республике Казахстан, на наш взгляд, является разработанная Академией государственного управления при Президенте Республики Казахстан – Концепция создания дистанционного обучения государственных служащих в Академии государственного управления при Президенте Республики Казахстан (далее – Концепция). В соответствии с данной Концепцией, существующая система повышения квалификации государственных служащих не в состоянии в полной степени охватить и удовлетворить потребности государственных служащих в переподготовке и повышении квалификации.

2 Программы используемые при разработке данного сайта

2.1 Краткая история PHP

Началом PHP можно считать осень 1994 года, когда Рasmus Лердорф (Rasmus Lerdorf) решил расширить возможности своей Home-page (Домашней страницы) и написать небольшой движок для выполнения простейших задач. Такой движок был готов к началу 1995 года и назывался Personal Home Page Tools. Умел он не очень много – понимал простейший язык и всего несколько макросов.

К середине 1995 года появилась вторая версия, которая называлась PHP/FI Version 2. Приставка FI – присоединилась из другого пакета Рasmus, который умел обрабатывать формы (Form Interpreter). PHP/FI компилировался внутрь Apache и использовал стандартный API Apache. PHP скрипты оказались быстрее аналогичных CGI – скриптов, так как серверу не было необходимости порождать новый процесс. Язык PHP по возможностям приблизился к Perl, самому популярному языку для написания CGI-программ. Была добавлена поддержка множества известных баз данных (например, MySQL и Oracle). Интерфейс к GD – библиотеке, позволял генерировать картинки на лету. С этого момента началось широкое распространение PHP/FI.

В конце 1997 Зэв Сураски (Zeev Suraski) и Энди Гутманс (Andi Gutmans) решили переписать внутренний движок, с целью исправить ошибки интерпретатора и повысить скорость выполнения скриптов. Через полгода, 6 июня 1998 года вышла новая версия, которая была названа PHP 3. К лету 1999 года PHP 3 был включен в несколько коммерческих продуктов. По данным NetCraft на ноябрь 1999 PHP использовался в более чем 1 млн. доменах.

Возможности и преимущества PHP

PHP – язык программирования, используемый на стороне WEB-сервера для динамической генерации HTML-страниц. Об этом говорит и расшифровка его названия: PHP – Personal HyperText Processor. PHP – один из немногих языков программирования, созданных специально для разработки веб-приложений. Поэтому он включает в себя все функции, необходимые именно для работы на веб-сервере, и при этом лишен избыточности, свойственной многим его конкурентам.

Очень приятная особенность PHP – то, что его команды включаются в обычные HTML-страницы с помощью специальных тегов, которые и заставляют PHP-машину выполнять на сервере нужные действия. Программам на PHP не нужны специальные CGI-директории с особыми правами доступа. Более того, на одной страничке можно произвольно чередовать "простой" HTML и PHP-код. PHP-GTK является расширением PHP и не поставляется вместе с дистрибутивом PHP.

PHP не зависит от платформы. PHP прекрасно интегрируется во все популярные веб-серверы: Apache и IIS, Zens и Netscape Enterprise Server, работает под Windows и OS/2, MacOS и практически всеми UNIX-подобными системами. Как следствие – PHP работает практически у всех хостеров, разрешающих собственные выполняемые скрипты.

Замечательная особенность PHP – его интегрированность практически со всеми современными интернет-технологиями. PHP поддерживает большинство современных веб-протоколов: IMAP, FTP, POP, XML, SNMP и другие. PHP прекрасно работает с базами данных. Трудно найти СУБД, поддержка которой не была бы реализована в PHP. MySQL и MS SQL Server, PostgreSQL и Oracle, Sybase и Interbase. Один только перечень баз данных, поддерживаемых PHP, займет, наверное, целый экран.

PHP включает в себя огромное количество встроенных функций: обработки строк и массивов, работы с файловой системой и с HTTP, электронной почтой, датой и временем, кириллицей и другими национальными алфавитами. Когда я впервые начал программировать на PHP, то был просто поражен обилием встроенных функций. Благодаря им многие алгоритмы, требующие в большинстве языков написания программного кода размером в несколько экранов, реализуются на PHP одной командой (точнее, вызовом одной функции). Современные тенденции развития языков программирования не обошли стороной и PHP. Средства объектно-ориентированного программирования появились еще в PHP3. А в объектной модели PHP4 в полном объеме реализованы классические понятия объектно-ориентированного программирования: наследование, инкапсуляция и полиморфизм.

Основное отличие от CGI-скриптов, написанных на других языках, типа Perl или C – это то, что в CGI-программах вы сами пишете выводимый HTML-код, а, используя PHP вы встраиваете свою программу в готовую HTML-страницу, используя открывающий и закрывающий теги (в примере `<?php и ?>`).

Отличие PHP от JavaScript, состоит в том, что PHP-скрипт выполняется на сервере, а клиенту передается результат работы, тогда как в JavaScript-код полностью передается на клиентскую машину и только там выполняется.

Любители Internet Information Server найдут, что PHP очень похож на Active Server Pages (ASP), а энтузиасты Java скажут, что PHP похож на Java Server Pages (JSP). Все три языка позволяют размещать код, выполняемый на Web-сервере, внутри HTML страниц.

В нескольких словах – на PHP можно сделать все, что можно сделать с помощью CGI-программ. Например: обрабатывать данные из форм, генерировать динамические страницы, получать и посылать куки (cookies).

Кроме этого в PHP включена поддержка многих баз данных (databases), что делает написание Web-приложений с использованием БД до невозможности простым.

Вдобавок ко всему PHP понимает протоколы IMAP, SNMP, NNTP, POP3 и даже HTTP, а также имеет возможность работать с сокетами (sockets) и общаться по другим протоколам.

Разработчикам Web-приложений нет необходимости говорить, что web-страницы – это не только текст и картинки. Достойный внимания сайт должен поддерживать некоторый уровень интерактивности с пользователем: поиск информации, продажа продуктов, конференции и т.п. Традиционно все это реализовалось CGI-скриптами, написанными на Perl. Но CGI-скрипты очень плохо масштабируемы. Каждый новый вызов CGI, требует от ядра порождения нового процесса, а это занимает процессорное время и тратит оперативную память. PHP предлагает другой вариант – он работает как часть Web-сервера, и этим самым похож на ASP от Microsoft. Синтаксис PHP очень похож на синтаксис C или Perl. Люди, знакомые с программированием, очень быстро смогут начать писать программы на PHP. В этом языке нет строгой типизации данных и нет необходимости в действиях по выделению/освобождению памяти. Программы, написанные на PHP, достаточно легко читаемы. Написанный PHP – код легко зрительно прочитать и понять, в отличие от Perl-программ.

2.2 Язык гипертекстовой разметки HTML

Hyper Text Markup Language (HTML) – язык разметки гипертекста – предназначен для написания гипертекстовых документов, публикуемых в World Wide Web.

Гипертекстовый документ – это текстовый файл, имеющий специальные метки, называемые тегами, которые впоследствии опознаются браузером и используются им для отображения содержимого файла на экране компьютера.

С помощью этих меток можно выделять заголовки документа, изменять цвет, размер и начертание букв, вставлять графические изображения и таблицы. Но основным преимуществом гипертекста перед обычным текстом является возможность добавления к содержимому документа гиперссылок – специальных конструкций языка HTML, которые позволяют щелчком мыши перейти к просмотру другого документа.

HTML – документ состоит из двух частей: собственно текста, т. е. данных, составляющих содержимое документа, и тегов – специальных конструкций языка HTML, используемых для разметки документа и управляющих его отображением. Теги языка HTML определяют, в каком виде будет представлен текст, какие его компоненты будут исполнять роль гипертекстовых ссылок, какие графические или мультимедийные объекты должны быть включены в документ.

Графическая и звуковая информация, включаемая в HTML-документ, хранится в отдельных файлах. Программы просмотра HTML-документов (браузеры) интерпретируют флаги разметки и располагают текст и графику на экране соответствующим образом. Для файлов, содержащих HTML-документы приняты расширения .htm или .html.

В большинстве случаев теги используются парами. Пара состоит из открывающего `<имя_тега>` и закрывающего `</имя_тега>` тегов. Действие любого парного тега начинается с того места, где встретился открывающий тег, и заканчивается при встрече соответствующего закрывающего тега. Часто пару, состоящую из открывающего и закрывающего тегов, называют контейнером, а часть текста, окаймленную открывающим и закрывающим тегом, – элементом.

Последовательность символов, составляющая текст может состоять из пробелов, табуляций, символов перехода на новую строку, символов возврата каретки, букв, знаков препинания, цифр, и специальных символов (например #, +, \$, @), за исключением следующих четырех символов, имеющих в HTML специальный смысл: `<` (меньше), `>` (больше), `&` (амперсанд) и `"` (двойная кавычка). Если необходимо включить в текст какой-либо из этих символов, то следует закодировать его особой последовательностью символов.

Структура HTML–документа

Самым главным из тегов HTML является одноименный тег `<html>`. Он всегда открывает документ, так же, как тег `</html>` должен непременно стоять в последней его строке. Эти теги обозначают, что находящиеся между ними строки представляют единый гипертекстовый документ. Без этих тегов браузер или другая программа просмотра не в состоянии идентифицировать формат документа и правильно его интерпретировать.

HTML–документ состоит из двух частей: заголовка (`head`) и тела (`body`), расположенных в следующем порядке:

```
<html>
<head> Заголовок документа </head>
<body> Тело документа </body>
</html>
```

Чаще всего в заголовок документа включают парный тег `<title>... </title>`, определяющий название документа. Многие программы просмотра используют его как заголовок окна, в котором выводят документ. Программы, индексирующие документы в сети Интернет, используют название для идентификации страницы. Хорошее название должно быть достаточно длинным для того, чтобы можно было корректно указать соответствующую страницу, и в то же время оно должно помещаться в заголовке окна.

Тело документа является обязательным элементом, так как в нем располагается весь материал документа. Тело документа размещается между тегами `<body>` и `</body>`. Все, что размещено между этими тегами, интерпретируется браузером в соответствии с правилами языка HTML позволяющими корректно отображать страницу на экране монитора.

Текст в HTML разделяется на абзацы при помощи тега `<p>`. Он размещается в начале каждого абзаца, и программа просмотра, встречая его, отделяет абзацы друг от друга пустой строкой. Использование закрывающего тега `</p>` необязательно.

Если требуется «разорвать» текст, перенеся его остаток на новую строку, при этом, не выделяя нового абзаца, используется тег разрыва строки
. Он заставляет программу просмотра выводить стоящие после него символы с новой строки. В отличие от тега абзаца, тег
 не добавляет пустую строку. У этого тега нет парного закрывающего тега.

Язык HTML поддерживает логическое и физическое форматирование содержимого документа. Логическое форматирование указывает на назначение данного фрагмента текста, а физическое форматирование задает его внешний вид.

При использовании логического форматирования текста браузером выделяются различные части текста в соответствии со структурой документа. Чтобы отобразить название, используется один из тегов заголовка. Заголовки в типичном документе разделяются по уровням. Язык HTML позволяет задать шесть уровней заголовков: h1 (заголовок первого уровня), h2, h3, h4, h5 и h6. Заголовок первого уровня имеет обычно больший размер и насыщенность по сравнению с заголовком второго уровня. Пример использования тегов заголовков:

```
<h1>1. Название главы</h1>  
<h2>1.1. Название раздела</h2>
```

Теги физического форматирования непосредственно задают вид текста на экране браузера, например пара выделяет текст полужирным начертанием, <u></u> задает подчеркивание текста, управляет шрифтом текста.

Тег вставляет изображение в документ, как если бы оно было просто одним большим символом. Пример применения тега:

```
<img src = "picture.gif">
```

Для создания гипертекстовой ссылки используется пара тегов <a>... . Фрагмент текста, изображение или любой другой объект, расположенный между этими тегами, отображается в окне браузера как гипертекстовая ссылка. Активация такого объекта приводит к загрузке в окно браузера нового документа или к отображению другой части текущей Web-страницы. Гипертекстовая ссылка формируется с помощью выражения:

```
<a href = "document.html">ссылка на документ</a>
```

Href здесь является обязательным атрибутом, значение которого и есть URL-адрес запрашиваемого ресурса. Кавычки в задании значения атрибута href не обязательны. Если задается ссылка на документ на другом сервере, то вид гиперссылки такой:

Фотография11-
А

С помощью различных тегов можно рисовать таблицы, форматировать текст, вставлять в документ изображения, видео, звуковые файлы и прочее.

2.3 CSS – каскадные таблицы стилей

CSS используется создателями веб–страниц для задания цветов, шрифтов, расположения и других аспектов представления документа. Основной целью разработки CSS являлось разделение содержимого (написанного на HTML или другом языке разметки) и представления документа (написанного на CSS). Это разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом. Кроме того, CSS позволяет представлять один и тот же документ в различных стилях или методах вывода, таких как экранное представление, печать, чтение голосом (специальным голосовым браузером или программой чтения с экрана), или при выводе устройствами, использующими Шрифт Брайля. Стандарт CSS определяет приоритеты, в порядке которых применяются правила стилей, если для какого–то элемента подходят несколько правил одновременно.

Если HTML используется для структурирования содержимого гипертекстовой страницы, то CSS используется для форматирования этого структурированного содержимого.

CSS – Это каскадные таблицы стилей, которые являются поразительным изобретением для улучшения дизайна web–страниц. Он экономит время при разработке. Требуется меньше ресурсов, так как код может использоваться на разных элементах страницы и предоставляет новые возможности для оформления сайта.

CSS работает с форматированием и отображением HTML–документов. Корректирует шрифты, цвета, поля, рисунки, позиции элементов и многое другое. На сегодняшний день CSS поддерживается всеми браузерами.

Преимущества CSS таблиц стилей:

- управление отображением множества документов с помощью одной таблицы стилей;
- более точный контроль над внешним видом страниц.

Существует шесть типов таблиц стилей:

1 Стиль браузера – это стандартная таблица стилей, используемая браузером. Если никакие правила стиля не объявлены, применяются эти стандартные стили.

2 Стиль пользователя – пользователь может написать свою таблицу стилей и переопределить любые стили, созданные вами, изменив настройки браузера. Такой подход используется редко.

3 Встроенный стиль – стиль, который определяется непосредственно в элементе и применяется с использованием атрибута style. Такой подход полезен для стилей, одновременно применяемых к одному элементу, однако он не считается идеальным.

4 Внедренный стиль – этот стиль управляет представлением одного документа и размещается внутри элемента style HTML–документа.

5 Связанный стиль – это таблица стилей, которая связана с HTML–документом при помощи элемента link, размещенного в разделе документа head. Любой документ, связанный с данным типом таблицы стилей, получает все стили, определенные в ней, в чем и заключается преимущество управления языком CSS.

6 Импортированный стиль – этот тип похож на связанные стили, однако позволяет импортировать стили в связанную таблицу стилей или непосредственно в документ. Он полезен для реализации обходных путей и для управления множеством документов.

Это называется «каскадом», в котором для правил рассчитываются приоритеты или «веса», что делает результаты предсказуемыми.

2.4 Объектно–ориентированный язык JavaScript

JavaScript – является объектно–ориентированным скриптовым языком программирования. Он используется в составе WEB–страниц и предназначен для написания сценариев, работающих как на стороне клиента, так и на стороне сервера, позволяя обеспечивать двустороннее взаимодействие между сайтом и пользователем. Таким образом увеличивает привлекательность сайта. JavaScript не является «полноценным» языком программирования, так как ориентирован на возможности той среды в которой он используется.

JavaScript имеет доступ к таким объектам как окна, меню, диалоги, текстовые области, фреймы, ввод и вывод в Web–страницу. А так же сценарии JavaScript можно присоединить к событиям как нажатие клавиш и движение мыши, загрузка и выгрузка страниц и графических образов. Одна из возможностей JavaScript является обеспечение запросов к базам данных.

Язык JavaScript по сравнению с родственными объектно–ориентированными языками Java и C++ не содержит в своем составе классы объектов. Он поддерживает конструкторы, которые создают объекты путем выделения для них памяти. Объекты создаются путем вызова конструктора в операции new название элемента.

Как и в других языках программирования, программа на языке JavaScript – это текст, который состоит из операторов, блоков и комментариев. Операторы могут содержать переменные, константы и выражения. Для добавления JavaScript – кода на страницу, в редакторе можно использовать теги <script></script>. Ниже приведен пример простого вывода модального окна с сообщением через JavaScript:

```
<script type="text/javascript"> -- начало скрипта
    Alert('Привет!!!'); -- модальное окно с сообщением
</script> -- закрытие скрипта
```

На html странице этот код будет выглядеть таким образом:

```
<html>
<head>
    <title>Пример JavaScript</title>
    <script type="text/javascript">
        Alert('Привет!!!'); </script>
</head>
<body>
    <input type=button name="p_button" value="Сообщение"
        onclick="form1">
</body>
</html>
```

2.5 Основы XML

На сегодняшний день уже всем специалистам в области Web-технологий стало очевидно, что существующих стандартов передачи данных по Internet недостаточно. Формат HTML, став в свое время прорывом в области отображения содержимого узлов Internet, уже не удовлетворяет всем необходимым на данный момент требованиям. Он позволяет описать то, каким образом должны быть отображены данные на экране конечного пользователя, но не предоставляет никаких средств для эффективного описания передаваемых данных и управления ими.

Кроме того, камнем преткновения для многих компаний, занимающихся разработкой программного обеспечения, является необходимость совместного использования различных компонент, обеспечения их взаимодействия, возможности обмена данными между ними.

До недавнего времени не существовало стандарта, предоставляющего средства для интеллектуального поиска информации, обмена данными, адаптивной обработки получаемых данных.

Решением всех описанных выше проблем стал утвержденный в 1998 году международной организацией W3C язык XML. XML (eXtensible Markup Language) – это расширяемый язык разметки, предназначенный для описания в текстовой форме структурированных данных. Этот текстовый (text-based) формат, во многом схожий с HTML, разработан специально для хранения и передачи данных.

XML позволяет описывать и передавать такие структурированные данные, как:

- отдельные документы;
- метаданные, описывающие содержимое какого-либо узла Internet;

- объекты, содержащие данные и методы работы с ними (например, элементы управления ActiveX или объекты Java);
- отдельные записи (например, результаты выполнения запросов к базам данных);
- всевозможные Web–ссылки на информационные и людские ресурсы Internet (адреса электронной почты, гипертекстовые ссылки и пр.).

Данные, описанные на языке XML, называются XML–документами. Язык XML легко читаем и достаточно прост для понимания. Если Вы были знакомы с HTML, то научиться составлять XML–документы не составит для Вас никакого труда.

Исходный текст XML–документа состоит из набора XML–элементов, каждый из которых содержит начальный и конечный тэги. Каждая пара тэгов представляет часть данных. То есть, как и HTML, язык XML для описания данных использует тэги. Но, в отличие от HTML, XML позволяет использовать неограниченный набор пар тэгов, каждая из которых представляет не то, как заключенные в нее данные должны выглядеть, а то, что они означают.

Любой элемент XML–документа может иметь атрибуты, уточняющие его характеристики. Атрибут – это пара имя = "значение", которая задается при определении элемента в начальном тэге.

Принцип расширяемости языка XML состоит в возможности использования неограниченного количества пар тэгов, определяемых создателем XML–документа.

Принцип независимости определения внутренней структуры документа от способов представления этой информации состоит в отделении данных от процесса их обработки и отображения. Таким образом, полученные данные можно использовать в соответствии с нуждами клиента, то есть выбирать нужное оформление, применять необходимые методы обработки.

Управлять отображением элементов в окне программы–клиента (например, в окне браузера) можно с помощью специальных инструкций – стилевых таблиц XSL (eXtensible Stylesheet Language). Эти таблицы XSL позволяют определять оформление элемента в зависимости от его месторасположения внутри документа, то есть к двум элементам с одинаковым названием могут применяться различные правила форматирования. Кроме того, языком, лежащим в основе XSL, является XML, а это означает, что таблицы XSL более универсальны, а для контроля корректности составления таких стилевых таблиц можно использовать DTD–описания или схемы данных, рассмотренные ниже.

Формат XML, по сравнению с HTML, имеет небольшой набор простых правил разбора, который позволяет разбирать XML–документы, не прибегая к каким–либо внешним описаниям используемых XML–элементов. В общем случае XML–документы должны удовлетворять следующим требованиям:

- каждый открывающий тэг, определяющий некоторую часть данных в документе, обязательно должен сопровождаться закрывающим, то есть, в отличие от HTML, нельзя опускать закрывающие тэги;

- вложенность тэгов в XML строго контролируется, поэтому необходимо следить за порядком следования открывающих и закрывающих тэгов;

- в XML учитывается регистр символов;

- вся информация, располагающаяся междуначальным и конечным тэгами, рассматривается в XML как данные, и поэтому учитываются все символы форматирования (то есть пробелы, переводы строк, табуляции не игнорируются, как в HTML).

В XML существует набор зарезервированных символов, которые должны быть заданы в XML–документе только специальным образом. Многие специалисты рассматривают XML как новую технологию интеграции программных компонент. Основными преимуществами использования XML являются:

- интеграция данных из различных источников. XML можно использовать для объединения разнородных структурированных данных на среднем уровне трехуровневых Web–систем, баз данных;

- локальная обработка данных. Полученные данные в формате XML можно разбирать, обрабатывать и отображать непосредственно на клиенте без дополнительных обращений к серверу.

Просмотр и манипулирование данными в различных разрезах. Полученные данные могут обрабатываться и просматриваться клиентом различными способами в зависимости от нужд конечного пользователя.

Возможность частичного обновления данных. С помощью XML можно обновлять только ту часть структурированных данных, которая была изменена, а не всю структуру целиком.

Все эти преимущества делают XML незаменимым инструментом для разработки гибких средств поиска информации в базах данных, мощных трехуровневых Web–приложений, а также приложений, поддерживающих транзакции. Другими словами, с помощью XML можно формировать запросы к базам данных различных структур, что позволяет осуществлять поиск информации в многочисленных несовместимых друг с другом базах данных.

2.6 Система управления базами данных MySQL

Компьютеры так прочно вошли в нашу жизнь, потому что их можно запрограммировать на выполнение утомительных, повторяющихся операций и решение задач, которые нам самим было бы не под силу решить без их вычислительной скорости и емкости информационных носителей. Помещение информации на бумагу и разработка схемы хранения бумаг в папках и картотеках – достаточно четко отработанный процесс, но многие вздохнули с облегчением, когда задача свелась к перемещению электронных документов в папки на жестком диске.

Одной из функций баз данных является упорядочение и индексация информации. Как и в библиотечной картотеке, не нужно просматривать

половину архива, чтобы найти нужную запись. Все выполняется гораздо быстрее.

Не все базы данных создаются на основе одних и тех же принципов, но традиционно в них применяется идея организации данных в виде записей. Каждая запись имеет фиксированный набор полей. Записи помещаются в таблицы, а совокупность таблиц формирует базу данных.

Для работы с базой данных необходима СУБД (система управления базами данных), т.е. программа, которая берет на себя все заботы, связанные с доступом к данным. Она содержит команды, позволяющие создавать таблицы, вставлять в них записи, искать и даже удалять таблицы.

MySQL – это быстрая, надежная, открыто распространяемая СУБД. MySQL, как и многие другие СУБД, функционирует по модели "клиент/сервер". Под этим подразумевается сетевая архитектура, в которой компьютеры играют роли клиентов либо серверов. На рисунке 2.1 изображена схема передачи информации между компьютером клиента и жестким диском сервера.

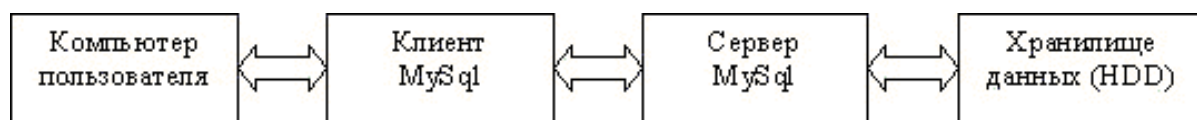


Рисунок 2.1 – Схема передачи данных в архитектуре "клиент/сервер"

СУБД управляет одной или несколькими базами данных. База данных представляет собой совокупность информации, организованной в виде множеств. Каждое множество содержит записи унифицированного вида. Сами записи состоят из полей. Обычно множества называют таблицами, а записи – строками таблиц.

Такова логическая модель данных. На жестком диске вся база данных может находиться в одном файле. В MySQL для каждой базы данных создается отдельный каталог, а каждой таблице соответствуют три файла. В других СУБД могут использоваться иные принципы физического хранения данных.

Строки таблиц могут быть связаны друг с другом одним из трех способов. Простейшее отношение – "один к одному". В этом случае строка первой таблицы соответствует одной единственной строке второй таблицы. На диаграммах такое отношение выражается записью 1:1.

Отношение "один ко многим" означает ситуацию, когда строка одной таблицы соответствует нескольким строкам другой таблицы. Это наиболее распространенный тип отношений. На диаграммах он выражается записью 1:N.

Наконец, при отношении "многие ко многим" строки первой таблицы могут быть связаны с произвольным числом строк во второй таблице. Такое отношение записывается как N:M.

Программист, работающий с базой данных, не заботится о том, как эти данные хранятся, и приложения, взаимодействующие с СУБД, не знают о

способе записи данных на диск. "Снаружи" виден лишь логический образ данных, и это позволяет менять код СУБД, не затрагивая код самих приложений.

Подобная обработка данных осуществляется посредством языка четвертого поколения (4GL), который поддерживает запросы, записываемые и исполняемые немедленно. Данные быстро утрачивают свою актуальность, поэтому скорость доступа к ним важна. Кроме того, программист должен иметь возможность формулировать новые запросы. Они называются нерегламентированными (ad hoc), поскольку не хранятся в самой базе данных и служат узкоспециализированным целям.

Язык четвертого поколения позволяет создавать схемы – точные определения данных и отношений между ними. Схема хранится как часть базы данных и может быть изменена без ущерба для данных.

Схема предназначена для контроля целостности данных. Если, к примеру, объявлено, что поле содержит целочисленные значения, то СУБД откажется записывать в него числа с плавающей запятой или строки. Отношения между записями тоже четко контролируются, и несогласованные данные не допускаются. Операции можно группировать в транзакции, выполняемые по принципу "все или ничего".

СУБД обеспечивает безопасность данных. Пользователям предоставляются определенные права доступа к информации. Некоторым пользователям разрешено лишь просматривать данные, тогда как другие пользователи могут менять содержимое таблиц.

СУБД поддерживает параллельный доступ к базе данных. Приложения могут обращаться к базе данных одновременно, что повышает общую производительность системы. Кроме того, отдельные операции могут "распараллеливаться" для еще большего улучшения производительности.

Наконец, СУБД помогает восстанавливать информацию в случае непредвиденного сбоя, незаметно для пользователей создавая резервные копии данных. Все изменения, вносимые в базу данных, регистрируются, поэтому многие операции можно отменять и выполнять повторно.

MySQL – свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой Sun Microsystems, которая ранее приобрела шведскую компанию MySQL AB. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией. Помимо этого, разработчики создают функциональность по заказу лицензионных пользователей. Именно благодаря такому заказу почти в самых ранних версиях появился механизм репликации.

MySQL является решением для малых и средних приложений. Входит в состав серверов WAMP, AppServ, LAMP и в портативные сборки серверов Денвер, XAMPP. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит

библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы.

Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц.

26 февраля 2008 года Sun Microsystems приобрела MySQL AB за \$1 млрд, 27 января 2010 года Oracle приобрела Sun Microsystems за \$7,4 млрд и включила MySQL в свою линейку СУБД.

Сообществом разработчиков MySQL созданы различные ответвления кода, такие как Drizzle (англ.), OurDelta, Percona Server, и MariaDB. Все эти ответвления уже существовали на момент поглощения компании Sun корпорацией Oracle.

MySQL портирована на большое количество платформ: AIX, BSDi, FreeBSD, HP-UX, Linux, MacOSX, NetBSD, OpenBSD, OS/2 Warp, SGI IRIX, Solaris, SunOS, SCO OpenServer, UnixWare, Tru64, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Server 2003, WinCE, Windows Vista и Windows 7. Существует также порт MySQL к OpenVMS. Важно отметить, что на официальном сайте СУБД для свободной загрузки предоставляются не только исходные коды, но и откомпилированные и оптимизированные под конкретные операционные системы готовые исполняемые модули СУБД MySQL.

Максимальный размер таблиц в MySQL 3.22 до 4 Гб, в последующих версиях максимальный размер ограничивается максимальным размером файла используемой операционной системы. Размер таблицы ограничен её типом. В общем случае тип MyISAM ограничен предельным размером файла в файловой системе операционной системы. Например в NTFS этот размер теоретически может быть до 32 эксабайт. В случае InnoDB одна таблица может храниться в нескольких файлах, представляющих единое табличное пространство. Размер последнего может достигать 64 терабайт. В отличие от MyISAM, в InnoDB имеется значительное ограничение на количество столбцов, которое можно добавить в одну таблицу. Размер страницы памяти по умолчанию составляет 16 килобайт, из которых под данные отведено 8123 байта. Размер указателя на динамические поля составляет 20 байт. Таким образом, в случае использования динамического формата строки (ROW_FORMAT=DYNAMIC), одна таблица может вместить максимум 409 столбцов типа blob или text.

MySQL имеет API для языков Delphi, C, C++, Эйфель, Java, Лисп, Perl, PHP, Python, Ruby, Smalltalk, Компонентный Паскаль и Tcl, библиотеки для языков платформы .NET, а также обеспечивает поддержку для ODBC посредством ODBC-драйвера MyODBC.

3 Создание базы данных и приложения к ней

3.1 СУБД

Данный разработанный сайт по функциональности требует базу данных. Необходимую для хранения и ввода информации. Для этого необходимо выбрать и создать базу данных.

Для разработки данного сайта как сервер я выбрал Wampserver, MySQL входит в состава пакета данного сервера. Исходя из всего выше сказано, база данных данного сайта осуществляется посредством MySQL.

Одним из основных достоинств Wampserver является то, что он включает практически самые новые компоненты (Apache, PHP, MySQL, SQLitemanager, Phpmyadmin).

Щелчок левой клавишей мышки на икону Wampserver позволит:

- управлять настройками Apache и MySQL;
- менять статус в сети/не в сети (доступно каждому пользователю или же лимитировано на локальном хосте);
- устанавливать или менять версию Apache, MySQL и PHP;
- управлять параметрами конфигураций серверов;
- получать доступ к логам;
- получать доступ к файлам конфигураций;
- создавать псевдонимы и др.

Клик правой клавишей мышки позволит поменять язык меню Wampserver и перейти сразу на эту страницу.

База данных должна содержать следующие информации:

- информацию о зарегистрированных пользователях;
- лекциях, курсах, тестах;
- компонентах;
- модулях;
- правах доступа;
- галерея, видео;
- настройки системы.

В таблице 3.1 представлено полный список необходимой информации и ее краткое содержание.

Т а б л и ц а 3.1 – Перечень хранимой информации

Название выходной формы	Содержание
Сведения о пользователях	Уникальный номер, логин, пользовательская группа, электронная почта, дата регистрации
Книги	Уникальный номер, название
Лекции	Уникальный номер, название
Курсы	Уникальный номер, название

Название выходной формы	Содержание
Тесты	Уникальный номер, название
Фотографии	Уникальный номер, название
Видео	Уникальный номер, название
Форум	Уникальный номер, название
Чат (Преподаватель–Студент)	Уникальный номер, название
Онлайн лекции	Уникальный номер, название
Онлайн конференция	Уникальный номер, название
Настройки	Уникальный номер, название, описание, тип

3.2 Разграничение доступа

Система управления базами данных MySQL очень часто применяется для хранения важной информации на веб-сайтах. Если это обычный сайт или форум – в базе могут храниться сообщения пользователей, данные для динамических страниц, данные о посещениях, если это какой-либо интерактивный сервис, то кроме данных про доступ (конфиденциальных), там хранится и другая информация о пользователе и его действиях. Все это приводит к тому, что общая безопасность сайта, вернее всей веб-системы, зависит от того, насколько защищен именно сервер базы данных.

На обычных виртуальных хостингах каждый клиент получает свой логин и пароль, и ему доступна только одна база, в которой он может создавать произвольное количество таблиц. Один и тот же физический сервер БД используют разные клиенты, каждый из которых имеет доступ только до одной определенной базы данных. Ситуация, когда у пользователя одна база, за владение которой сражаются и "движок" форума (которому требуется создать сотню и больше таблиц), и скрипты списков рассылки, новостей, поисковый скрипт, а если еще установлена система управления контентом (CMS) или электронный магазин – тогда в этой базе возникает такое огромное количество различных таблиц, порой с очень странными и ничего не обозначающими названиями (хорошо, если два скрипта не используют таблицы с одинаковыми названиями, но разной структурой). В таких случаях очень желательно иметь возможность создать несколько отдельных баз данных, и выделить их для разных приложений (к примеру, одна база для форума, другая для электронного магазина).

Кроме сложностей с управлением несколькими сотнями таблиц в одной базе, вы столкнетесь с необходимостью ограничивать доступ разных пользователей к таблицам, базам и даже отдельным столбцам конкретной таблицы. Спешим вас успокоить – разработчики СУБД MySQL уже позаботились о подобной ситуации – в MySQL есть очень гибкий и мощный

механизм управления и разграничения доступа пользователей к базам и таблицам.

Работает этот механизм, естественно, через служебные таблицы. В списке баз данных есть одна служебная база под названием "mysql", в которой хранятся в нескольких таблицах все служебные данные, необходимые для работы сервера. Пока нас интересует только управление правами доступа представлена на рисунке 3.1, за которые отвечают следующие таблицы:

- columns_priv – доступ на уровне столбцов таблицы;
- db – доступ к отдельным базам данных;
- host – доступ к базам с конкретных хостов/IP;
- tables_priv – доступ к отдельным таблицам базы данных;
- user – глобальные настройки доступа для конкретных пользователей.



таблица	Действие	Записи	Тип	Размер
<input type="checkbox"/> columns_priv		0	MyISAM	1.0 KB
<input checked="" type="checkbox"/> db		3	MyISAM	3.4 KB
<input type="checkbox"/> func		0	MyISAM	1.0 KB
<input type="checkbox"/> host		0	MyISAM	1.0 KB
<input type="checkbox"/> tables_priv		0	MyISAM	1.0 KB
<input type="checkbox"/> user		6	MyISAM	2.6 KB
6 таблиц(ы)		Всего		10.1 KB

Рисунок 3.1 – Управления правами доступа

Давайте детально рассмотрим таблицу user и специфику работы с ней, работа же с остальными таблицами аналогична и не будет сложнее.

Права, определяемые в таблице user, являются глобальными, это значит, что они применяются ко всем базам данных, к которым имеет доступ конкретный пользователь. также в этой таблице хранятся сами имена пользователей и их пароли. Причем имена пользователей хранятся в виде открытого текста, а пароли зашифрованы функцией PASSWORD(). Сервер MySQL применяет свой метод шифрования вместо системного (если такая функция предоставляется ОС, к примеру, Linux/UNIX).

Поскольку MySQL предназначена для работы через сеть, то к серверу могут подключаться клиенты с различных хостов. Поэтому в таблице привилегий есть столбец host, в котором можно задать конкретный компьютер, IP или диапазон адресов, с которых определенный пользователь (или все, любые пользователи) могут подключаться, или же им полностью запрещен доступ. К примеру, если ваш тестовый сервер находится в локальной сети, то можно выборочно разрешить доступ к нему с отдельных компьютеров, а для всех остальных полностью заблокировать доступ.

Если для одного и того же пользователя надо разрешить доступ с различных компьютеров, то в таблице привилегий надо создать две (три или больше – сколько понадобится) записей, определяя в каждой отдельный

параметр доступа. К примеру, для доступа пользователя root с компьютеров localhost и 192.168.1.138 надо в таблицу user внести две одинаковые записи, которые различаются только полем host. Хотя можно, конечно, и не делать их одинаковыми – тогда тот же пользователь, но в зависимости от места, откуда он подключился, будет иметь разные привилегии.

Значение столбца host можно задавать в различных форматах. Это может быть как имя хоста – localhost, так и IP-адрес – 192.168.1.138. При необходимости, можно применять символы подстановки. Они аналогичны тем же. Что применяются в синтаксисе SQL-запросов. Символ "_" означает любой символ, а "%" – любое количество символов. К примеру, для разрешения доступа с любого компьютера, содержащего в имени hostinfo.ru надо прописать %.hostinfo.ru в столбце host. таким же способом можно задавать и диапазоны IP-адресов. Для задания маски сети доступен также вариант написания через слеш – 192.168.1/255 представлено на рисунке 3.2. Для глобального задания доступа можно в столбце host прописать только символ "%" или оставить его пустым, что равнозначно.

←T→			Host	User	Password	Select_priv	Insert_priv	Update_priv
<input type="checkbox"/>			localhost	root		Y	Y	Y
<input type="checkbox"/>			%	root		Y	Y	Y
<input type="checkbox"/>			localhost			Y	Y	Y
<input type="checkbox"/>			%			N	N	N
<input type="checkbox"/>			localhost	s		N	N	N
<input type="checkbox"/>			localhost	___		N	N	N

Рисунок 3.2 – Задания маски доступа

Следующие поля – user и password, определяют имя и пароль пользователя. Эти поля уже, в отличие от host, различают регистр введенных символов, поэтому пользователь Root и root – это два разных пользователя. Знаки подстановки в этом поле уже не работают – имя пользователя "%" означает не "любой пользователь", а именно "пользователь с именем %". Для задания анонимного пользователя надо просто оставить поле пустым, иначе любой символ будет считаться именем пользователя. так же и в поле password – пароль хранится в зашифрованном виде, знаки подстановки недействительны, а пустое поле означает, что пользователь вовсе не должен указывать пароль, а не то, что пароль может быть произвольным. Напомним, что просто пароль (пусть и зашифрованный) нельзя помещать в поле password, предварительно его надо зашифровать функцией PASSWORD("пароль_открытым_текстом"). Например:

```
INSERT INTO user SET host="192.168.1.138", user="raiden",  
password=PASSWORD("my_password");
```

Следующим полем является поле базы данных – db. В других таблицах оно также присутствует, поэтому во всех таблицах правила написания имен баз данных должны быть одинаковыми. Разрешается использовать символы подстановки "%" и "_", пустое значение соответствует "все базы". Поле чувствительно к регистру.

Далее идет целый набор столбцов, разрешающие те или иные действия для пользователя. К примеру, Select_priv, Insert_priv, Update_priv, Delete_priv, Index_priv и Alter_priv стоит установить в "Y" (разрешено) для обычного пользователя, так как такие запросы чаще всего встречаются в обычных приложениях. Остальные возможности для большей безопасности надо отключить для всех пользователей (особенно, если есть учетная запись анонимного пользователя). Поэтому в следующих столбцах – Create_priv, Drop_priv, Reload_priv, Shutdown_priv, Process_priv, File_priv, Grant_priv и References_priv надо выставить значение "N"(запрещено), а при необходимости разрешить действия только администратору и максимально ограничить возможности подключения такого пользователя – указать конкретный IP-адрес или имя компьютера, выбрать длинный и сложный для подбора пароль.

Аналогична таблице user и таблица db, в которой можно задавать отдельные привилегии для конкретных баз данных. таким образом, можно разрешить пользователям создавать новые таблицы только в своих базах данных и запретить даже простой просмотр чужих баз. Синтаксис всех полей полностью аналогичный таблице user.

При подключении пользователя сервер MySQL сначала обрабатывает таблицу user (записанные там правила доступа глобальные), а потом, если прав недостаточно или нельзя определить права доступа к конкретной базе – тогда уже поиск продолжается в таблице db. Если же и во второй таблице нет соответствующих привилегий, сервер пробует найти их в таблицах tables_priv и columns_priv. И только когда на основании информации со всех доступных таблиц сервер определяет, что клиенту не разрешено запрошенное действие, он отвергает клиентский запрос.

Таблица host также аналогична описанным выше, только вместо имен и паролей пользователей для идентификации привилегий служит имя хоста или IP-адрес. На первый взгляд, есть существенное дублирование информации. Ведь таблицы db и host почти равнозначны и содержат одинаковые данные. На самом деле это немного не так. таблица host недоступна для изменения через операторы SQL GRANT и REVOKE, ее можно только править непосредственно. Когда сервер принимает решение о допуске клиента к базам, он сначала ищет упоминание имени хоста или IP в таблице host. Если там присутствует соответствующая запись, то тогда права из таблиц host и db объединяются и клиент получает некоторые привилегии. Если же в таблице host нет указаний на счет хоста или IP, то сервер даже не рассматривает дальше

таблицу db, так как если клиенту не разрешено подключение с хоста, то тогда бессмысленно искать права на доступ к отдельным базам.

И, напоследок, давайте остановимся на одном нюансе при определении привилегий. Сервер MySQL, когда ищет и сравнивает привилегии, не сравнивает их напрямую. К примеру, символьное обозначение параметра имеет больший приоритет, чем использование символов подстановки. Пример: если есть две записи – `%.hostinfo.ru` и `admin.hostinfo.ru`, то вторая запись имеет больший приоритет. Поэтому если для одного пользователя существуют разные привилегии в зависимости от хоста или других параметров, то будет выбираться первой та запись, где более точно и однозначно указан параметр. Это же касается и IP-адресов.

3.3 Создание базы данных на Wampserver

Для начала необходимо установить Wampserver. Скачать Wampserver можно по адресу <http://www.wampserver.com/ru/> в разделе загрузки. Распространяется он по свободной лицензии, бесплатно и для всех желающих. Скачанный файл – обычный экзешник. Двойной клик и установка началась.

После установки в трее появится значок wampserver. Кликнув по нему правой кнопкой можно выбрать язык представлено на рисунке 3.3.



Рисунок 3.3 – Выбор языка

Далее необходимо кликнуть по localhost и получим всю информацию. Нам необходимо ссылка phpmyadmin в разделе tools и раздел Your Aliases представлено на рисунке 3.4.

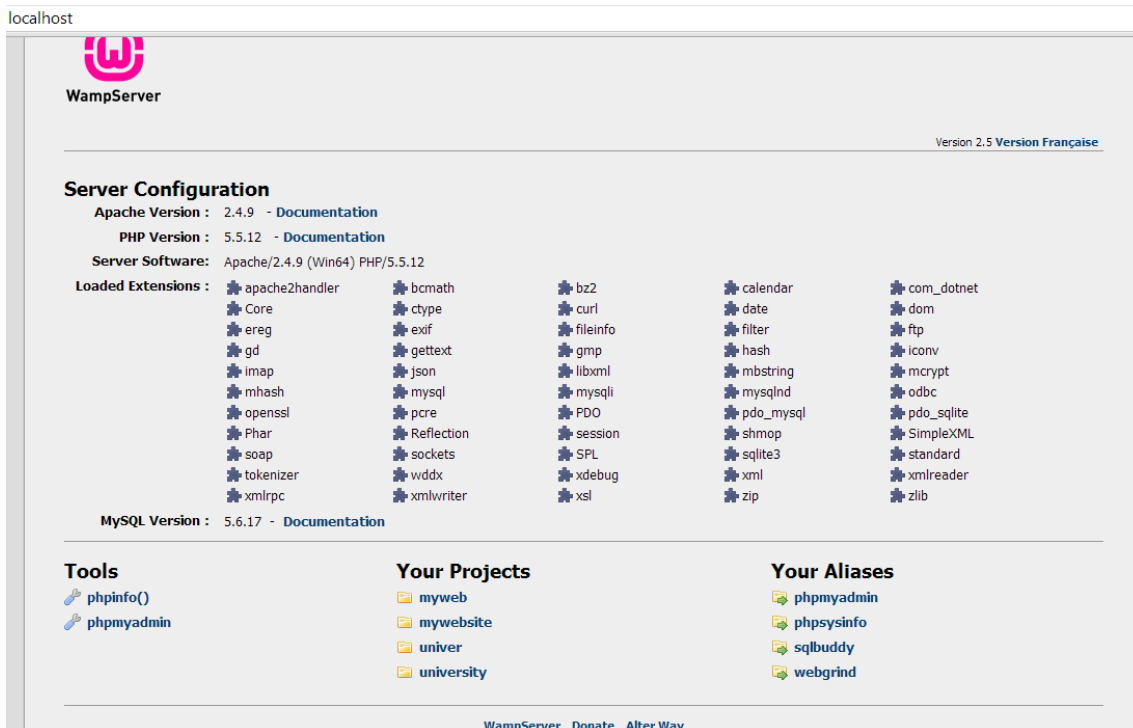


Рисунок 3.4 – Установка базы данных

После того как мы установили сервер создаем базу данных. Но для начала вамп сервере есть папка WWW, создаем в ней папку для хранения всех файлов сайта, в моем случае University. Ниже представлено на рисунке 3.5 создание базы данных.

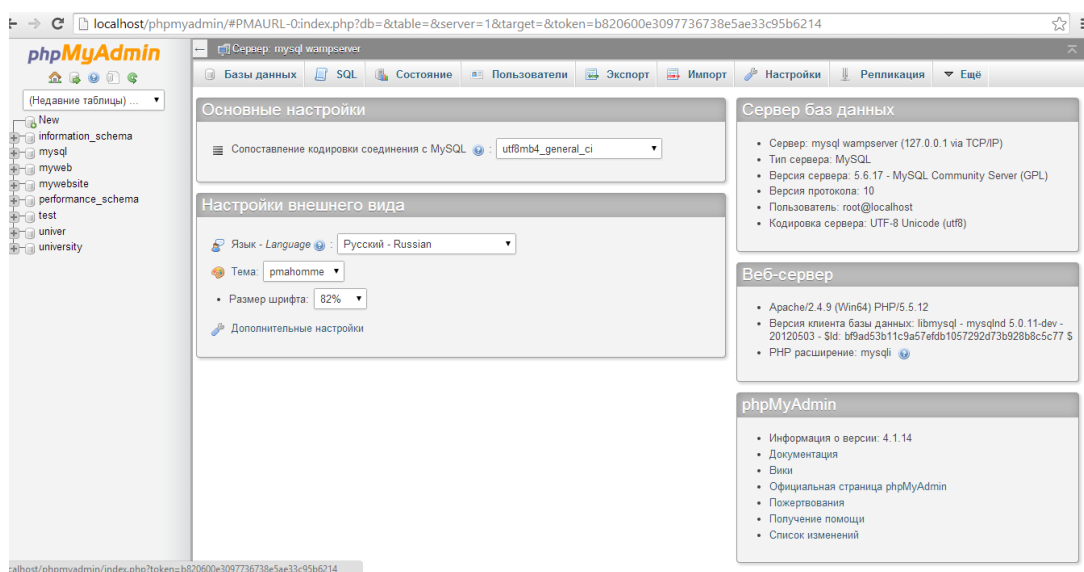


Рисунок 3.5 – Создание базы данных

После создание базы данных создаем приложение и заполняем базу данных всей необходимой информацией.

3.4 Описание программы

Приложение создано на языке PHP с применением HTML, XHTML, таблицы стилей CSS и JavaScript. Структура сайта для пользователя представляет собой пункты меню, информационные вкладки и категории, которая позволяет им перемещаться по страницам сайта для получения нужной информации. В зависимости от прав доступа пользователи получают информацию.

В правом верхнем углу страницы располагается форма входа представлено на рисунке 3.6.

Вход

Логин

Пароль

Запомнить логин

[Забыли логин или пароль?](#)

Рисунок 3.6 – Форма входа

Ниже представлен окно главной страницы сайта "Дистанционного обучения АУЭС" на рисунке 3.7.

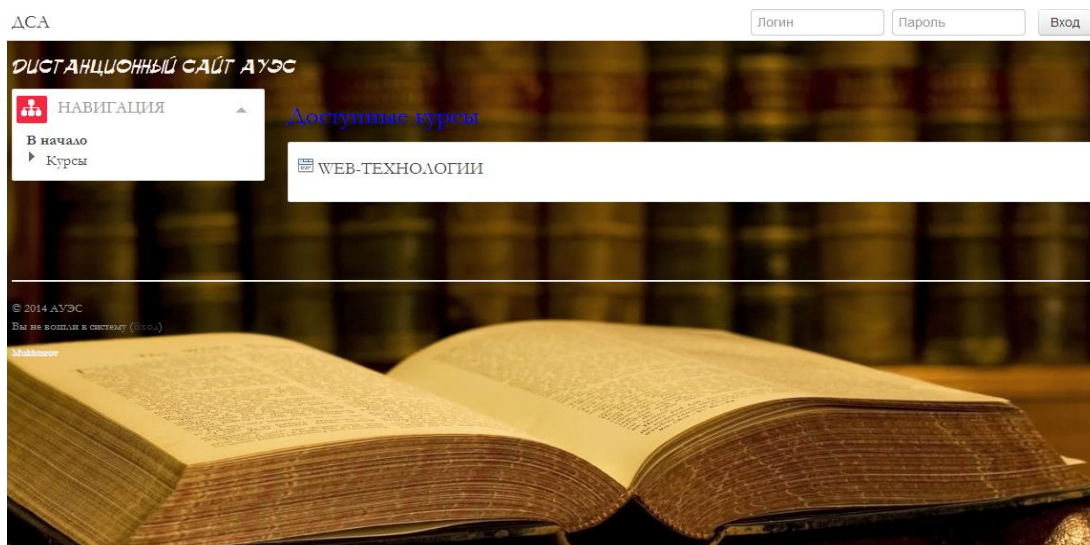


Рисунок 3.7 – Главная страница

Представлено окно при входе под логином студент на рисунке 3.8.

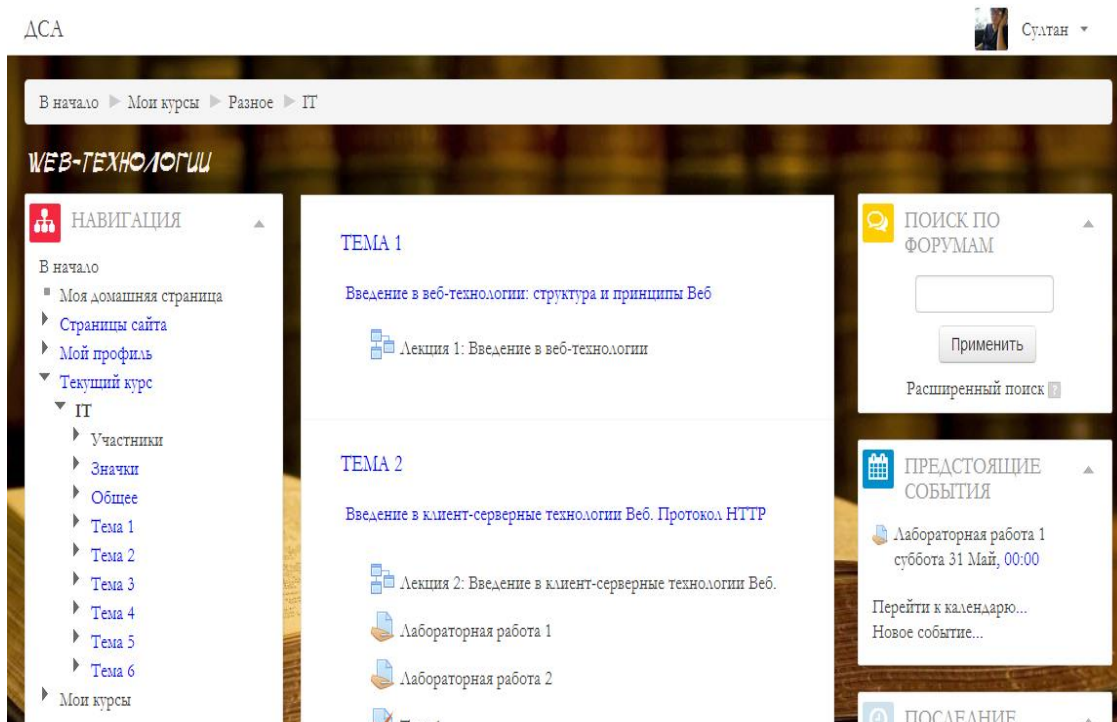


Рисунок 3.8 – Страница студента

При выборе предмета, открывается окно с темами предмета в которых присутствуют лекции, лабораторные работы, тесты, а также итоговая онлайн лекция. В лабораторных работах указаны сроки выполнения, необходимо загружать выполненную лабораторную работу в соответствующий раздел (Рисунок 3.9).

ДСА

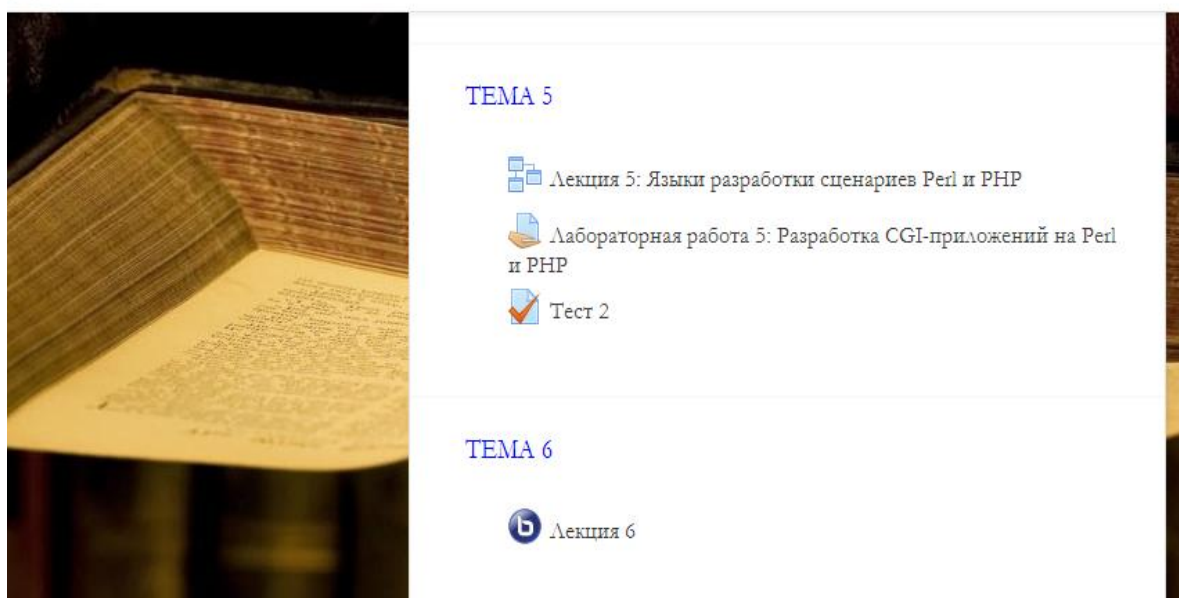


Рисунок 3.9 – Окно текущие предметы

Страница итогового теста по пройденному предмету на рисунке 3.10.

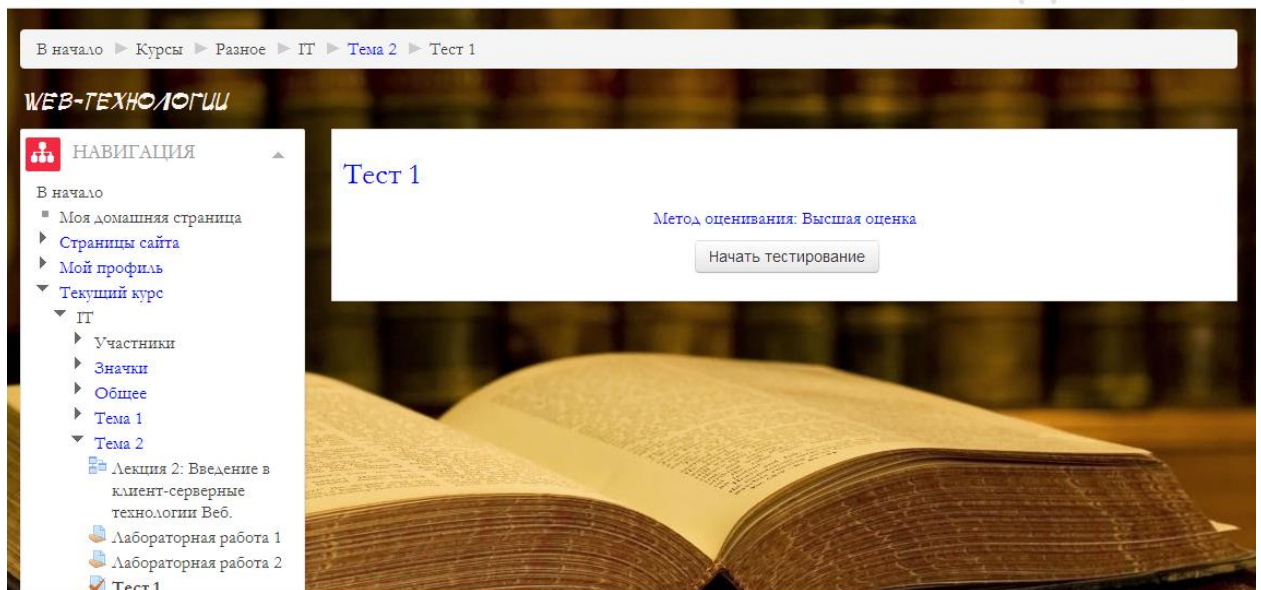


Рисунок 3.10 – Итоговый тест

Окно сдачи итогового теста на рисунке 3.11.

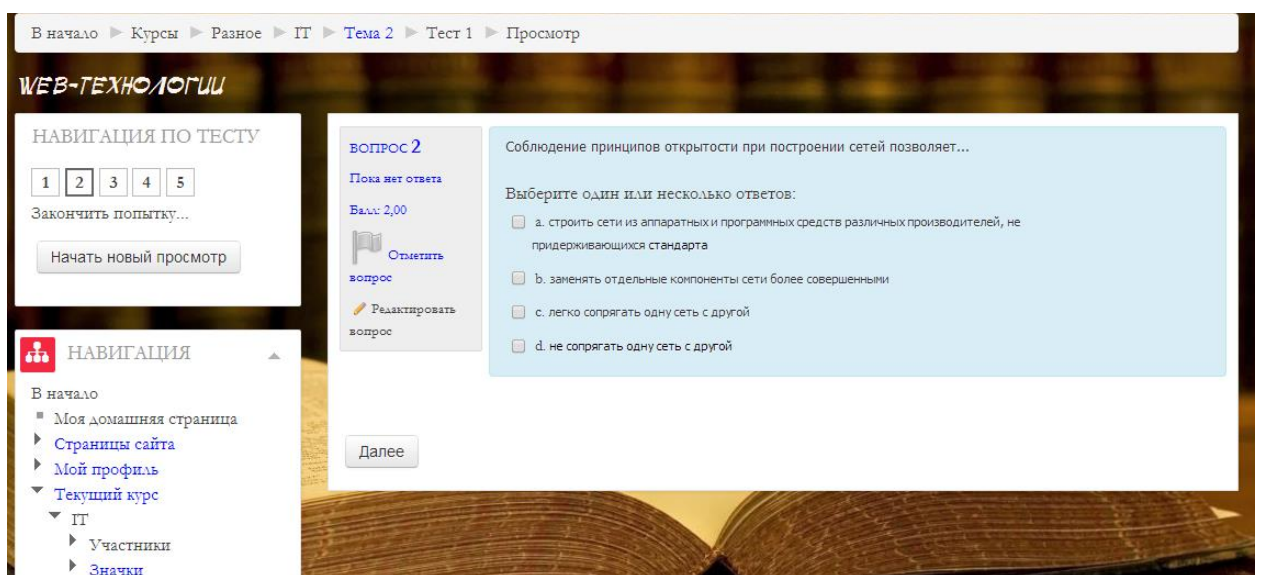


Рисунок 3.11 – Сдача итогового теста

Окно результат попытки, сдачи теста на рисунке 3.12. У студентов есть возможность вернуться к попытке повторной сдачи (2 попытки) или отправить тесты на проверку и завершить тест.

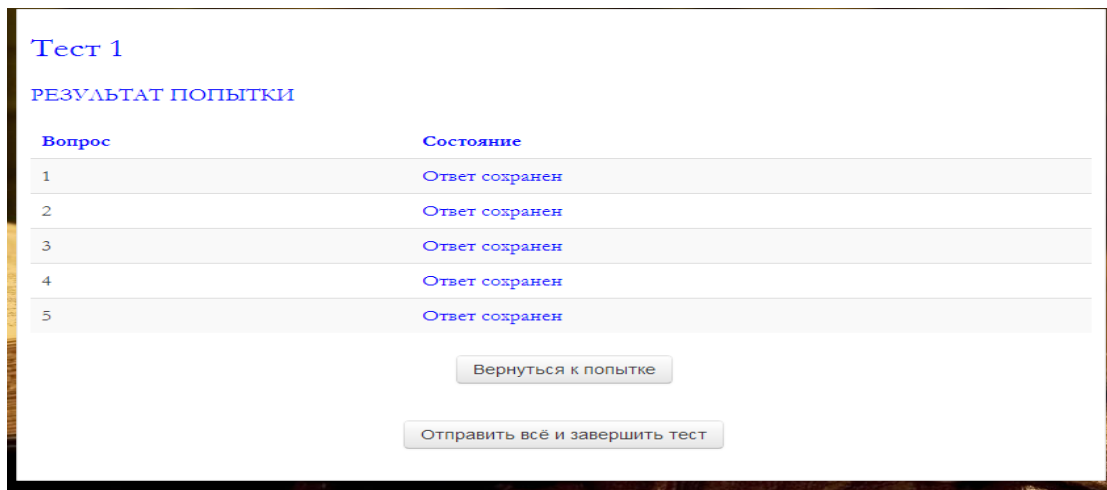


Рисунок 3.12 – Окно результат попытки

Окно просмотр результатов пройденного теста на рисунке 3.13. В окне указано дата назначения теста, дата прохождения, время, оценка и отзыв.

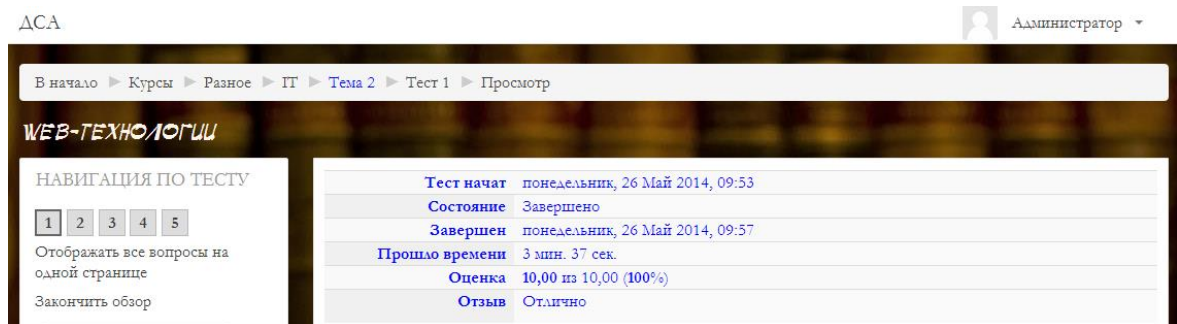


Рисунок 3.13 – Результаты теста

Преподаватель может назначать в домашней странице задания для студентов представлено на рисунке 3.14.

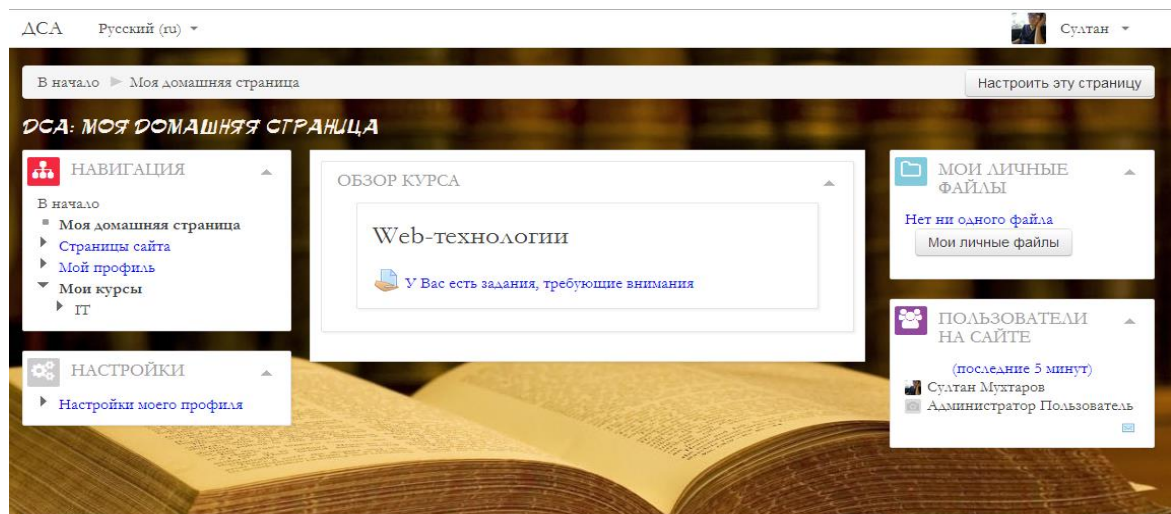


Рисунок 3.14 – Домашняя страница студента

Преподаватель может загружать презентации в итоговую онлайн лекцию представлено на рисунке 3.15 – 3.17 .

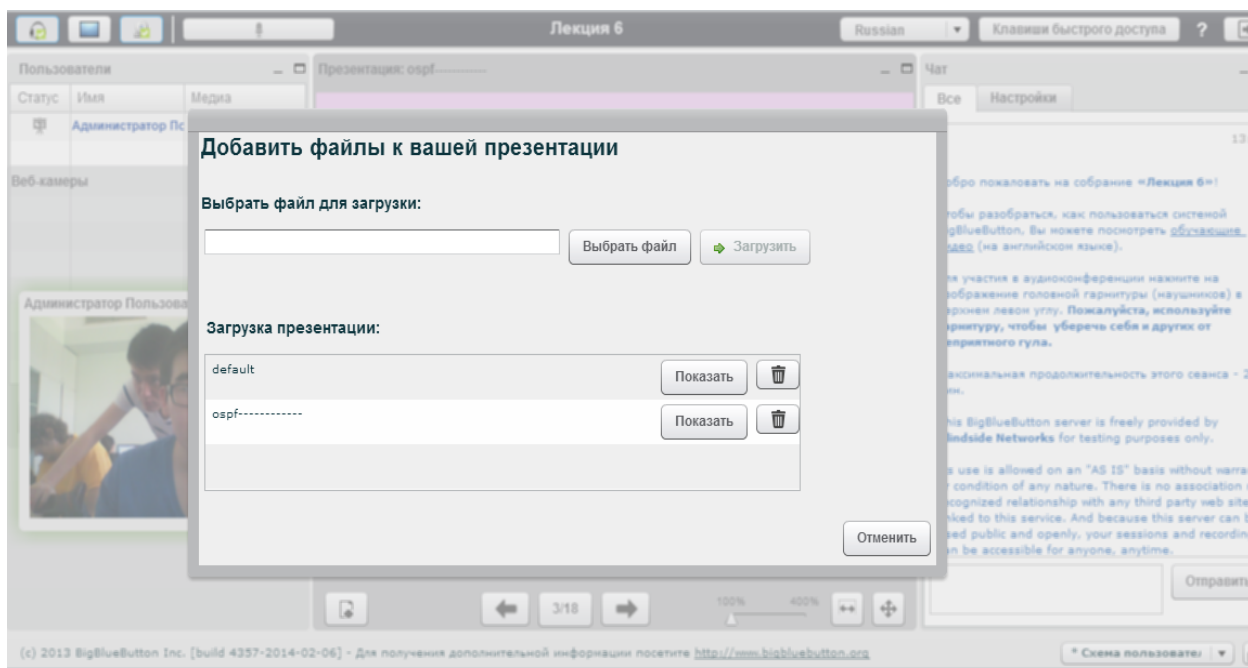


Рисунок 3.15 – Добавления файла в презентацию

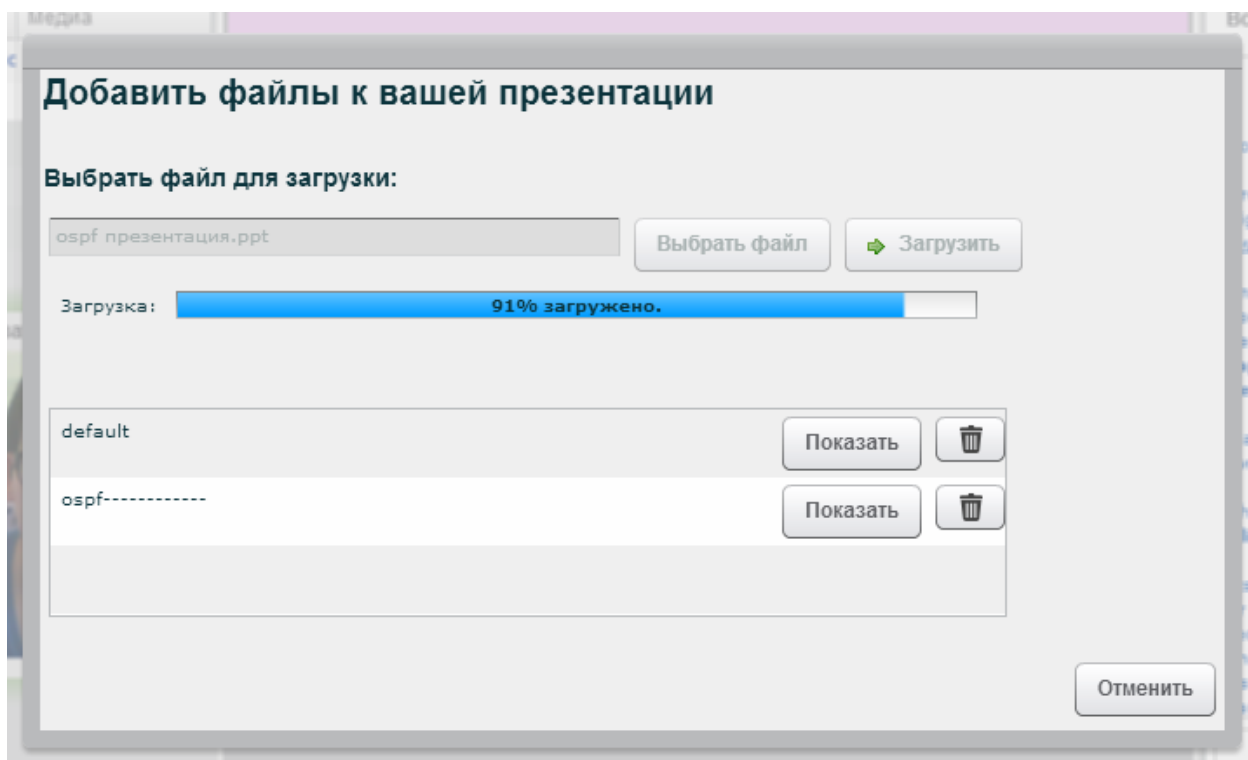
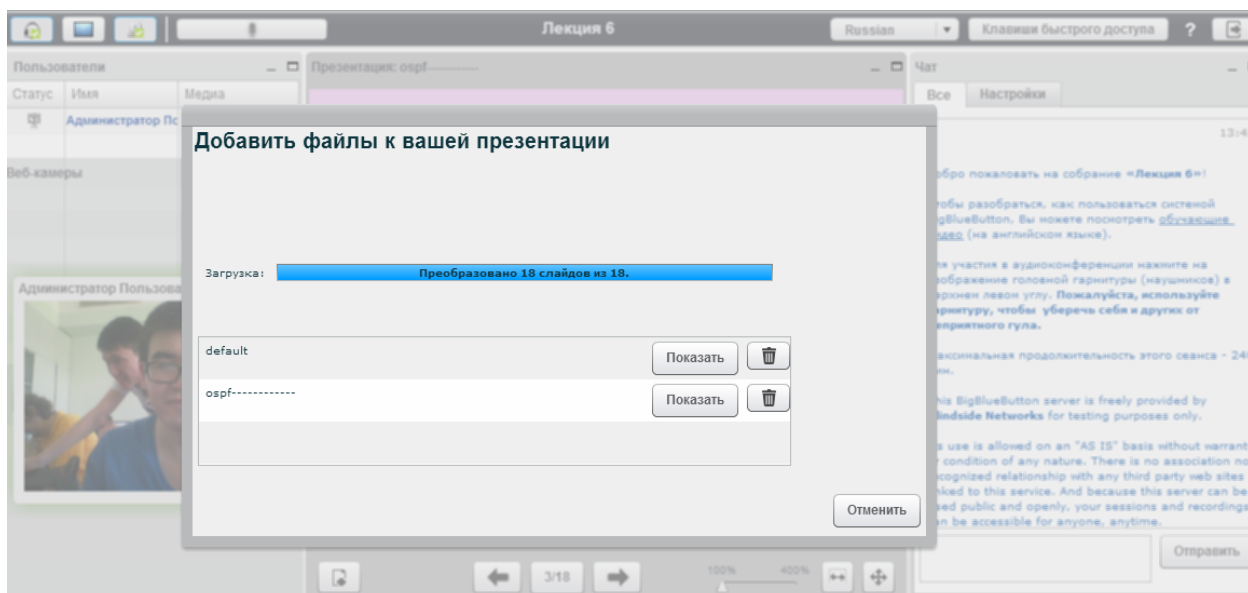


Рисунок 3.16 – Загрузка файла к презентации



Загрузка 3.17 – Форматирование файла в презентацию

Окно итоговая онлайн лекция, в зависимости от группы определяется количество участников онлайн лекции представлено на рисунке 3.18.



Рисунок 3.18 – Итоговая онлайн лекция

При проведении итоговой онлайн лекции, преподаватель имеет возможность выделять, подчеркивать лекции в онлайн режиме представлено на рисунке 3.19.

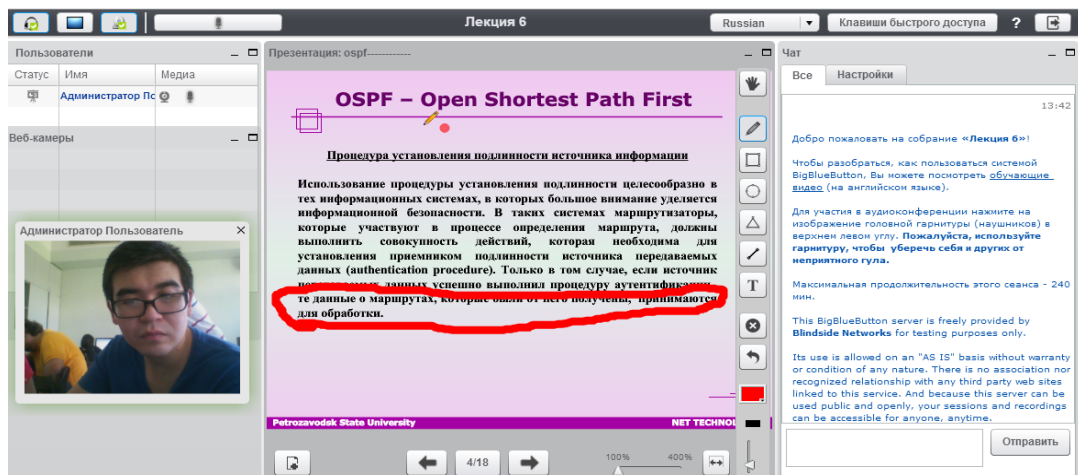


Рисунок 3.19 – Итоговая онлайн лекция

Информация о профиле студента на рисунке 3.20.



Рисунок 3.20 – Профиль студента

Окно редактирования личной информации студента на рисунке 3.21.

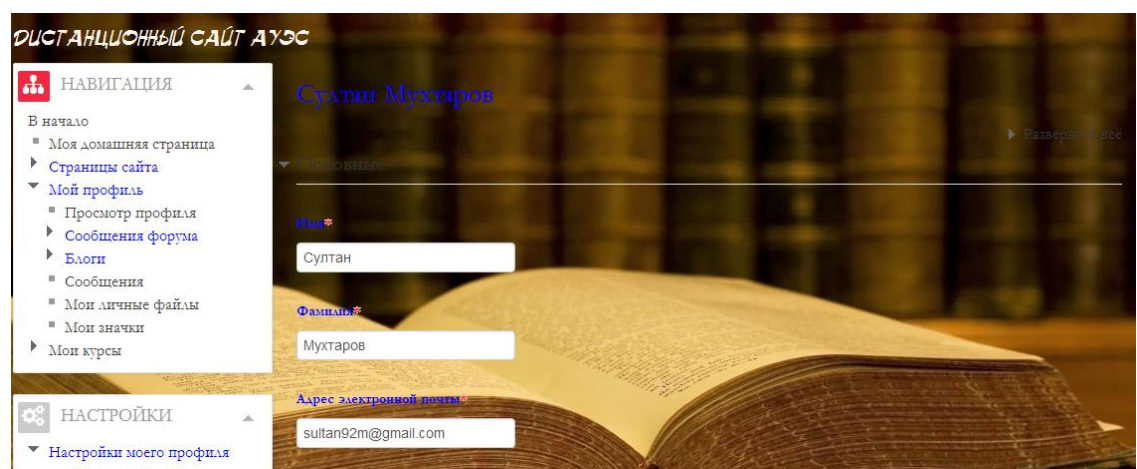


Рисунок 3.21 – Редактирования личной информации

Окно сообщения, пользователи могут отправлять письма друг другу на рисунке 3.22.

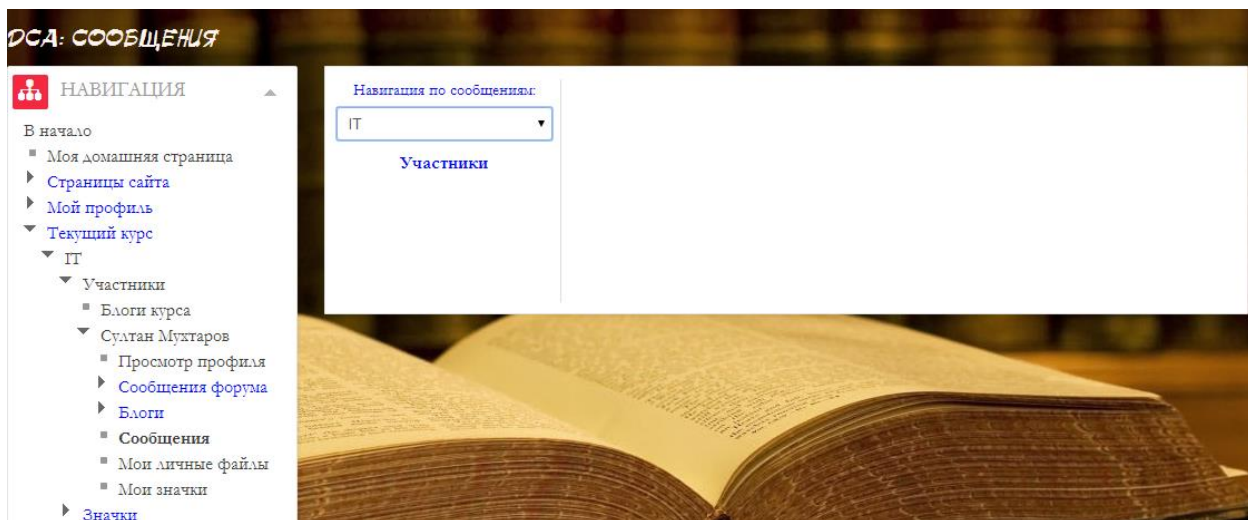


Рисунок 3.22 – Меню сообщения

Окно предоставляющая список пользователей, проходящих данный курс представлено на рисунке 3.23.

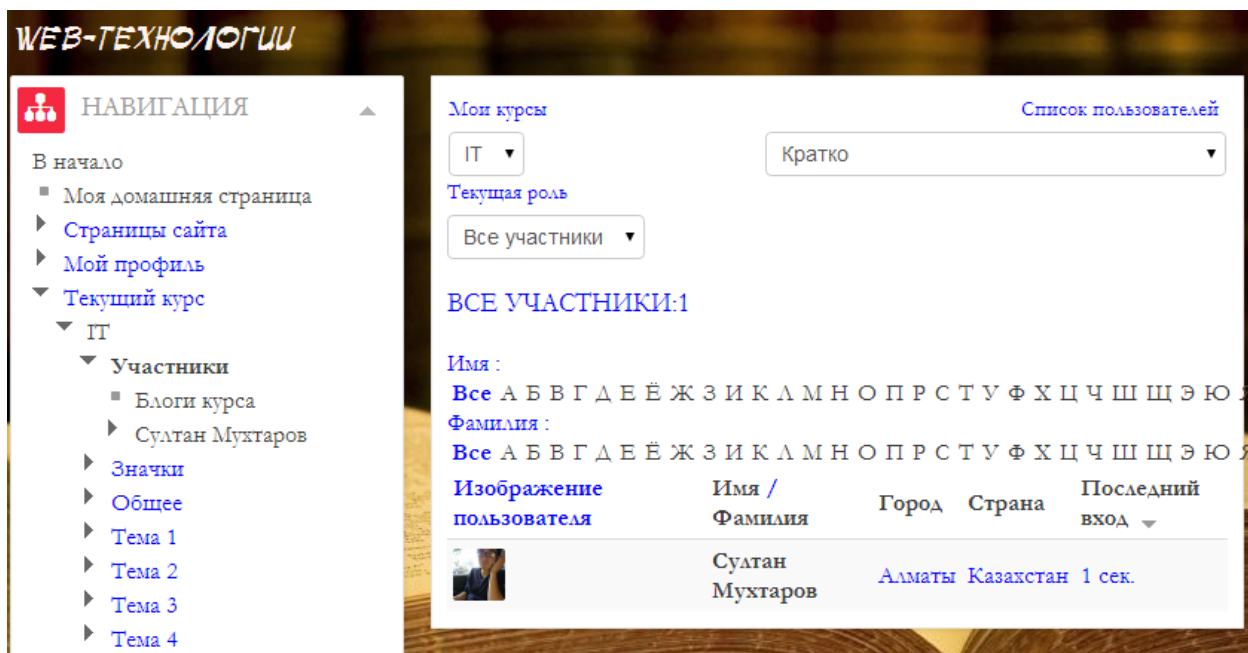


Рисунок 3.23 – Список пользователей

Пользователи могут выполнять при необходимости поиск по форуму, представлено на рисунке 3.24.

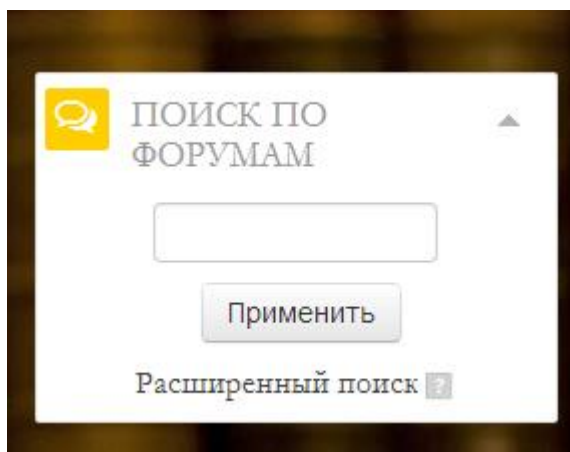


Рисунок 3.24 – Поиск по форуму

В главной странице ежедневно указаны предстоящие события на рисунке 3.25.

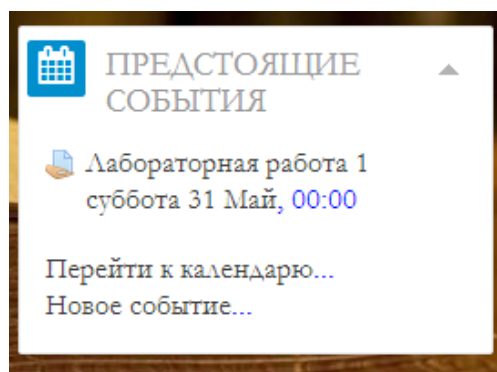


Рисунок 3.25 – Предстоящие события

Пользователей уведомляют на главной странице о последнем его действии на сайте представлено на рисунке 3.26.

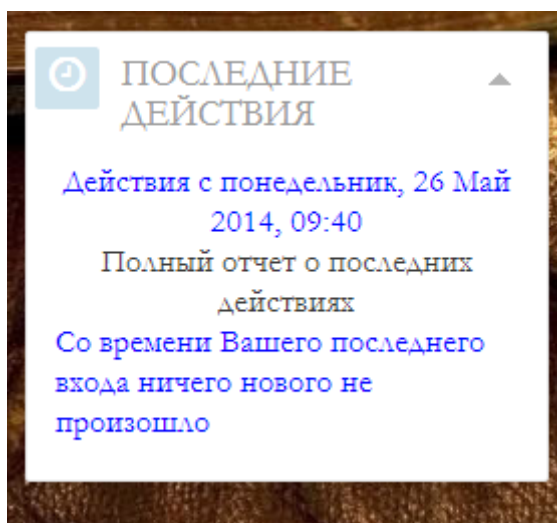


Рисунок 3.26 – Последние действия

Окно отчет по пользователю, которые содержат информацию о пройденных лекциях, лабораторных работах представлено на рисунке 3.27.

Web-технологии				
Лекция 1: Введение в веб-технологии	-	0-100	-	
Лекция 2: Введение в клиент-серверные технологии Веб.	-	0-100	-	
Лабораторная работа 1	-	0-100	-	
Лабораторная работа 2	-	0-100	-	
Лабораторная работа 3	-	0-100	-	
Лабораторная работа 4	-	0-100	-	

Рисунок 3.27 – отчет по пользователю

Окно обзорный отчет содержит название пройденного курса и оценку по данному курсу на рисунке 3.28.

Название курса	Оценка
IT	-

Рисунок 3.28 – Обзорный отчет

Меню настройки содержит пункт управления оценками на рисунке 3.29.

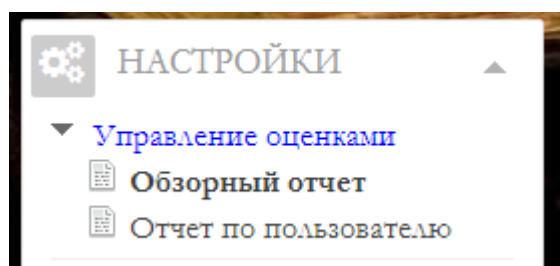


Рисунок 3.29 – Настройка

Окно редактирования блога на рисунке 3.30.

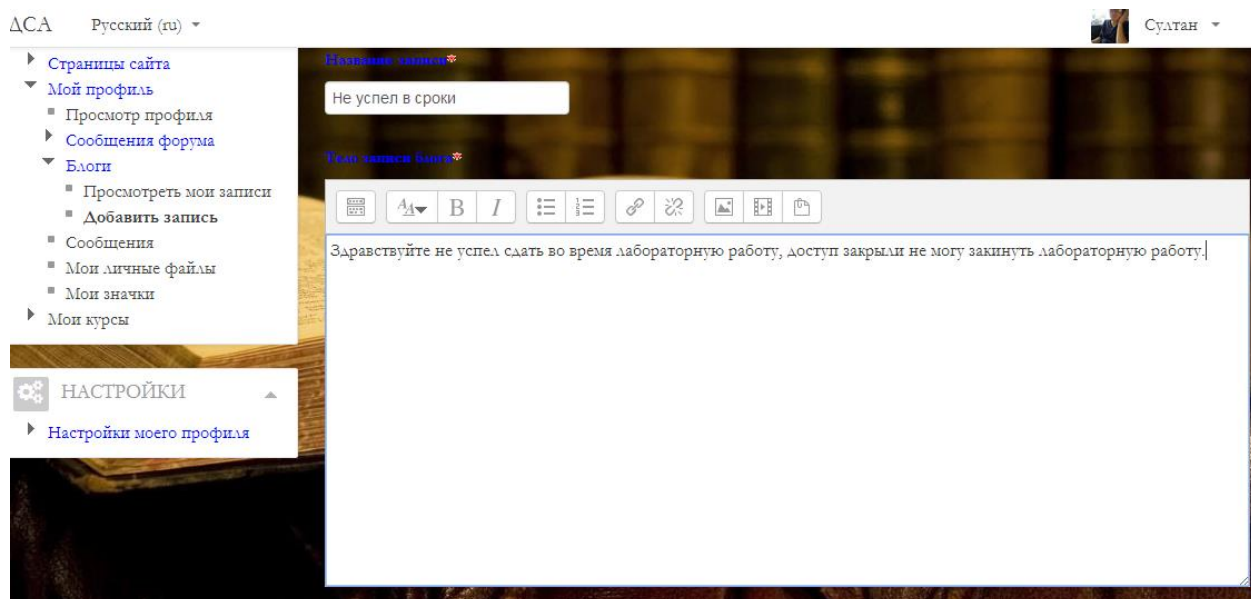


Рисунок 3.30 – Окно блога

Окно блога пользователя на рисунке 3.31.

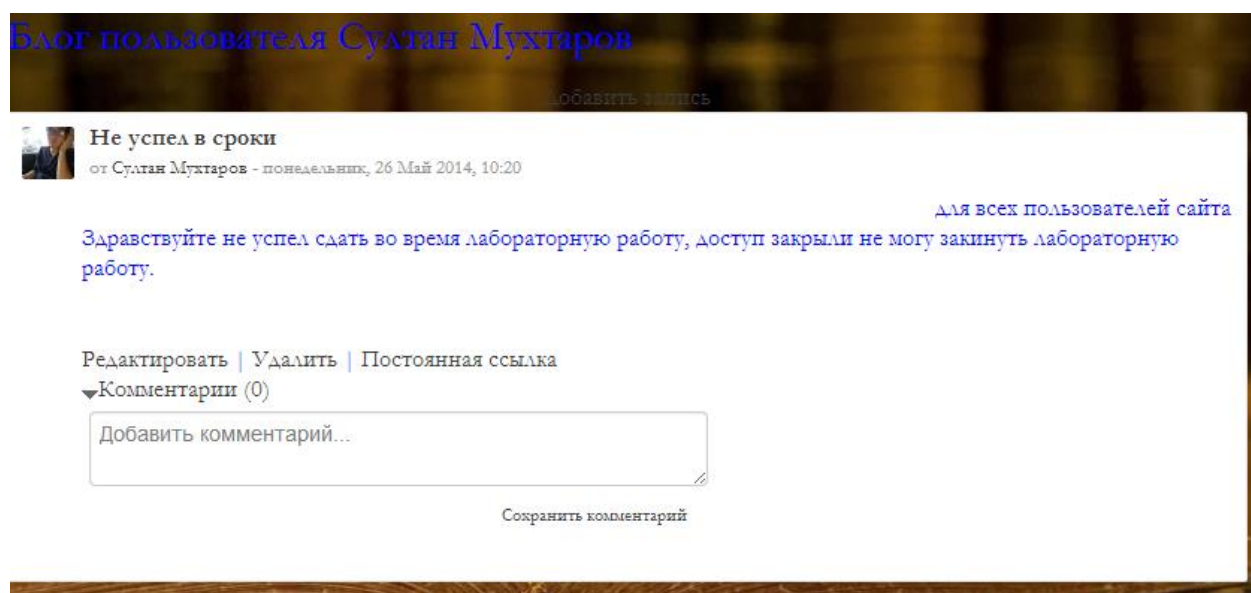


Рисунок 3.31 – Блог пользователя

4 Технико–экономическое обоснование

4.1. Расчет затрат на разработку информационных технологий

Под информационными технологиями понимаются экономические информационные системы (ЭИС), программные продукты (ПП), информационные базы данных и т.д.

Определение объема и трудоемкости разработки программного обеспечения. Общий объем (V_0) программного продукта определяется исходя из количества и объема функции, реализуемых программой

$$V_0 = \sum V_i \quad (4.1)$$

где V_i – объем отдельной функции ПО.

Общий объем равен

$$V_0 = 8\,000$$

На основании информации о функциях разрабатываемого ПО по каталогу функций определяется объем функций и общий объем ПО, который уточняется (корректируется) с учетом условий разработки ПО в организации в соответствии с Приложением Б. Каталог функции программного обеспечения. Уточненный объем ПО (V_y) рассчитывается по формуле

$$V_y = \sum V_{yi} \quad (4.2)$$

где V_{yi} – уточненный объем отдельной функции ПО (ЛОС).

Уточненный объем ПО будет равен

$$V_y = 8\,097$$

Общая трудоемкость небольших проектов рассчитывается по формуле

$$T_o = T_n * K_c * K_m * K_n \quad (4.3)$$

где K_c – коэффициент, учитывающий сложность ПО (Приложение В. дополнительные коэффициенты сложности ПО);

K_t – поправочный коэффициент, учитывающий степень использования при разработке стандартных модулей (Приложении Г. Значения поправочного коэффициента, учитывающего использование стандартных модулей типовых программ и ПО (K_t));

K_n – коэффициент, учитывающий степень новизны ПО (Приложении Д. Поправочные коэффициенты, учитывающие новизну ПО (K_n)).

Поправочный коэффициент рассчитывающий новизну разрабатываемого ПО (K_n) определяется на основе данных представленных в таблице 4.1.

Т а б л и ц а 4.1 – поправочный коэффициент рассчитывающий новизну разрабатываемого ПО

Категория новизны	Степень новизны	Использование		Значение K_n
		На основе нового типа ПК	В среде новой ОС	
Б	Принципиально новые ПО, не имеющие доступных аналогов	–	–	1,0

Коэффициент сложности рассчитывается по формуле

$$K_c = 1 + \sum_{i=1}^n K_i \quad (4.4)$$

где K_i – коэффициент, соответствующий степени повышения сложности ПО за счет конкретной характеристики;

n – количество учитываемых характеристик;

$K_c=0,12$ так как у меня две характеристики.

Поправочный коэффициент, учитывающий степень использования при разработке ПО стандартных модулей (K_m), определяется удельным весом этих модулей в общем объеме проектируемого продукта. В данной работе составляет от 40% до 60, следовательно значение $K_m=0,6$.

Нормативная трудоёмкость (T_n) определяется на основе Укрупненных норм времени на разработку ПО в зависимости от уточненного объема ПО и группы сложности ПО (см. Приложения Б) и составляет 210 (вторая категория сложности, объем ПО 8 000). Следовательно T_o будет равно

$$T_o = 210 * 0,12 * 0,6 * 1,0 = 15,2$$

где $\Phi_{эф}$ – эффективный фонд времени работы одного работника в течение года (дн.);

T_o – общая трудоёмкость разработки проекта (чел./дн.);

T_p – срок разработки проекта (лет).

Срок разработки проекта (T_p) определяется по формуле

$$T_p = T_o / (C_p \times \Phi_{эф}) \quad (4.5)$$

Эффективный фонд времени работы одного работника ($\Phi_{эф}$) рассчитывается по формуле

$$\Phi_{эф} = D_r - D_n - D_v - D_o \quad (4.6)$$

где D_r – количество дней в году;

D_n – количество праздничных дней в году;

D_v – количество выходных дней в году;

D_o – количество дней отпуска.

Так как $D_r = 365$; $D_n = 14$; $D_v = 104$; $D_o = 24$, то по формуле (4.6) получим

$$\Phi_{эф} = 365 - 14 - 104 - 24 = 223 \text{ дней}$$

Плановое число разработчиков $Ч_p = 1$, следовательно по формуле (4.5)

$$T_p = 15,2 / (1 * 223) = 0,07$$

Итого на разработку данного ПО нужно $365 * 0,07 = 26$ дней.

Результаты выполненных расчетов представлены в таблицу 4.2.

Т а б л и ц а 4.2 – Сводные результаты расчета трудоемкости

Название	Условное обозначение	Значение
Общий объем ПО	V_o	8 000
Уточненный объем ПО V_v	V_v	8 097
Нормативная трудоемкость разработки	T_n	210
Общая трудоемкость	T_o	15,2
Эффективный фонд времени работы	$\Phi_{эф}$	223 дня
Срок разработки проекта	T_p	0,07 лет
Численность исполнителей проекта	$Ч_p$	1 чел

Расчет полных затрат на разработку проектного решения в виде информационных технологий ($C_{пi}$) осуществляется по формуле

$$C_{пi} = Z_{фот} + Z_{сзи} + M_i + P_{ci} + P_{mi} + P_{нки} + П_{зи} + P_{ни} \quad (4.7)$$

где $Z_{фот}$ – общий фонд оплаты труда разработчиков, тенге;

$Z_{сзи}$ – отчисления по социальному налогу, тенге;

M_i – затраты на материалы, тенге;

P_{ci} – затраты на специальные программные средства, необходимые для разработки проектного решения, тенге;

P_{mi} – затраты, связанные с эксплуатацией техники, тенге;

$P_{нки}$ – затраты на научные командировки, тенге;

$П_{зи}$ – прочие затраты, тенге;

$P_{ни}$ – накладные расходы, тенге.

Размер фонда оплаты труда разработчиков ($Z_{\text{ФОТ}}$) рассчитывается по формуле

$$Z_{\text{ФОТ}} = Z_{\text{oi}} + Z_{\text{di}} \quad (4.8)$$

где Z_{oi} – основная заработная плата, тенге;

Z_{di} – дополнительная заработная плата, тенге.

Основная заработная плата исполнителей на конкретное ПО рассчитывается по формуле

$$Z_{\text{oi}} = \sum T_{\text{чи}} \times T_{\text{ч}} \times \Phi_{\text{п}} \times K \quad (4.9)$$

где n – количество исполнителей, занятых разработкой конкретного ПО;

$T_{\text{чи}}$ – часовая тарифная ставка i -го исполнителя (тыс.тенге);

$\Phi_{\text{п}}$ – плановый фонд рабочего времени i -го исполнителя (дней) = 22 раб день в месяц;

$T_{\text{ч}}$ – количество часов работы в день (час) = 8 часов;

K – коэффициент премирования = 1,25.

По данным о специфике и сложности выполняемых функций составляется штатное расписание группы специалистов–исполнителей, участвующих в разработке ПО, с определением образования, специальности, квалификации и должности представленно в таблице 4.3.

Т а б л и ц а 4.3 – Сведения по работникам, задействованным в проекте

Специалист – Исполнитель	Количество, человек	Зарботная плата в месяц, тенге
Программист	1	100 000
Итого		100 000

Часовая тарифная ставка рассчитывается путем деления месячной тарифной ставки на установленную при 40–часовой недельной норме рабочего времени расчетную среднемесячную норму рабочего времени в часах ($\Phi_{\text{р}}$)

$$T_{\text{ч}} = T_{\text{м}} / \Phi_{\text{р}} \quad (4.10)$$

где $T_{\text{ч}}$ – часовая тарифная ставка (тыс.тенге);

$T_{\text{м}}$ – месячная тарифная ставка (тыс.тенге).

По формуле (4.10) можно определить часовую тарифную ставку исполнителя

$$T_{\text{ч}} = 100\,000 / 208 = 480,77 \text{ тенге в час}$$

По формуле (4.9) можно рассчитать основную заработную плату исполнителей

$$З_{oi} = 480,77 * 8 * 26 * 1,25 = 125\ 000 \text{ тенге}$$

Дополнительная заработная плата составляет 10% от основной заработной платы и рассчитывается по формуле

$$З_{di} = З_{oi} * H_d / 100 \quad (4.11)$$

где H_d – коэффициент дополнительной заработной платы разработчиков.

Дополнительная заработная плата равна

$$З_{di} = 125\ 000 * 0,1 = 12\ 500 \text{ тенге}$$

$$З_{фот} = 125\ 000 + 12\ 500 = 137\ 500 \text{ тенге}$$

Социальный налог составляет 11% (ст. 358 п. 1 НК РК) от дохода работника, и рассчитывается по формуле

$$З_{си} = (ФОТ - ПО) * 11\% \quad (4.12)$$

где ПО – пенсионные отчисления, которые составляют 10% от ФОТ социальным налогом не облагаются

$$ПО = ФОТ * 10\% \quad (4.13)$$

По формулам (4.12) и (4.13) получается

$$ПО = 125\ 000 * 0,1 = 12\ 500 \text{ тенге}$$

$$З_{си} = (125\ 000 - 12\ 500) * 0,11 = 12\ 375 \text{ тенге}$$

Величина затрат на материалы на основании исходных данных определяется по формуле

$$M_i = (З_{очн} * H_{мз}) / 100\% \quad (4.14)$$

где $H_{мз}$ – норма расхода материалов от основной заработной платы (3–5%).

$$M_i = 125\ 000 * 0,045 = 5625 \text{ тенге}$$

Расходы по статье «Спецоборудование» (P_{ci}) включают затраты средств на приобретение вспомогательных специального назначения технических и

программных средств, необходимых для разработки конкретного ПО, включая расходы на их проектирование, изготовление, отладку, установку и эксплуатацию

$$P_{ci} = \sum C_{ci} \quad (4.15)$$

где C_{ci} – стоимость конкретного специального оборудования (тыс.тенге);
 n – количество применяемого специального оборудования.

Затраты средств на приобретение вспомогательных специального назначения технических и программных средств указана в таблице 4.4.

Т а б л и ц а 4.4 – Затраты средств на приобретение вспомогательных программных средств

Наименование	Цена, тенге
Dreamveawer	86 000
Windows 8	25 000
Kaspersky antivirus	15 000
Итого:	126 000

$$P_{ci} = 126\ 000 \text{ тенге}$$

Расходы по статье «Машинное время» (P_{mi}) включают оплату машинного времени, необходимого для разработки и отладки ПО, которое определяется по нормативам (в машино–часах) на 100 строк исходного кода (H_{mb}) машинного времени в зависимости от характера решаемых задач и типа ПК

$$P_{mi} = C_{mi} \times (V_{oi}/100) \times H_{mb} \quad (4.16)$$

где C_{mi} – цена одного машино–часа (тыс.тенге);

V_{oi} – общий объем ПО (строк исходного кода);

H_{mb} – норматив расхода машинного времени на отладку 100 строк исходного кода (машино–часов) = 12. ч/100 строк кода в соответствии с Приложением Б.

$$P_{mi} = 480,76 * (8\ 000 / 100) * 12 = 461\ 538,5 \text{ тенге}$$

Расходы по статье «Научные командировки» (P_{nki}) на конкретное ПО определяются по нормативу, разрабатываемому в целом по организации, в процентах к основной заработной плате

$$P_{nki} = Z_{oi} \times H_{pнк} / 100 \quad (4.17)$$

где $H_{pнк}$ – норматив расходов на командировки в целом по организации в (%), в дипломной работе нужно брать 30% .

Подставляем все данные в формулу (4.17) получаем

$$P_{\text{нкi}} = 125\,000 * 0,3 = 37\,500 \text{ тенге}$$

Расходы по статье «Прочие затраты» (P_{3i}) на конкретное ПО включают затраты на приобретение и подготовку специальной научно–технической информации и специальной литературы. Определяются по нормативу, разрабатываемому в целом по организации, в процентах к основной заработной плате

$$P_{3i} = Z_{oi} \times N_{пз} / 100 \quad (4.18)$$

где $N_{пз}$ – норматив прочих затрат в целом по организации в (%), в дипломной работе нужно брать 20% .

Подставляем все данные в формулу (4.18) получаем

$$P_{3i} = 125\,000 * 0,2 = 25\,000 \text{ тенге}$$

Затраты по статье «Накладные расходы» ($P_{\text{ни}}$), связанные с необходимостью содержания аппарата управления, вспомогательных хозяйств и опытных (экспериментальных) производств, а также с расходами на общехозяйственные нужды ($P_{\text{ни}}$), относятся на конкретное ПО по нормативу ($N_{\text{рн}}$) в процентном отношении к основной заработной плате исполнителей. Норматив устанавливается в целом по организации

$$P_{\text{ни}} = Z_{oi} \times N_{\text{рн}} / 100\% \quad (4.19)$$

где $N_{\text{рн}}$ – норматив накладных расходов в целом по организации в (%), в дипломной работе нужно брать 70% .

Подставляем все данные в формулу (4.19) получаем

$$P_{\text{ни}} = 125\,000 * 0,7 = 87\,500 \text{ тенге}$$

Результаты выполненных расчетов представлены в таблицу 4.5.

Т а б л и ц а 4.5 – Затраты на разработку

Затраты на разработку	Условное обозначение	Значение, тенге	В процентах от общей суммы
Фонд оплаты труда	$Z_{\text{фот}}$	137 500	15,4
Социальный налог	$Z_{\text{сзи}}$	12 375	1,4
Материалы	M_i	5 625	0,6
Спецоборудование	P_{ci}	126 000	14,1

Затраты на разработку	Условное обозначение	Значение, тенге	В процентах от общей суммы
Машинное время	P_{mi}	461 538,5	51,7
Научные командировки	P_{nki}	37 500	4,2
Прочие затраты	P_{zi}	25 000	2,8
Накладные расходы	P_{ni}	87 500	9,8
Итого:		893 038,5	100

Отсюда себестоимость равна

$$C_{pi} = 893\,038,5 \text{ тенге}$$

Структура затрат на разработку представлена на диаграмме 4.1.

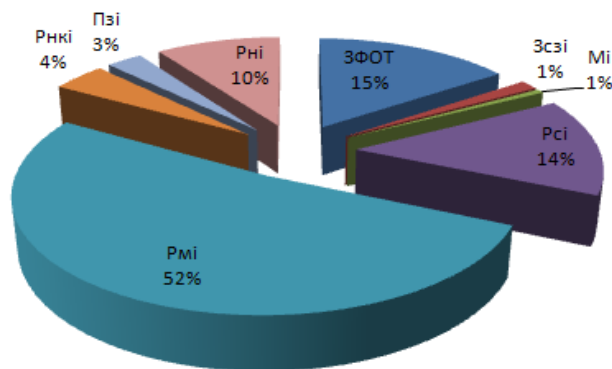


Диаграмма 4.1 – Затраты на разработку

Рентабельность и прибыль по создаваемому ПО (P_{ci}) определяются исходя из результатов анализа рыночных условий, переговоров с заказчиком (потребителем) и согласования с ним отпускной цены, включающей дополнительно налог на добавленную стоимость. В случае разработки ПО для использования внутри организации оценка программного продукта производится по действующим правилам и показателям внутреннего хозрасчета (по ценам, устанавливаемым для расчета за услуги между подразделениями). Прибыль рассчитывается по формуле

$$P_{oi} = C_{pi} \times Y_{rpi} / 100 \quad (4.20)$$

где P_{oi} – прибыль от реализации ПО заказчику (тыс.тенге);

Y_{rpi} – уровень рентабельности ПО (%) = 51%;

C_{pi} – себестоимость ПО (тыс.тенге).

Подставим данные в формулу (4.20) получим

$$P_{oi} = 893\,038,5 * 0,51 = 455\,449,6 \text{ тенге}$$

Прогнозируемая цена ПО без налогов (C_{pi})

$$C_{pi} = C_{pi} + P_{oi} \quad (4.21)$$

Зная формулу (4.21) получается

$$C_{pi} = 893\,038,5 + 455\,449,6 = 1\,348\,488,1 \text{ тенге}$$

Прогнозируемая отпускная цена (C_{oi})

$$C_{oi} = C_{pi} + \text{НДС} \quad (4.22)$$

Зная формулу (4.22) получается

$$C_{oi} = 1\,348\,488,1 * (1 + 0,12) = 1\,510\,306,6 \text{ тенге}$$

Ставка налога на добавленную стоимость НДС в РК на 2014 год составляет 12% от отпускной цены ПО.

Затраты на освоение определяются по нормативу ($H_o = 10\%$) от себестоимости ПО в расчете на 3 месяца и рассчитываются по формуле

$$P_{oi} = C_{pi} * H_o / 100\% \quad (4.23)$$

Исходя из формулы (4.23) получим

$$P_{oi} = 893\,038,5 * 0,1 = 89\,303,8 \text{ тенге}$$

Затраты на сопровождение ПО (P_{ci}). Организация–разработчик осуществляет сопровождение ПО и несет соответствующие расходы, которые оплачиваются заказчиком в соответствии с договором и сметой на сопровождение. Затраты на сопровождение определяются по установленному нормативу ($H_c = 20\%$) от себестоимости ПО (в расчете на год) и рассчитываются по формуле

$$P_{ci} = C_{pi} * H_c / 100\% \quad (4.24)$$

По формуле (4.24) получаем

$$P_{ci} = 893\,038,5 * 0,2 = 178\,607,7 \text{ тенге}$$

4.2 Расчет цены программного продукта

Расчет цены ПП в организационно–экономической части дипломной работы предлагается производить следующим образом:

- если ПП разработан одной организацией по заказу другой и не предназначен для тиражирования, то затраты на разработку ПП считаются его себестоимостью, и при формировании цены применяется затратный метод;
- если ПП предназначен для тиражирования, то конечная цена определяется путем экспертных оценок на основании ценностного подхода с учетом текущих цен конкурентов (если существуют аналогичные ПП).

Расчет цены ПП, который разработан одной организацией по заказу другой и не предназначен для тиражирования, осуществляется по формуле

$$Ц_{ПП} = З_{РПР} + П_{п} + НДС \quad (4.25)$$

где $Ц_{ПП}$ – цена программного продукта, тенге;

$З_{РПР}$ – затраты на разработку проектного решения, в данном случае программного продукта, тенге;

$П_{п}$ – планируемая прибыль, тенге;

НДС – налог на добавленную стоимость, тенге.

Планируемая прибыль рассчитывается по формуле

$$П_{п} = З_{РПР} * R_{НПП} \quad (4.26)$$

где $R_{НПП}$ – нормативная рентабельность ПП, определяемая организацией.

НДС, начисленный на ПП, определяется следующим образом

$$НДС = (З_{РПР} + П_{п}) * k_{НДС} \quad (4.27)$$

где $k_{НДС}$ – ставка налога на добавленную стоимость.

По формуле (4.26) учитывая, что $З_{РПР} = C_{пi}$, $R_{НПП}$ – это процент рентабельности по отношению к себестоимости составляет 22%, отсюда получаем

$$П_{п} = C_{пi} * 0,22 = 893\,038,5 * 0,22 = 196\,468,5 \text{ тенге}$$

По формуле (4.27) получаем

$$НДС = (893\,038,5 + 196\,468,5) * 0,12 = 130\,740,8 \text{ тенге}$$

По формуле (4.25) получаем

$$Ц_{ПП} = 893\,038,5 + 196\,468,5 + 130\,740,8 = 1\,220\,247,8 \text{ тенге}$$

4.3 Расчет эксплуатационных затрат при использовании информационных технологий

Важное значение при внедрении информационных технологий имеет изменение величины текущих эксплуатационных затрат при производстве продукции, услуг или работ. Во внимание должны приниматься только те статьи затрат, которые сократятся или вырастут в результате внедрения ИТ.

Годовые эксплуатационные текущие затраты в условиях функционирования информационных технологий (С) рассчитываются по формуле

$$C = ЗП + ОТ + А + М + НР \quad (4.28)$$

где ЗП – годовые затраты на оплату труда специалистов при выполнении ими своих функций в рамках автоматизируемого процесса после внедрения ИТ, тенге;

ОТ – отчисления по социальному налогу, тенге;

А – затраты на амортизацию, тенге;

М – годовые материальные затраты на сопровождение программного продукта, тенге;

НР – накладные расходы, тенге.

Годовые затраты по заработной плате специалистов после внедрения ИТ определяются по формуле

$$ЗП = O_c * Ч_c * 12 / \Phi_{р.в.} * t_{общ} * 12 * (1 + K_d) \quad (4.29)$$

где O_c – оклад специалиста, тенге./мес.;

$Ч_c$ – численность специалистов, участвующих в процессе, чел.;

$\Phi_{р.в.}$ – годовой фонд рабочего времени, час;

$t_{общ}$ – трудоемкость решения задач в условиях функционирования ИТ в месяц, час;

K_d – коэффициент дополнительной заработной платы.

Трудоемкость решения задач в условиях функционирования ИТ в месяц вычисляется следующим образом

$$t_{общ} = \sum_{\beta=1}^n t_{\beta} \times K_{\beta} \quad (4.30)$$

где t_{β} – затраты времени на решение β -й задачи, час;

K_{β} – количество решаемых β -х задач в месяц, ед.

Трудоемкость решения задач в условиях функционирования ИТ в месяц по формуле (4.30) будет составлять

$$t_{\text{общ}} = (3*20) + (2*8)*6 = 156 \text{ часов}$$

Опираясь на среднюю заработную плату аналогичных систем дистанционного обучения заработная плата будет равна 50 000 тенге. Предполагаемое количество сотрудников будут состоят из одного администратора и 6 учителей, в общем сотрудников будет 7 человек. Годовой фонд рабочего времени равен 223 дня, в часах 1 784 часа. Коэффициент дополнительной платы равен 10%.

Годовые затраты по заработной плате специалистов после внедрения ИТ определяющиеся по формуле (4.29) будут равны

$$ЗП = (50\,000 * 7 * 12) / 1\,784 * 156 * 12 * (1+0,1) = 4\,847\,892 \text{ тенге}$$

Так как у нас 7 специалистов то основная зарплата будет 350 000 тенге. Отчисления по социальному налогу рассчитывается по формуле (4.12)

$$ОТ = (350\,000 - 35\,000) * 0,11 = 34\,650 \text{ тенге}$$

Амортизационные отчисления будут равны

$$A = Ц_{\text{ПП}} \times H_a \quad (4.31)$$

где H_a – норма амортизации (25 %);

$Ц_{\text{ПП}}$ – цена программного продукта, тенге.

Амортизационные отчисления равны

$$A = 1\,220\,247,8 * 0,25 = 305\,062 \text{ тенге}$$

Годовые затраты на сопровождение программного продукта будут равны стоимости хостинга сайта 25 000 тенге.

Накладные расходы рассчитываются по формуле (4.19)

$$P_{\text{нi}} = 350\,000 * 0,7 = 245\,000 \text{ тенге}$$

Таким образом годовые эксплуатационные текущие затраты в условиях функционирования информационных технологий

$$C = 4\,847\,892 + 34\,650 + 305\,062 + 25\,000 + 245\,000 = 5\,457\,604 \text{ тенге}$$

Результаты выполненных расчетов представлены в таблице 4.6.

Т а б л и ц а 4.6 – Текущие эксплуатационные затраты

Затраты на разработку	Условное обозначение	Значение, тенге	В процентах от общей суммы
Заработная плата	ЗП	4 847 892	88,8
Социальный налог	ОТ	34 650	0,6
Амортизационные отчисления	А	305 062	5,6
Материалы	М	25 000	0,5
Накладные расходы	НР	245 000	4,5
Итого:		5 457 604	100

4.4 Расчет основных показателей экономической эффективности

Оценка экономической эффективности внедрения системы дистанционного обучения производится на основе коэффициента абсолютной экономической эффективности и рассчитывается по формуле

$$E_p = \frac{\Pi}{K} \quad (4.32)$$

где E_p – расчетный коэффициент экономической эффективности капитальных вложений;

Π – ожидаемая прибыль, тенге;

K – капитальные вложения на создание системы, тенге.

Прибыль за год рассчитывается по формуле

$$\Pi = Д - И \quad (4.33)$$

где $Д$ – доход университета;

$И$ – издержка (эксплуатационные расходы).

Доходы от предоставления услуг дистанционного обучения рассчитываются по формуле

$$Д = N * P * (1 + k) \quad (4.34)$$

где N – количество обучающихся;

P – оплата за обучение;

k – коэффициент добавочной стоимости от оплаты обучения за использования сайта = 0,05.

Плата за обучения проводится по семестрам, стоимость обучения 80 000 тенге. За год оплата составляет 160 000 тысяч тенге. Планируемое количество обучающихся 37 студентов, доход составит

$$Д = 39 * 160\,000 * 1,05 = 6\,216\,000 \text{ тенге}$$

Прибыль согласно формуле (4.33) будет равна

$$\Pi = 6\,216\,000 - 5\,457\,604 = 758\,396 \text{ тенге}$$

Таким образом, согласно по произведенным расчетам коэффициент абсолютной экономической эффективности от реализации проекта составит

$$E_p = \frac{758\,396}{893\,038,5} = 0,85$$

Расчетный срок окупаемости капитальных вложений составляет

$$T_p = \frac{1}{E_p}, \quad (4.34)$$

где E_p – коэффициент экономической эффективности капитальных вложений.

Ниже рассчитан срок окупаемости капитальных вложений

$$T_p = \frac{1}{0,85} = 1,18 \text{ лет}$$

Результаты выполненных расчетов представлены в таблице 4.7.

Т а б л и ц а 4.7 – Показатели абсолютной экономической эффективности от внедрения системы дистанционного обучения

Наименование показателей	Значение
Ожидаемая прибыль, тенге	758 396
Коэффициент экономической эффективности капитальных вложений (E_p)	0,85
Срок окупаемости капитальных вложений (T_p), лет	1,18 лет

Вывод

Суммарные затраты на разработку программного продукта составляют 893 038 тенге, цена реализации – 1 220 247 тенге с учетом НДС. Основную долю в себестоимости программного продукта составляют машинное время – 461 538 тенге или 51,7%, и фонд оплаты труда – 137 500 тенге или 15,4%.

Приложение является собственностью разработчика, может свободно меняться, дополняться и улучшаться при сравнительно небольших дополнительных вложениях.

5 Безопасность жизнедеятельности

5.1 Анализ условий труда

Рассмотрим помещения в котором ведется разработка данной программы. Помещение имеет размеры: длина (L) = 6.5 метров, ширина (B) = 4.5 метра, высота (H) = 4 метра. Помещение находится в здании на 4–м этаже. Согласно санитарным требованиям от 01.12.2011 года площадь на одно рабочее место пользователей ПК и ВТ на базе электронно–лучевой трубки (далее – ЭЛТ) составляет не менее 6 квадратных метров. Общая площадь рабочего помещения удовлетворяет санитарным требованиям 29,25 кв.м, что на 5.25 кв.м больше.

В помещении, при работе персонального компьютера, соблюдаются следующие климатические условия:

- а) Холодный период года:
 - 1) температура 23 – 22°С;
 - 2) относительная влажность 50 – 60 %;
 - 3) скорость движение воздуха относительная 0,1 м/с.
- б) Тёплый период года:
 - 1) температура 23 – 22°С;
 - 2) относительная влажность 40 – 60;
 - 3) скорость движение воздуха относительная 0,15 м/с.

Согласно санитарно–эпидемиологическим требованиям к условиям работы с источниками физических факторов (компьютеры и видеотерминалы), оказывающих воздействие на человека от 1 декабря 2011 года, оптимальными микроклиматическими условиями при легкой 1б категории работ являются следующие показатели:

- температура воздуха 23 – 21°С – в холодное время года, 22 – 24°С – в теплое;
- относительная влажность воздуха 40 – 60% как в холодное, так и в теплое время года;
- скорость движения воздуха 0,2 м/с – в теплое время года, 0,1 м/с – в холодное.

Вышеуказанные климатические условия соответствует установленным нормам производственного микроклимата.

В соответствии с санитарно–эпидемиологическими требованиями к условиям работы с источниками физических факторов (компьютеры и видеотерминалы), оказывающих воздействие на человека от 1 декабря 2011 года уровень звукового давления в помещениях работы операторов не должен превышать 75 дБА. Согласно паспортным данным оборудования уровень звукового давления составляет 50 дБА, что не превышает нормы.

Зрительную работу оператора следует отнести к 4–ому разряду, следовательно, освещенность должна быть не менее 300 лк, предусмотренная

санитарными правилами «Санитарно–эпидемиологические требования к условиям работы с источниками физических факторов (компьютеры и видеотерминалы), оказывающих воздействие на человека» от 1 декабря 2011 года. Искусственное освещение: 4 светильников, в каждом по 2 люминесцентные лампы – ЛДЦ40. Естественное освещение: 2 окна, размером 2х1,5 м, с двойным остеклением. Помещение оборудовано общей системой кондиционирования.

Так как разработка программного продукта ведется и в темное время суток необходимо рассчитать искусственное освещение в помещении. При длительной зрительной работе у компьютера с недостаточным освещением могут развиваться различные глазные болезни и нарушение нервной системы.

Программисты сталкиваются с воздействием таких физически опасных и вредных производственных факторов, как повышенный уровень шума, повышенная температура внешней среды, отсутствие или недостаток естественного света, недостаточная освещённость рабочей зоны, электрический ток, статическое электричество, электромагнитное излучение и др. Многие разработчики связаны с воздействием таких психофизиологических факторов, как умственное перенапряжение, перенапряжение зрительных и слуховых анализаторов, монотонность труда, эмоциональные перегрузки.

Воздействие указанных неблагоприятных факторов приводит к снижению работоспособности, вызываемое развивающимся утомлением. Появление и развитие утомления связано с изменениями, возникающими в процессе работы в центральной нервной системе, с тормозными процессами в коре головного мозга.

Повышенная температура внешней среды не создаёт нарушения состояния здоровья работающего, но вызывает дискомфортные ощущения, ухудшает самочувствие и понижает работоспособность.

Правильно спроектированное и выполненное освещение обеспечивает высокий уровень работоспособности, оказывает положительное влияние на работающих.

Проявление вредного воздействия шума на организм человека разнообразно: шум затрудняет разборчивость речи, вызывает снижение работоспособности и мешает нормальному отдыху. Действие шума не ограничивается только на органы слуха. Через нервные волокна шум передаётся в центральную и вегетативную нервную системы, а через них воздействует на внутренние органы, приводя к значительным изменениям в функциональном состоянии организма. Люди, работающие в условиях повышенного шума, жалуются на быструю утомляемость, головную боль, бессонницу. У человека ослабляется внимание, страдает память.

План помещения выбранного для размещения оборудования и технического персонала изображен на рисунке 5.1.

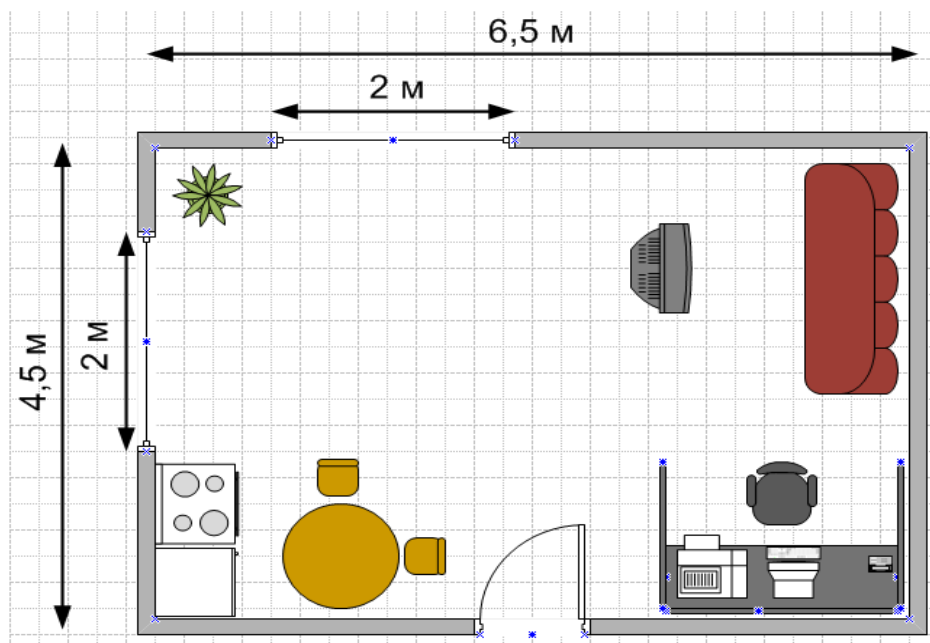


Рисунок 5.1 – План рабочего помещения

5.2 Анализ оборудования

Технические характеристики устройства:

- Intel(R)Core(TM) i7–3612QM(CPU 2.10 GHz);
- HD Graphics 4000 2 ГБ;
- AMD RADEON 8750M1 ГБ;
- ОЗУ 8ГБ;
- HDD 1ТБ;
- габариты: 1366x768(персональный компьютер + стол);
- электропитание: переменное напряжение 220 – 250 В, частотой 50 Гц, мощность 400Вт;
- стул: высота 0,5;
- стол: высота – 0,8 м; длина – 2м; ширина 3,5 м.

5.3 Расчет естественного освещения

При проектировании естественного освещения помещения необходимо определить площадь световых проемов, обеспечивающих нормированное значение КЕО (коэффициент естественной освещенности) в соответствии с требованиями СНиП РК 2.04–05–2002 «Естественное и искусственное освещение. Нормы проектирования». Расчет производится по следующим параметрам: Помещение 6,5x4,5x3, высота окна 1,5м., высота подоконника – 0,8м. Количество светильников 4. Разряд зрительных работ IV, в. $N_{зд}=10$, коэффициенты отражения 50x30x10.

Площадь световых проемов при боковом освещении рассчитывается по формуле

$$100 \frac{S_o}{S_n} = \frac{e_n K_3 \eta_o}{\tau_o r_1} K_{зд} \quad (5.1)$$

где S_o – площадь световых проемов при боковом освещении, m^2 ;

S_n – площадь пола помещения, m^2 ;

e_n – нормируемое значение КЕО;

K_3 – коэффициент запаса;

η_o – световая характеристика окон;

τ_o – общий коэффициент светопропускания;

r_1 – коэффициент, учитывающий повышение КЕО при боковом освещении, благодаря свету, отраженному от поверхности помещения и подстилающего слоя, примыкающего к зданию;

$K_{зд}$ – коэффициент, учитывающий затемнение окон противостоящими зданиями.

Выводим из формулы 1 площадь световых проемов S_o

$$S_o = \frac{S_n e_n K_3 \eta_o}{100 \cdot \tau_o r_1} K_{зд} \quad (5.2)$$

где S_n – площадь пола учебного помещения равна

$$S_n = L \cdot B \quad (5.3)$$

где e_n – нормируемое значение КЕО для зданий располагаемых в различных районах определяется по формуле

$$e_n = e_{кео} m_n \quad (5.4)$$

где $e_{кео}$ – значение КЕО, с учетом разряда зрительных работ (IV, в) при боковом совместном освещении, $e_{кео} = 0,9\%$;

m_n – коэффициент светового климата. Для световых проемов в наружных стенах зданий с ориентацией на юг для светового пояса г. Алматы, $m_n = 0,8$;

K_3 – коэффициент запаса. $K_3 = 1,2$;

η_o – световая характеристика окон.

Рассчитываем необходимые значения

$$S_n = 4,5 \cdot 6,5 = 29,45 m^2$$

$$e_n = 0,9 \cdot 0,8 = 0,72$$

При двустороннем боковом освещении нормируется минимальное значение КЕО в точке посередине помещения

$$l = \frac{B}{2} \quad (5.5)$$

При боковом одностороннем освещении нормируется минимальное значение КЕО в точке, расположенной на расстоянии 1 метра от стены, наиболее удаленной от световых проемов

$$l = B - 1 \quad (5.6)$$

В нашем помещении освещение двустороннее боковое, поэтому глубина помещения принимается как

$$l = \frac{B}{2} = \frac{4,5}{2} = 2,25 \text{ м}$$

Отношение длины помещения к его глубине

$$\frac{L}{l} = \frac{6,5}{2,25} = 2,9$$

Высота от уровня условной рабочей поверхности до верха окна вычисляется по формуле

$$h_1 = h_{\text{ок}} + h_{\text{пок}} - h_{\text{рпов}} \quad (5.7)$$

где $h_{\text{ок}}$ – высота окна;

$h_{\text{пок}}$ – высота подоконника;

$h_{\text{рпов}}$ – уровень рабочей поверхности, который изменяется в пределах $0,8 \div 1$ м. Условно примем, что $h_{\text{рпов}} = 0,9$ м

$$h_1 = 1,5 + 0,8 - 0,9 = 1,4 \text{ м}$$

Тогда отношение глубины помещения к его высоте от уровня условной рабочей поверхности до верха окна будет равен

$$\frac{l}{h_1} = \frac{2,25}{1,4} = 1,6 \quad (5.8)$$

По справочной таблице определяем значение $\eta_0 = 9$.

τ_0 – общий коэффициент светопропускания определяется по формуле

$$\tau_0 = \tau_1 \tau_2 \tau_3 \tau_4 \quad (5.9)$$

где τ_1 – коэффициент светопропускания материала. Для двойного листового оконного стекла $\tau_1=0,8$;

τ_2 – коэффициент, учитывающий потери света в переплетах светопроема. Для деревянных двойных отдельных переплетов $\tau_2=0,6$;

τ_3 – коэффициент, учитывающий потери света в несущих конструкциях, при боковом освещении равен 1;

τ_4 – коэффициент, учитывающий потери света в солнцезащитных устройствах. $\tau_4=1$.

Рассчитаем коэффициент светопропускания

$$\tau_0 = 0,8 \cdot 0,6 \cdot 1 \cdot 1 = 0,48$$

r_1 – коэффициент, учитывающий повышение КЕО при боковом освещении, благодаря свету, отраженному от поверхности помещения и подстилающего слоя, примыкающего к зданию. Для определения значения r_1 рассчитаем отношение расстояния расчетной точки от наружной стены к глубине помещения

$$\frac{l}{B} = \frac{2,25}{4,5} = 0,5 \quad (5.10)$$

Средневзвешенный коэффициент отражения потолка, стен и пола

$$\rho = \frac{\rho_{\text{пот}} + \rho_{\text{пол}} + \rho_{\text{ст}}}{3} \quad (5.11)$$

$$\rho = \frac{50 + 30 + 10}{3} = 30$$

По справочной таблице $r_1=(1,4+1,15)/2=1,275$.

$K_{\text{зд}}$ – коэффициент, учитывающий затемнение окон противостоящими зданиями, где отношение расстояния до рядом стоящего здания к высоте здания равно

$$\frac{P}{H_{\text{зд}}} = \frac{5}{10} = 0,5 \quad (5.12)$$

Тогда $K_{\text{зд}} = 1,7$

Подставим все найденные значения в формулу 2 и найдем общую площадь световых проемов

$$S_0 = \frac{29,25 \cdot 0,72 \cdot 1,2 \cdot 9}{100 \cdot 0,48 \cdot 1,275} \cdot 1,7 = 6,13 \text{ м}^2$$

Учитывая размеры помещения: 6,5x4,5x4, поделим общую площадь световых проемов на 2 окна, по одному на северную и западную сторону

$$S_{\text{окн}} = \frac{S_0}{2} = \frac{6,13}{2} = 3 \text{ м}^2$$

Найдем ширину каждого окна

$$l_{\text{ок}} = \frac{S_{\text{окн}}}{h_{\text{ок}}} = \frac{3}{1,5} = 2 \text{ м}$$

Таким образом размеры окна: высота $h_{\text{ок}} = 1,5 \text{ м}$, а ширина $l_{\text{ок}} = 2 \text{ м}$.

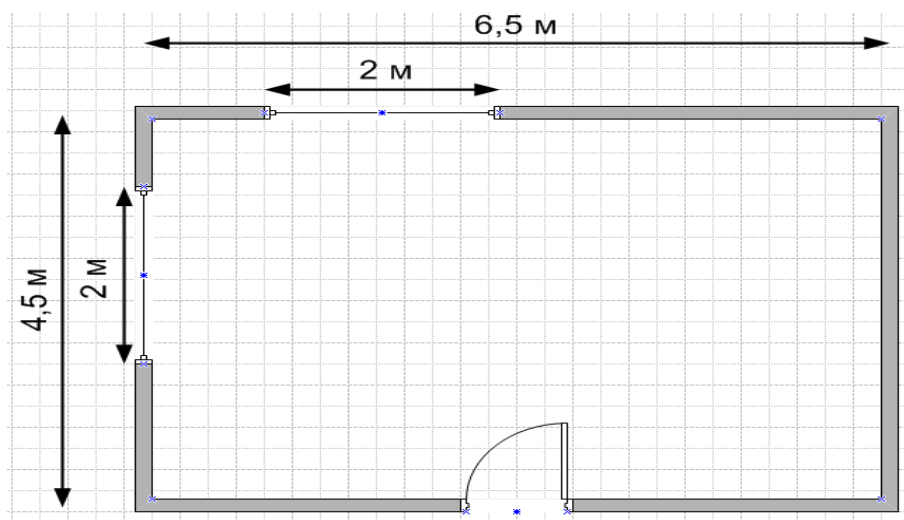


Рисунок 5.2 – Расположение окон в рабочем помещении

5.4 Расчет системы кондиционирования офиса

Приведем расчет системы кондиционирования в помещении. Кондиционирование обеспечивает наилучшее микроклимата в помещении и условия работы точной и чувствительной аппаратуры, и должно выполняться в соответствии с главой СНиП 11–33–75 "Отопление, вентиляция и кондиционирование воздуха".

Определим количество воздуха L м³/ч, которое необходимо вывести за один час из помещения, чтобы вместе с ним удалить избыток тепла $Q_{изб}$, определяется по формуле

$$L = \frac{Q_{изб}}{C_B \times t \times y_B} = \text{м}^3/\text{ч} \quad (5.13)$$

где C_B – теплоемкость сухого воздуха, ккал/кг ($C_B=0,24$ ккал/кг град);

$t=t_{вх} - t_{вых}$ при расчетах возьмем $t=8^\circ\text{C}$;

y_B – плотность уходящего воздуха, определяемая в зависимости от температуры, кг/м³ (при расчетах принимается $y_B=1,20$ кг/м³).

Определим избыточное тепло $Q_{изб}$ ккал/ч

$$Q_{изб} = Q_n - Q_{от} \quad (5.14)$$

где Q_n – количества тепла поступающего в воздух помещения, ккал/ч;

$Q_{от}$ – теплоотдача в окружающую среду через наружные ограждения (в теплое время года, при расчетах можно принять нулю).

Количество тепловыделений Q_n зависит от мощности оборудования, числа работающих людей и тепла, которое вносится в помещение через оконные проемы

$$Q_n = Q_{об} + Q_l + Q_p + Q_{осв} \quad (5.15)$$

где $Q_{об}$ – тепло, выделяемое производственным оборудованием, ккал/ч;

Q_l – тепло выделяемое людьми, ккал/ч;

Q_p – тепло, вносимое солнечной радиацией, ккал/ч;

$Q_{осв}$ – тепло, выделяемое системой освещения.

Тепло, выделяемое производственным оборудованием в рабочем помещении, определяется из соотношения

$$Q_{об} = 860 \times P_{об} \times n \times N \quad (5.16)$$

где 860 тепловой эквивалент 1 кВт·ч, то есть тепло, эквивалентное 1 кВт·ч электрической энергии;

$P_{об}$ – мощность, потребляемая оборудованием $P_{об}=350$ Вт;

n – коэффициент перехода тепла в помещение, $n=0,75$;

N – количество производственного оборудования.

Тепло, выделяемое производственным оборудованием равно

$$Q_{ог} = 860 \times 0,35 \times 0,75 \times 5 = 1128,75 \text{ ккал/ч}$$

Тепло, вносимое солнечной, радиацией, определяется из соотношения

$$Q_p = m \times F \times g_{ог} \quad (5.17)$$

где m – количество окон в помещении;

F – площадь одного окна $F=3 \text{ м}^2$;

$g_{ог}$ – солнечная радиация через остекленную поверхность, то есть количество тепла, вносимое за 1ч через остекление площадью в 1 м^2

$$Q_p = 1 \times 3 \times 145 = 435 \text{ ккал}$$

Количество тепла, выделяемого находящимися в помещении людьми

$$Q_{л} = N \times q_{л} \quad (5.18)$$

где N – количество людей в помещении;

$q_{л}$, ккал/ч – тепловыделения одного человека $q_{л}=60$

$$Q_{л} = 5 \times 60 = 300 \text{ ккал/ч}$$

Количество тепла, выделяемого системой освещения

$$Q_{осв} = 860 \times P_{осв} \times \alpha \times \beta \times \cos(\varphi) \quad (5.19)$$

где α – коэф. перевода электрической энергии в тепловую, люминесцентные лампы $\alpha = 0,46 - 0,48$;

β – коэффициент одновременности работы (при работе всех светильников $\beta = 1$);

$\cos(\varphi)$ – коэффициент мощности = $0,7 - 0,8$;

$P_{осв}$ – мощность осветительной установки = $0,04 \text{ кВт}$

$$Q_{осв} = 860 \times 0,04 \times 0,48 \times 0,8 \times 0,8 = 10,57 \text{ ккал/ч}$$

Тогда тепловыделение составит

$$Q_{изб} = Q_{п} = 1128,75 + 435 + 300 + 10,57 = 1874,32 \text{ ккал/ч}$$

Необходимый воздухообмен рассчитаем

$$L=1874,32/(0,24 \times 8 \times 1,2)=813,51 \text{ м}^3/\text{ч}$$

Отношение количества воздуха, поступающего в помещение за один час, к объему помещения называется кратностью воздухообмена

$$K = \frac{L}{V_{\text{п}}} \quad (5.20)$$

где $V_{\text{п}}$ – объем помещения $V_{\text{п}}=117 \text{ м}^3$

$$K=813,51/117=6,9$$

Находим требуемую производительность кондиционера

$$W_{\text{к}} = k_3 \times L \quad (5.21)$$

где k_3 – коэффициент запаса, $k_3=1,3 \div 2,0$

$$W_{\text{к}} = 2 \times 813,51=1627,02 \text{ м}^3/\text{ч}$$

Исходя из расчетов в помещение с оборудованием, для соблюдения требуемых параметров микроклимата следует установить один кондиционер с производительностью не менее $1627,02 \text{ м}^3/\text{ч}$. Выбранный кондиционер Mitsubishi LMK48HRS1. Расход воздуха (внутр./внешн.), $\text{м}^3/\text{ч}$: 1600/1800/2000. Размещение кондиционера в помещении указан на рисунке 5.3.

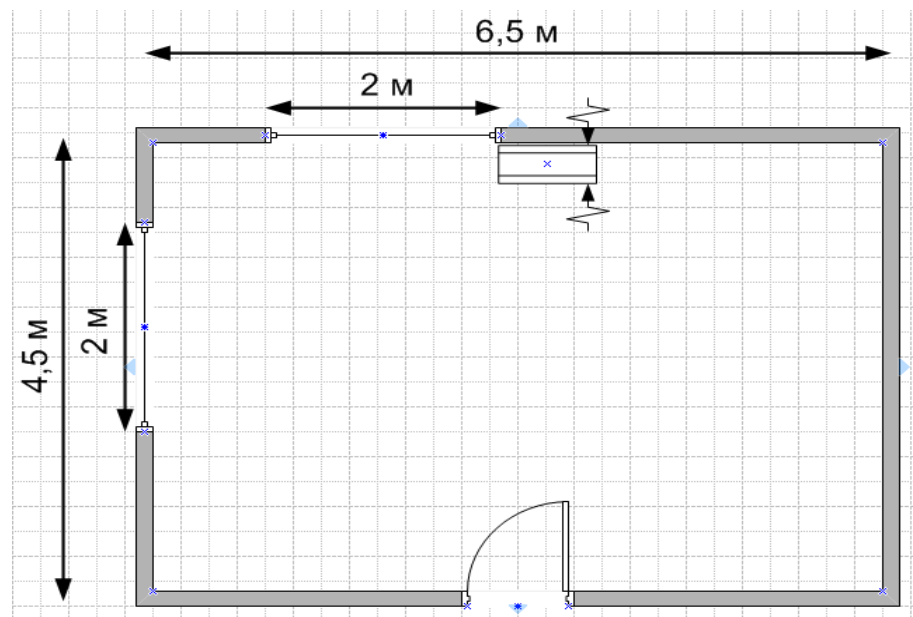


Рисунок 5.3 размещение кондиционера в помещении

Вывод

В данном разделе был произведён анализ условий труда в рабочем помещении администратора сайта. Уровень условий труда признан допустимым, и данные, полученные из расчетов, полностью удовлетворяют требованиям стандартов безопасности жизнедеятельности.

Исходя из расчетов в помещение с оборудованием, для соблюдения требуемых параметров естественного освещения общая площадь световых проемов должна составлять $6,13\text{ м}^2$, так как у нас два окна площадь каждого окна должна составлять приблизительно 3 м^2 . Размеры имеющихся окон $2 \times 1,5$ что и составляют площадь в 3 м^2 . Исходя из расчетов системы кондиционирования офиса следует установить один кондиционер с производительностью не менее $1627,02\text{ м}^3/\text{ч}$. Для этого мы выбрали кондиционер японской марки Mitsushito, что удовлетворяет требованию системы кондиционирования.

Заключение

Получения образования, т.е. быть образованным является закономерным, естественным и необходимым для современного человека. От самого рождения, на любой стадии развития человек получает информацию необходимую для его жизнедеятельности. Ведь не зря говорят "тот кто владеет информацией владеет миром". Все больше жителей мира и в том числе жители нашей страны стараются получить все больше и больше информации, данная информация находит свои источники в настоящее время в интернете. Скорость нахождения информации, и легко восприятия его в не зависимости от времени вот это и есть характеристики жизни современных людей.

Исходя из выше сказанного можно сделать вывод, что получить дистанционно образование возможно в не зависимости от времени, места, деятельности человека а также возрастных категорий. Современное структура дистанционного образования составлена таким образом, что она включает в себя элементы параллельно нескольких разных методов образования: очного, заочного, экстерната.

Использование в качестве связующего звена видеоконференции и других новых систем, максимально приблизит дистанционное образование к традиционным методам, а также будет прямой контакт и взаимоотношение между преподавателем и студентом. Столетиями проводимые групповые семинарские занятия, можно осуществить также посредством видеоконференций.

Конечно же, в настоящее время данный метод только развивается в нашей стране, но научно–исследовательские и практические работы над проблемами дистанционного вида образования активно ведутся.

Я с уверенностью утверждаю, что дистанционное обучение это образование XXI–го века, новая эра в сфере получения образования. С года в год этот метод будет модернизироваться, максимально приближаться к идеальному методу и внедрится во все ВУЗы нашей страны, а также во всем мире.

Список литературы

- 1 Линн Бейли, Майкл Моррисон. Изучаем PHP и MySQL. – М.: Эксмо, 2010. – 786 с.
- 2 Веллинг Л., Томсон Л., Разработка web-приложений с помощью PHP и MySQL – М.: Вильямс, 2008. – 880 с.
- 3 Суэринг С., Конверс С., Парк Д. PHP и MySQL. Библия программиста. – СПб: Диалектика, 2010. – 912 с.
- 4 Котеров Д., Костарев А. PHP 5. Серия в подлиннике. – СПб: BHV. – 1120 с.
- 5 Сайт <http://php.com>
- 6 Томсон Л. Разработка Web-приложений на PHP и MySQL. /Пер. с англ./Томсон Л., Веллинг Л. -2-е изд. – СПб: ООО ДиаСофтЮП, 2003.
- 7 Сайт <http://htmlbook.ru>
- 8 Сайт <http://javascript.com>
- 9 Харрис Э. PHP и MySQL для начинающих. – СПб: Кудиц-образ, 2005. – 384 с.
- 10 Суэринг С., Конверс С., Парк Д. PHP и MySQL. Библия программиста. – СПб: Диалектика, 2010. – 912 с.
- 11 Сайт <http://mysql.com>
- 12 Закон Республики Казахстан от 27 июля 2007 года «Об образовании»
- 13 СНиП РК 2.04 – 05 – 2002 «Естественное и искусственное освещение. Общие требования».
- 14 СНиП РК 3.02 – 04 – 2009 «Отопление, вентиляция и кондиционирование воздуха».
- 15 Безопасность жизнедеятельности. / Под ред. Арустамова Э.А. – Москва: Издательский дом «Дашков и К», 2002. – 676с.
- 16 З.Д. Еркешева, Г.Ш. Боканова. Методические указания к выполнению экономической части дипломных работ для студентов специальности 5В070400 – Вычислительная техника и программное обеспечение. – Алматы: АУЭС, 2013 – 40 с.
- 17 Экономика, организация и управление на предприятии Учебник/ Корсаков М.Н., Ребрин Ю.И., Федосова Т.В., Макареня Т.А., Шевченко И.К. и др.; Под ред. М.А.Боровской. – Таганрог: ТТИ ЮФУ, 2008. – 440с.
- 18 Аманбаев У.А. Экономика предприятия. –А.: «Бастау», 2012.
- 19 Буров В.П. Бизнес-план фирмы. –М.: «Инфра–М», 2011.
- 20 Куатова Д.Я. Экономика предприятия. –А.: «Экономика», 2011.

Приложение А

Листинг программы

```
<?php
/**
 * This page provides the Administration -> ... -> Theme selector
 UI.
 *
 * @package core
 * @license http://www.univer.org/copyleft/gpl.html
 * author Mukhtarov Sultan
 */

require_once(dirname(__FILE__) . '/../config.php');
require_once($CFG->libdir . '/adminlib.php');

$choose = optional_param('choose', '', PARAM_PLUGIN);
$reset = optional_param('reset', 0, PARAM_BOOL);
$device = optional_param('device', '', PARAM_TEXT);
$unsettheme = optional_param('unsettheme', 0, PARAM_BOOL);

admin_externalpage_setup('themeselector');

if (!empty($device)) {
    // Make sure the device requested is valid.
    $devices = core_useragent::get_device_type_list();
    if (!in_array($device, $devices)) {
        // The provided device isn't a valid device throw an
 error.
        print_error('invaliddevicetype');
    }
}
unset($SESSION->theme);
if ($reset and confirm_sesskey()) {
    theme_reset_all_caches();

} else if ($choose && $device && !$unsettheme &&
confirm_sesskey()) {
    // Load the theme to make sure it is valid.
    $theme = theme_config::load($choose);
    // Get the config argument for the chosen device.
    $themename =
core_useragent::get_device_type_cfg_var_name($device);
    set_config($themename, $theme->name);

    // Create a new page for the display of the themes readme.
    // This ensures that the readme page is shown using the new
 theme.
    $confirmpage = new univer_page();
    $confirmpage->set_context($PAGE->context);
    $confirmpage->set_url($PAGE->url);
}
```


Продолжение приложения А

```
$confirmpage->set_pagetype($PAGE->pagetype);
$confirmpage->set_title($PAGE->title);
$confirmpage->set_heading($PAGE->heading);

// Get the core renderer for the new theme.
$output = $confirmpage->get_renderer('core');

echo $output->header();
echo $output->heading(get_string('themesaved'));
echo $output->box_start();
echo format_text(get_string('choosereadme', 'theme_'. $theme-
>name), FORMAT_SKT);
echo $output->box_end();
echo $output->continue_button($CFG->wwwroot .
'/theme/index.php');
echo $output->footer();
exit;
} else if ($device && $unsettheme && confirm_sesskey() && ($device
!= 'default')) {
    // Unset the theme and continue.

unset_config(core_useragent::get_device_type_cfg_var_name($device)
);
    $device = '';
}

// Otherwise, show either a list of devices, or is
enableddevicedetection set to no or a
// device is specified show a list of themes.

$table = new html_table();
$table->data = array();
$heading = '';
if (!empty($CFG->enableddevicedetection) && empty($device)) {
    $heading = get_string('selectdevice', 'admin');
    // Display a list of devices that a user can select a theme
for.

    $strthemenosselected = get_string('themenosselected', 'admin');
    $strthemeselect = get_string('themeselect', 'admin');

    // Display the device selection screen.
    $table->id = 'admindeviceselector';
    $table->head = array(get_string('devicetype', 'admin'),
get_string('currenttheme', 'admin'), get_string('info'));

    $devices = core_useragent::get_device_type_list();
    foreach ($devices as $thedevice) {
```

Продолжение приложения А

```
$headingthemename = ''; // To output the picked theme name
when needed.
    $themename =
core_useragent::get_device_type_theme($thedevice);
    if (!$themename && $thedevice == 'default') {
        $themename = theme_config::DEFAULT_THEME;
    }

    $screenshotcell = $strthemename;
    $unsetthemebutton = '';
    if ($themename) {
        // Check the theme exists.
        $themename = clean_param($themename, PARAM_THEME);
        if (empty($themename)) {
            // Likely the theme has been deleted.

unset_config(core_useragent::get_device_type_cfg_var_name($thedevice));
        } else {
            $strthemename = get_string('pluginname',
'theme_'.$themename);
            // Link to the screenshot, now mandatory - the
image path is hardcoded because we need image from other themes,
            // not the current one.
            $screenshoturl = new
univer_url('/theme/image.php',
            array('theme' => $themename, 'image' =>
'screenshot', 'component' => 'theme'));
            // Contents of the screenshot/preview cell.
            $screenshotcell = html_writer::empty_tag('img',
array('src' => $screenshoturl, 'alt' => $strthemename));
            // Show the name of the picked theme.
            $headingthemename = $OUTPUT-
>heading($strthemename, 3);
        }
        // If not default device then show option to unset
theme.
        if ($thedevice != 'default') {
            $unsetthemestr = get_string('unsettheme',
'admin');
            $unsetthemeurl = new
univer_url('/theme/index.php',
            array('device' => $thedevice, 'sesskey' =>
sesskey(), 'unsettheme' => true));
            $unsetthemebutton = new
single_button($unsetthemeurl, $unsetthemestr, 'get');
            $unsetthemebutton = $OUTPUT-
>render($unsetthemebutton);
        }
    }
}
```

Продолжение приложения A

```
$deviceurl = new skt_url('/theme/index.php',
array('device' => $thedevice, 'sesskey' => sesskey()));
$select = new single_button($deviceurl, $strthemeselect,
'get');

$table->data[] = array(
    $OUTPUT->heading(ucfirst($thedevice), 3),
    $screenshotcell,
    $headingthemename . $OUTPUT->render($select) .
$unsetthemebutton
);
}
} else {
    // Either a device has been selected or $CFG->
    >enabledevicedetection is off so display a list
    // of themes to select.
    $heading = get_string('selecttheme', 'admin', $device);
    if (empty($device)) {
        // If $CFG->enabledevicedetection is off this will return
        'default'.
        $device = core_useragent::get_device_type();
    }

    $table->id = 'adminthemeselector';
    $table->head = array(get_string('theme'), get_string('info'));

    $themes = core_component::get_plugin_list('theme');

    foreach ($themes as $themenamename => $themedir) {

        // Load the theme config.

        try {
            $theme = theme_config::load($themenamename);
        } catch (Exception $e) {
            // Bad theme, just skip it for now.
            continue;
        }
        if ($themenamename !== $theme->name) {
            // Obsoleted or broken theme, just skip for now.
            continue;
        }
        if (empty($CFG->themedesignermode) && $theme->
        >hidefromselector) {
            // The theme doesn't want to be shown in the theme
            selector and as theme
            // designer mode is switched off we will respect that
            decision.
            continue;
        }
    }
}
```

Продолжение приложения A

```
$strthemename = get_string('pluginname',
'theme_'.$themename);

// Build the table row, and also a list of items to go in
the second cell.
$row = array();
$infoitems = array();
$rowclasses = array();

// Set up bools whether this theme is chosen either main
or legacy.
$ischosentheme = ($themename ==
core_useragent::get_device_type_theme($device));

if ($ischosentheme) {
    // Is the chosen main theme.
    $rowclasses[] = 'selectedtheme';
}

// Link to the screenshot, now mandatory - the image path
is hardcoded because we need image from other themes,
// not the current one.
$screenshotpath = new univer_url('/theme/image.php',
    array('theme'=>$themename, 'image'=>'screenshot',
'component'=>'theme'));
// Contents of the first screenshot/preview cell.
$row[] = html_writer::empty_tag('img',
array('src'=>$screenshotpath, 'alt'=>$strthemename));
// Contents of the second cell.
$infocell = $OUTPUT->heading($strthemename, 3);
// Button to choose this as the main theme or unset this
theme for devices other than default.
if (($ischosentheme) && ($device != 'default')) {
    $unsetthemestr = get_string('unsettheme', 'admin');
    $unsetthemeurl = new skt_url('/theme/index.php',
        array('device' => $device, 'unsettheme' => true,
'sesskey' => sesskey()));
    $unsetbutton = new single_button($unsetthemeurl,
$unsetthemestr, 'get');
    $infocell .= $OUTPUT->render($unsetbutton);
} else if (!$ischosentheme) {
    $setthemestr = get_string('usethe theme');
    $setthemeurl = new skt_url('/theme/index.php',
        array('device' => $device, 'choose' => $themename,
'sesskey' => sesskey()));
    $setthemebutton = new single_button($setthemeurl,
$setthemestr, 'get');
    $infocell .= $OUTPUT->render($setthemebutton);
}
}
```

Продолжение приложения A

```
$row[] = $infocell;

$table->data[$themenam] = $row;
$table->rowclasses[$themenam] = join(' ', $rowclasses);
}
}
echo $OUTPUT->header('themeselector');
echo $OUTPUT->heading($heading);

$params = array('sesskey' => sesskey(), 'reset' => 1);
if (!empty($device)) {
    $params['device'] = $device;
}
echo $OUTPUT->single_button(new univer_url('index.php', $params),
get_string('themeresetcaches', 'admin'));

echo html_writer::table($table);

echo $OUTPUT->footer();
/* Custom CSS
-----*/
#page-header h1 {
    color:#000000;
    font-size:24px;
    font-family:Georgia;
    text-transform:uppercase;
    float:left;
    margin:-10px 0 5px;
}
a.logo {
    background: url([[setting:logo]]) no-repeat 0 0;
    display: block;
    float: left;
    height: 75px;
    margin: 0;
    padding: 0;
    width: 100%;
}

.dir-rtl a.logo {
    background: url([[setting:logo]]) no-repeat 100% 0;
    display: block;
    float: right;
}

/* Custom CSS Settings
-----*/
[[setting:customcss]]
<?php
// This file is part of SKT - http://
```

Продолжение приложения A

```
// comment out when debugging or better look into error log!
define('NO_DEBUG_DISPLAY', true);

// we need just the values from config.php and minlib.php
define('ABORT_AFTER_CONFIG', true);
require('../config.php'); // this stops immediately at the
beginning of lib/setup.php
require_once("$CFG->dirroot/lib/jslib.php");

if ($slashargument = min_get_slash_argument()) {
    $slashargument = ltrim($slashargument, '/');
    if (substr_count($slashargument, '/') < 2) {
        image_not_found();
    }
    // image must be last because it may contain "/"
    list($themename, $rev, $type) = explode('/', $slashargument,
3);
    $themename = min_clean_param($themename, 'SAFEDIR');
    $rev        = min_clean_param($rev, 'INT');
    $type       = min_clean_param($type, 'SAFEDIR');

} else {
    $themename = min_optional_param('theme', 'standard',
'SAFEDIR');
    $rev        = min_optional_param('rev', -1, 'INT');
    $type       = min_optional_param('type', 'head', 'RAW');
}

if ($type !== 'head' and $type !== 'footer') {
    header('HTTP/1.0 404 not found');
    die('Theme was not found, sorry.');
```

```
}

if (file_exists("$CFG->dirroot/theme/$themename/config.php")) {
    // exists
} else if (!empty($CFG->themedir) and file_exists("$CFG->
>themedir/$themename/config.php")) {
    // exists
} else {
    header('HTTP/1.0 404 not found');
    die('Theme was not found, sorry.');
```

```
}

$candidate = "$CFG->
>localcachedir/theme/$rev/$themename/javascript_$type.js";
$etag = sha1("$rev/$themename/$type");
```

```
if ($rev > 0 and file_exists($candidate)) {
    if (!empty($_SERVER['HTTP_IF_NONE_MATCH']) ||
!empty($_SERVER['HTTP_IF_MODIFIED_SINCE'])) {
```

Продолжение приложения А

```
// we do not actually need to verify the etag value because our
files
// never change in cache because we increment the rev parameter
    js_send_unmodified(filemtime($candidate), $etag);
    }
    js_send_cached($candidate, $etag);
}

// ok, now we need to start normal script, we need to load all
libs and $DB
define('ABORT_AFTER_CONFIG_CANCEL', true);
define('NO_UPGRADE_CHECK', true); // Ignore upgrade check
require("$CFG->dirroot/lib/setup.php");

$theme = theme_config::load($themename);
$themerev = theme_get_revision();

if ($themerev <= 0 or $rev != $themerev) {
    // Do not send caching headers if they do not request current
revision,
    // we do not want to pollute browser caches with outdated JS.
    js_send_uncached($theme->javascript_content($type));
}

make_localcache_directory('theme', false);

js_write_cache_file_content($candidate,
core_minify::js_files($theme->javascript_files($type)));
// Verify nothing failed in cache file creation.
clearstatcache();
if (file_exists($candidate)) {
    js_send_cached($candidate, $etag);
}

js_send_uncached($theme->javascript_content($type));
<?php
// it under the terms of the GNU General Public License as
published by
// the Free Software Foundation, either version 3 of the License,
or
// (at your option) any later version.
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
// GNU General Public License for more details.
//
// You should have received a copy of the GNU General Public
License
```

Продолжение приложения А

```
/**
 * jQuery serving script.
 *
 * Do not include jQuery scripts or CSS directly, always use
 * $PAGE->requires->jquery() or $PAGE->requires-
 >jquery_plugin('xx', 'yy').
 *
 * @package    core
 * @copyright  2014 Mukhtarov Sultan {@link http://univer.org}
 * @license    http://www.gnu.org/copyleft/gpl.html GNU GPL v3 or
 later
 */
// comment out when debugging or better look into error log!
define('NO_DEBUG_DISPLAY', true);
// We need just the values from config.php and minlib.php.
define('ABORT_AFTER_CONFIG', true);
require('../config.php'); // This stops immediately at the
beginning of lib/setup.php.

if ($slashargument = min_get_slash_argument()) {
    $path = ltrim($slashargument, '/');
} else {
    $path = min_optional_param('file', '', 'SAFEPATH');
    $path = ltrim($path, '/');
}

if (strpos($path, '/') === false) {
    jquery_file_not_found();
}

list($component, $path) = explode('/', $path, 2);

if (empty($path) or empty($component)) {
    jquery_file_not_found();
}

// Find the jQuery dir for this component.
if ($component === 'core') {
    $componentdir = "$CFG->dirroot/lib";
} else if (strpos($component, 'theme_')) {
    if (!empty($CFG->themedir)) {
        $componentdir = "$CFG->themedir/$component";
    } else {
        $componentdir = "$CFG->dirroot/theme/$component";
    }
} else {
    $componentdir =
core_component::get_component_directory($component);
}
```


Продолжение приложения А

```
if (!file_exists($componentdir) or
!file_exists("$componentdir/jquery/plugins.php")) {
    jquery_file_not_found();
}

$file = realpath("$componentdir/jquery/$path");

if (!$file or is_dir($file)) {
    jquery_file_not_found();
}

$etag = sha1("$component/$path");
$lifetime = 60*60*24*120; // 120 days should be enough.
$pathinfo = pathinfo($path);

if (empty($pathinfo['extension'])) {
    jquery_file_not_found();
}

$filename = $pathinfo['filename'].'.'.$pathinfo['extension'];

switch($pathinfo['extension']) {
    case 'gif' : $mimetype = 'image/gif';
                break;
    case 'png' : $mimetype = 'image/png';
                break;
    case 'jpg' : $mimetype = 'image/jpeg';
                break;
    case 'jpeg' : $mimetype = 'image/jpeg';
                break;
    case 'ico' : $mimetype = 'image/vnd.microsoft.icon';
                break;
    case 'svg' : $mimetype = 'image/svg+xml';
                break;
    case 'js' : $mimetype = 'application/javascript';
                break;
    case 'css' : $mimetype = 'text/css';
                break;
    case 'php' : jquery_file_not_found();
                break;
    default : $mimetype = 'document/unknown';
}

if (!empty($_SERVER['HTTP_IF_NONE_MATCH']) ||
!empty($_SERVER['HTTP_IF_MODIFIED_SINCE'])) {
    // We do not actually need to verify the etag value because
these files
    // never change, devs need to change file names on update!
    header('HTTP/1.1 304 Not Modified');
```

Продолжение приложения А

```
    header('Expires: '. gmdate('D, d M Y H:i:s', time() +
    $lifetime) .' GMT');
    header('Cache-Control: public, max-age= '.$lifetime);
    header('Content-Type: '.$mimetype);
    header('Etag: "'.$etag.'"');
    die;
}

require_once("$CFG->dirroot/lib/xsendfilelib.php");

header('Etag: "'.$etag.'"');
header('Content-Disposition: inline; filename="'.$filename.'"');
header('Last-Modified: '. gmdate('D, d M Y H:i:s',
filemtime($file)) .' GMT');
header('Expires: '. gmdate('D, d M Y H:i:s', time() + $lifetime)
.' GMT');
header('Pragma: ');
header('Cache-Control: public, max-age= '.$lifetime);
header('Accept-Ranges: none');
header('Content-Type: '.$mimetype);
if (xsendfile($file)) {
    die;
}
if ($mimetype === 'text/css' or $mimetype ===
'application/javascript') {
    if (!min_enable_zlib_compression()) {
        header('Content-Length: '.filesize($file));
    }
} else {
    // No need to compress images.
    header('Content-Length: '.filesize($file));
}
readfile($file);
die;
function jquery_file_not_found() {
    // Note: we can not disclose the exact file path here, sorry.
    header('HTTP/1.0 404 not found');
    die('File was not found, sorry.');
```

Продолжение приложения А

```
if (strpos($slashargument, '_s/') === 0) {
    // Can't use SVG.
    $slashargument = substr($slashargument, 3);
    $usesvg = false;
} else {
    $usesvg = true;
}

$chunk = null;
if (preg_match('#/(chunk(\d+)(/|$))#', $slashargument,
$matches)) {
    $chunk = (int)$matches[2];
    $slashargument = str_replace($matches[1], '',
$slashargument);
}

list($themename, $rev, $type) = explode('/', $slashargument,
3);
$themename = min_clean_param($themename, 'SAFEDIR');
$rev       = min_clean_param($rev, 'INT');
$type     = min_clean_param($type, 'SAFEDIR');

} else {
    $themename = min_optional_param('theme', 'standard',
'SAFEDIR');
    $rev       = min_optional_param('rev', 0, 'INT');
    $type     = min_optional_param('type', 'all', 'SAFEDIR');
    $chunk    = min_optional_param('chunk', null, 'INT');
    $usesvg   = (bool)min_optional_param('svg', '1', 'INT');
}

if ($type === 'editor') {
    // The editor CSS is never chunked.
    $chunk = null;
} else if ($type === 'all') {
    // We're fine.
} else {
    css_send_css_not_found();
}

if (file_exists("$CFG->dirroot/theme/$themename/config.php")) {
    // The theme exists in standard location - ok.
} else if (!empty($CFG->themedir) and file_exists("$CFG->
themedir/$themename/config.php")) {
    // Alternative theme location contains this theme - ok.
} else {
    header('HTTP/1.0 404 not found');
    die('Theme was not found, sorry.');
```

```
$candidatedir = "$CFG->localcachedir/theme/$rev/$themename/css";
```

Продолжение приложения A

```
$etag = "$rev/$themenamе/$type";
$candidatename = $type;
if (!$usesvg) {
    // Add to the sheet name, one day we'll be able to just drop
    this.
    $candidatedir .= '/nosvg';
    $etag .= '/nosvg';
}

if ($chunk != null) {
    $etag .= '/chunk'.$chunk;
    $candidatename .= '.'.$chunk;
}
$candidatesheet = "$candidatedir/$candidatename.css";
$etag = sha1($etag);

if (file_exists($candidatesheet)) {
    if (!empty($_SERVER['HTTP_IF_NONE_MATCH']) ||
!empty($_SERVER['HTTP_IF_MODIFIED_SINCE'])) {
        // We do not actually need to verify the etag value
        because our files
        // never change in cache because we increment the rev
        counter.
        css_send_unmodified(filemtime($candidatesheet), $etag);
    }
    css_send_cached_css($candidatesheet, $etag);
}

// Ok, now we need to start normal univеr script, we need to load
all libs and $DB.
define('ABORT_AFTER_CONFIG_CANCEL', true);
define('NO_UPGRADE_CHECK', true); // Ignore upgrade check.

require("$CFG->dirroot/lib/setup.php");

$theme = theme_config::load($themenamе);
$theme->force_svg_use($usesvg);

$themerev = theme_get_revision();

$cache = true;
if ($themerev <= 0 or $themerev != $rev) {
    $rev = $themerev;
    $cache = false;

    $candidatedir = "$CFG-
>localcachedir/theme/$rev/$themenamе/css";
    $etag = "$rev/$themenamе/$type";
    $candidatename = $type;
    if (!$usesvg) {
```

Продолжение приложения А

```
// Add to the sheet name, one day we'll be able to just
drop this.
$candidatedir .= '/nosvg';
$setag .= '/nosvg';
}

if ($chunk !== null) {
    $setag .= '/chunk' . $chunk;
    $candidatename .= '.' . $chunk;
}
$candidatesheet = "$candidatedir/$candidatename.css";
$setag = sha1($setag);
}

make_localcache_directory('theme', false);

if ($type === 'editor') {
    $csscontent = $theme->get_css_content_editor();
    css_store_css($theme, "$candidatedir/editor.css", $csscontent,
false);
} else {

    $lock = null;

    // Lock system to prevent concurrent requests to compile LESS,
    which is really slow and CPU intensive.
    // Each client should wait for one to finish the compilation
    before starting a new compiling process.
    // We only do this when the file will be cached...
    if ($type === 'less' && $cache) {
        $lockfactory =
\core\lock\lock_config::get_lock_factory('core_theme_get_css_conte
nt');
        // We wait for the lock to be acquired, the timeout does
        not need to be strict here.
        $lock = $lockfactory->get_lock($themename, rand(15, 30));
        if (file_exists($candidatesheet)) {
            // The file was built while we waited for the lock, we
            release the lock and serve the file.
            if ($lock) {
                $lock->release();
            }
            css_send_cached_css($candidatesheet, $setag);
        }
    }

    // Older IEs require smaller chunks.
    $csscontent = $theme->get_css_content();
}
```

Продолжение приложения A

```
$relroot = preg_replace('|^http.?://[^\/>|', '', $CFG->wwwroot);
if (!empty($slashargument)) {
    if ($usesvg) {
        $chunkurl =
"{ $relroot }/theme/styles.php/{ $themename }/{ $rev }/all";
    } else {
        $chunkurl =
"{ $relroot }/theme/styles.php/_s/{ $themename }/{ $rev }/all";
    }
} else {
    if ($usesvg) {
        $chunkurl =
"{ $relroot }/theme/styles.php?theme={ $themename }&rev={ $rev }&type=all";
    } else {
        $chunkurl =
"{ $relroot }/theme/styles.php?theme={ $themename }&rev={ $rev }&type=all&svg=0";
    }
}

css_store_css($theme, "$candidatedir/all.css", $csscontent,
true, $chunkurl);

// Release the lock.
if ($lock) {
    $lock->release();
}

if (!$cache) {
    // Do not pollute browser caches if invalid revision
    requested,
    // let's ignore legacy IE breakage here too.
    css_send_uncached_css($csscontent);
} else if ($chunk !== null and file_exists($candidatesheet)) {
    // Greetings stupid legacy IEs!
    css_send_cached_css($candidatesheet, $etag);
} else {
    // Real browsers - this is the expected result!
    css_send_cached_css_content($csscontent, $etag);
}
define('NO_DEBUG_DISPLAY', true);
define('NO_UPGRADE_CHECK', true);
define('NO_UNIVER_COOKIES', true);

require('../config.php');
```

Продолжение приложения А

```
require_once($CFG->dirroot.'/lib/csslib.php');

$themename = optional_param('theme', 'standard', PARAM_SAFEDIR);
$type       = optional_param('type', '', PARAM_SAFEDIR);
$subtype    = optional_param('subtype', '', PARAM_SAFEDIR);
$sheet      = optional_param('sheet', '', PARAM_SAFEDIR);
$usesvg     = optional_param('svg', 1, PARAM_BOOL);
$chunk      = optional_param('chunk', null, PARAM_INT);

if (file_exists("$CFG->dirroot/theme/$themename/config.php")) {
    // The theme exists in standard location - ok.
} else if (!empty($CFG->themedir) and file_exists("$CFG->
themedir/$themename/config.php")) {
    // Alternative theme location contains this theme - ok.
} else {
    css_send_css_not_found();
}

$theme = theme_config::load($themename);
$theme->force_svg_use($usesvg);

if ($type === 'editor') {
    $csscontent = $theme->get_css_content_editor();
    css_send_uncached_css($csscontent);
}

$chunkurl = new univer_url($CFG->httpswwwroot .
'/theme/styles_debug.php', array('theme' => $themename,
    'type' => $type, 'subtype' => $subtype, 'sheet' => $sheet,
    'usesvg' => $usesvg));
// We need some kind of caching here because otherwise the page
navigation becomes
// way too slow in theme designer mode. Feel free to create full
cache definition later...
$key = "$type $subtype $sheet $usesvg";
$cache = cache::make_from_params(cache_store::MODE_APPLICATION,
'core', 'themedesigner', array('theme' => $themename));
if ($content = $cache->get($key)) {
    if ($content['created'] > time() -
THEME_DESIGNER_CACHE_LIFETIME) {
        $csscontent = $content['data'];

        // We need to chunk the content.
        if ($chunk !== null) {
            $chunks = css_chunk_by_selector_count($csscontent,
$chunkurl->out(false));
            $csscontent = ($chunk === 0) ? end($chunks) :
$chunks[$chunk - 1];
        }
    }
}
```

Продолжение приложения A

```
        css_send_uncached_css($csscontent);
    }
}

$csscontent = $theme->get_css_content_debug($type, $subtype,
$sheet);
$cache->set($key, array('data' => $csscontent, 'created' =>
time()));

// We need to chunk the content.
if ($chunk !== null) {
    // The chunks are ordered so that the last chunk is the one
    containing the @import, and so
    // the first one to be included. All the other chunks are set
    in the array before that one.
    // See {@link css_chunk_by_selector_count()} for more details.
    $chunks = css_chunk_by_selector_count($csscontent, $chunkurl-
>out(false));
    $csscontent = ($chunk === 0) ? end($chunks) : $chunks[$chunk -
1];
}

css_send_uncached_css($csscontent);
require('../config.php');

$url          = required_param('url', PARAM_LOCALURL);
$newdevice    = required_param('device', PARAM_TEXT);

require_sesskey();

core_useragent::set_user_device_type($newdevice);
redirect($url);
define('NO_DEBUG_DISPLAY', true);

// we need just the values from config.php and minlib.php
define('ABORT_AFTER_CONFIG', true);
require('../config.php'); // this stops immediately at the
beginning of lib/setup.php

// get special url parameters

list($parts, $slasharguments) = combo_params();
if (!$parts) {
    combo_not_found();
}

$etag = sha1($parts);
$parts = trim($parts, '&');

// find out what we are serving - only one type per request
```


Продолжение приложения А

```
$content = '';
if (substr($parts, -3) === '.js') {
    $mimetype = 'application/javascript';
} else if (substr($parts, -4) === '.css') {
    $mimetype = 'text/css';
} else {
    combo_not_found();
}

// if they are requesting a revision that's not -1, and they have
// supplied an
// If-Modified-Since header, we can send back a 304 Not Modified
// since the
// content never changes (the rev number is increased any time the
// content changes)
if (strpos($parts, '/-1/') === false and
(!empty($_SERVER['HTTP_IF_NONE_MATCH']) ||
!empty($_SERVER['HTTP_IF_MODIFIED_SINCE']))) {
    $lifetime = 60*60*24*360; // 1 year, we do not change YUI
versions often, there are a few custom yui modules
    header('HTTP/1.1 304 Not Modified');
    header('Expires: ' . gmdate('D, d M Y H:i:s', time() +
$lifetime) . ' GMT');
    header('Cache-Control: public, max-age='.$lifetime);
    header('Content-Type: '.$mimetype);
    header('Etag: "'.$etag.'"');
    die;
}

$parts = explode('&', $parts);
$cache = true;
$lastmodified = 0;

while (count($parts)) {
    $part = array_shift($parts);
    if (empty($part)) {
        continue;
    }
    $filecontent = '';
    $part = min_clean_param($part, 'SAFEPATH');
    $bits = explode('/', $part);
    if (count($bits) < 2) {
        $content .= "\n// Wrong combo resource $part!\n";
        continue;
    }

    $version = array_shift($bits);
    if ($version === 'rollup') {
        $yuipatchedversion = explode('_', array_shift($bits));
        $revision = $yuipatchedversion[0];
    }
}
```

Продолжение приложения A

```
$rollupname = array_shift($bits);

if (strpos($rollupname, 'yui-universimple') !== false) {
    if (substr($rollupname, -3) === '.js') {
        // Determine which version of this rollup should
be used.
        $filesuffix = '.js';
        preg_match('/(- (debug|min))?\.\.js/', $rollupname,
$matches);
        if (isset($matches[1])) {
            $filesuffix = $matches[0];
        }

        $type = 'js';
    } else if (substr($rollupname, -4) === '.css') {
        $type = 'css';
    } else {
        continue;
    }

    // Allow support for revisions on YUI between official
releases.
    // We can just discard the subrevision since it is
only used to invalidate the browser cache.
    $yuipatchedversion = explode('_', $revision);
    $yuiversion = $yuipatchedversion[0];

    $yuimodules = array(
        'yui',
        'oop',
        'event-custom-base',
        'dom-core',
        'dom-base',
        'color-base',
        'dom-style',
        'selector-native',
        'selector',
        'node-core',
        'node-base',
        'event-base',
        'event-base-ie',
        'pluginhost-base',
        'pluginhost-config',
        'event-delegate',
        'node-event-delegate',
        'node-pluginhost',
        'dom-screen',
        'node-screen',
        'node-style',
```

Продолжение приложения А

```
'querystring-stringify-simple',
'io-base',
'json-parse',
'transition',
'selector-css2',
'selector-css3',
'dom-style-ie',

// Some extras we use everywhere.
'escape',

'attribute-core',
'event-custom-complex',
'base-core',
'attribute-base',
'attribute-extras',
'attribute-observable',
'base-observable',
'base-base',
'base-pluginhost',
'base-build',
'event-synthetic',

'attribute-complex',
'event-mouseenter',
'event-key',
'event-outside',
'event-autohide',
'event-focus',
'classnamemanager',
'widget-base',
'widget-htmlparser',
'widget-skin',
'widget-uievents',
'widget-stdmod',
'widget-position',
'widget-position-align',
'widget-stack',
'widget-position-constrain',
'overlay',

'widget-autohide',
'button-core',
'button-plugin',
'widget-buttons',
'widget-modality',
'panel',
'yui-throttle',
'dd-ddm-base',
```

Продолжение приложения А

```
'dd-drag',
'dd-plugin',

// Cache is used by univer-core-tooltip which we
include everywhere.
'cache-base',
);

// We need to add these new parts to the beginning of
the $parts list, not the end.
if ($type === 'js') {
    $newparts = array();
    foreach ($yuimodules as $module) {
        $newparts[] = $yuiversion . '/' . $module .
        '/' . $module . $filesuffix;
    }
    $newparts[] = 'yuiuseall/yuiuseall';
    $parts = array_merge($newparts, $parts);
} else {
    $newparts = array();
    foreach ($yuimodules as $module) {
        $candidate = $yuiversion . '/' . $module .
        '/assets/skins/sam/' . $module . '.css';
        if (!file_exists("$CFG-
>libdir/yuilib/$candidate")) {
            continue;
        }
        $newparts[] = $candidate;
    }
    if ($newparts) {
        $parts = array_merge($newparts, $parts);
    }
}

// Handle the mcore rollup.
if (strpos($rollupname, 'mcore') !== false) {
    $yuimodules = array(
        'core/tooltip/tooltip',
        'core/popuphelp/popuphelp',
        'core/widget-focusafterclose/widget-
focusafterclose',
        'core/dock/dock-loader',
        'core/notification/notification-dialogue',
    );
}

// Determine which version of this rollup should be
used.
$filesuffix = '.js';
```

Продолжение приложения A

```
preg_match('/(-(debug|min))?.js/', $rollupname,
$matches);
    if (isset($matches[1])) {
        $filesuffix = $matches[0];
    }

    // We need to add these new parts to the beginning of
the $parts list, not the end.
    $newparts = array();
    foreach ($yuimodules as $module) {
        $newparts[] = 'm/' . $revision . '/' . $module .
$filesuffix;
    }
    $parts = array_merge($newparts, $parts);
}
continue;
}
if ($version === 'm') {
    $version = 'skt';
}
if ($version === 'skt') {
    if (count($bits) <= 3) {
        // This is an invalid skt load attempt.
        $content .= "\n// Incorrect skt module inclusion. Not
enough component information in {$part}.\n";
        continue;
    }
    $revision = (int)array_shift($bits);
    if ($revision === -1) {
        // Revision -1 says please don't cache the JS
        $cache = false;
    }
    $frankenstyle = array_shift($bits);
    $filename = array_pop($bits);
    $modulename = $bits[0];
    $dir =
core_component::get_component_directory($frankenstyle);

    // For shifted YUI modules, we need the YUI module name in
frankenstyle format.
    $frankenstylemodulename = join('-', array($version,
$frankenstyle, $modulename));
    $frankenstylefilename = preg_replace('/' . $modulename .
'/', $frankenstylemodulename, $filename);

    // Submodules are stored in a directory with the full
submodule name.
    // We need to remove the -debug.js, -min.js, and .js from
the file name to calculate that directory name.
```

Продолжение приложения A

```
$frankenstyledirectoryname = str_replace(array('-min.js',
'-debug.js', '.js', '.css'), '', $frankenstylefilename);

// By default, try and use the /yui/build directory.
$contentfile = $dir . '/yui/build/' .
$frankenstyledirectoryname;
if ($mimetype == 'text/css') {
    // CSS assets are in a slightly different place to the
JS.
    $contentfile = $contentfile . '/assets/skins/sam/' .
$frankenstylefilename;

    // Add the path to the bits to handle fallback for
non-shifted assets.
    $bits[] = 'assets';
    $bits[] = 'skins';
    $bits[] = 'sam';
} else {
    $contentfile = $contentfile . '/' .
$frankenstylefilename;
}

// If the shifted versions don't exist, fall back to the
non-shifted file.
if (!file_exists($contentfile) or !is_file($contentfile))
{
    // We have to revert to the non-minified and non-debug
versions.
    $filename = preg_replace('/-(min|debug)\./', '.',
$filename);
    $contentfile = $dir . '/yui/' . join('/', $bits) . '/'
. $filename;
}
} else if ($version === '2in3') {
    $contentfile = "$CFG->libdir/yuilib/$part";
} else if ($version == 'gallery') {
    if (count($bits) <= 2) {
        // This is an invalid module load attempt.
        $content .= "\n// Incorrect univer module inclusion.
Not enough component information in {$part}.\n";
        continue;
    }
    $revision = (int)array_shift($bits);
    if ($revision === -1) {
        // Revision -1 says please don't cache the JS
        $cache = false;
    }
    $contentfile = "$CFG->libdir/yuilib/gallery/" . join('/',
$bits);
```

Продолжение приложения A

```
} else if ($version == 'yuiuseall') {
    // Create global Y that is available in global scope,
    // this is the trick behind original SimpleYUI.
    $filecontent = "var Y = YUI().use('*');"

} else {
    // Allow support for revisions on YUI between official
releases.
    // We can just discard the subrevision since it is only
used to invalidate the browser cache.
    $yuipatchedversion = explode('_', $version);
    $yuiversion = $yuipatchedversion[0];
    if ($yuiversion != $CFG->yui3version) {
        $content .= "\n// Wrong yui version $part!\n";
        continue;
    }
    $newpart = explode('/', $part);
    $newpart[0] = $yuiversion;
    $part = implode('/', $newpart);
    $contentfile = "$CFG->libdir/yuilib/$part";
}
if (!file_exists($contentfile) or !is_file($contentfile)) {
    $location = '$CFG->dirroot'.preg_replace('/^'.preg_quote($CFG->dirroot, '/').'/',' ', $contentfile);
    $content .= "\n// Combo resource $part ($location) not
found!\n";
    continue;
}

if (empty($filecontent)) {
    $filecontent = file_get_contents($contentfile);
}
$fmodified = filemtime($contentfile);
if ($fmodified > $lastmodified) {
    $lastmodified = $fmodified;
}

$relroot = preg_replace('|^http.?://[^\|]+|', '', $CFG->wwwroot);
$sep = ($slasharguments ? '/' : '?file=');

if ($mimetype === 'text/css') {
    if ($version == 'univer') {
        // Search for all images in the file and replace with
an appropriate link to the yui_image.php script
        $imagebits = array(
            $sep . $version,
            $frankenstyle,
```

Продолжение приложения A

```
    $modulename,  
    array_shift($bits),  
    '$1.$2'  
);  
  
    $filecontent = preg_replace('/([a-z0-9_-  
]+)\.(png|gif)/', $relroot . '/theme/yui_image.php' . implode('/',  
$imagebits), $filecontent);  
    } else if ($version == '2in3') {  
        // First we need to remove relative paths to images.  
These are used by YUI modules to make use of global assets.  
        // I've added this as a separate regex so it can be  
easily removed once  
        // YUI standardise there CSS methods  
        $filecontent =  
preg_replace('#(\.\.\./\.\.\./\.\.\./\.\.\./assets/skins/sam/)?([a-z0-9_-  
]+)\.(png|gif)#', '$2.$3', $filecontent);  
  
        // search for all images in yui2 CSS and serve them  
through the yui_image.php script  
        $filecontent = preg_replace('/([a-z0-9_-  
]+)\.(png|gif)/', $relroot.'/theme/yui_image.php'.$sep.$CFG-  
>yui2version.'/$1.$2', $filecontent);  
  
    } else if ($version == 'gallery') {  
        // Replace any references to the CDN with a relative  
link.  
        $filecontent = preg_replace('#(' .  
preg_quote('http://yui.yahooapis.com/') . '(gallery-[^\/*/]*)#',  
'../../../../../', $filecontent);  
  
        // Replace all relative image links with the a link to  
yui_image.php.  
        $filecontent = preg_replace('#(' .  
preg_quote('../../../../../') . ') (gallery-[^\/*/]*/assets/skins/sam/[a-  
z0-9_-]+)\.(png|gif)#',  
  
            $relroot . '/theme/yui_image.php' . $sep .  
'/gallery/' . $revision . '/$2.$3', $filecontent);  
  
    } else {  
        // First we need to remove relative paths to images.  
These are used by YUI modules to make use of global assets.  
        // I've added this as a separate regex so it can be  
easily removed once  
        // YUI standardise there CSS methods  
        $filecontent =  
preg_replace('#(\.\.\./\.\.\./\.\.\./\.\.\./assets/skins/sam/)?([a-z0-9_-  
]+)\.(png|gif)#', '$2.$3', $filecontent);
```


Продолжение приложения A

```
        // search for all images in yui2 CSS and serve them
        through the yui_image.php script
        $filecontent = preg_replace('/([a-z0-9_-
]+)\.(png|gif)/',
$relroot.'/theme/yui_image.php'.$sep.$version.'/'.$1.$2',
$filecontent);
    }
}

$content .= $filecontent;
}

if ($lastmodified == 0) {
    $lastmodified = time();
}

if ($cache) {
    combo_send_cached($content, $mimetype, $etag, $lastmodified);
} else {
    combo_send_uncached($content, $mimetype);
}

/**
 * Send the JavaScript cached
 * @param string $content
 * @param string $mimetype
 * @param string $etag
 * @param int $lastmodified
 */
function combo_send_cached($content, $mimetype, $etag,
$lastmodified) {
    $lifetime = 60*60*24*360; // 1 year, we do not change YUI
versions often, there are a few custom yui modules

    header('Content-Disposition: inline; filename="combo"');
    header('Last-Modified: '. gmdate('D, d M Y H:i:s',
$lastmodified) .' GMT');
    header('Expires: '. gmdate('D, d M Y H:i:s', time() +
$lifetime) .' GMT');
    header('Pragma: ');
    header('Cache-Control: public, max-age='.$lifetime);
    header('Accept-Ranges: none');
    header('Content-Type: '.$mimetype);
    header('Etag: "'.$etag.'");
    if (!min_enable_zlib_compression()) {
        header('Content-Length: '.strlen($content));
    }

    echo $content;
}
```

Продолжение приложения А

```
    die;
}

/**
 * Send the JavaScript uncached
 * @param string $content
 * @param string $mimetype
 */
function combo_send_uncached($content, $mimetype) {
    header('Content-Disposition: inline; filename="combo"');
    header('Last-Modified: '. gmdate('D, d M Y H:i:s', time()) .'
GMT');
    header('Expires: '. gmdate('D, d M Y H:i:s', time() + 2) .'
GMT');
    header('Pragma: ');
    header('Accept-Ranges: none');
    header('Content-Type: '.$mimetype);
    if (!min_enable_zlib_compression()) {
        header('Content-Length: '.strlen($content));
    }

    echo $content;
    die;
}

function combo_not_found($message = '') {
    header('HTTP/1.0 404 not found');
    if ($message) {
        echo $message;
    } else {
        echo 'Combo resource not found, sorry.';
    }
    die;
}

function combo_params() {
    if (isset($_SERVER['QUERY_STRING']) and
strpos($_SERVER['QUERY_STRING'], 'file=/') === 0) {
        // url rewriting
        $slashargument = substr($_SERVER['QUERY_STRING'], 6);
        return array($slashargument, true);

    } else if (isset($_SERVER['REQUEST_URI']) and
strpos($_SERVER['REQUEST_URI'], '?') !== false) {
        $parts = explode('?', $_SERVER['REQUEST_URI'], 2);
        return array($parts[1], false);

    } else if (isset($_SERVER['QUERY_STRING']) and
strpos($_SERVER['QUERY_STRING'], '?') !== false) {
```

Продолжение приложения A

```
// note: buggy or misconfigured IIS does return the query
string in REQUEST_URI
return array($_SERVER['QUERY_STRING'], false);

} else if ($slashargument = min_get_slash_argument()) {
    $slashargument = ltrim($slashargument, '/');
    return array($slashargument, true);

} else {
    // unsupported server, sorry!
    combo_not_found('Unsupported server - query string can not
be determined, try disabling YUI combo loading in admin
settings.');
```

```
}
```

```
}
```

```
define('NO_DEBUG_DISPLAY', true);
```

```
// we need just the values from config.php and minlib.php
```

```
define('ABORT_AFTER_CONFIG', true);
```

```
require('../config.php'); // this stops immediately at the
beginning of lib/setup.php
```

```
if ($slashargument = min_get_slash_argument()) {
```

```
    $path = ltrim($slashargument, '/');
```

```
} else {
```

```
    $path = min_optional_param('file', '', 'SAFEPATH');
```

```
}
```

```
$etag = sha1($path);
```

```
$parts = explode('/', $path);
```

```
$version = array_shift($parts);
```

```
if ($version === 'm') {
```

```
    $version = 'univer';
```

```
}
```

```
if ($version == 'univer' && count($parts) >= 3) {
```

```
    $frankenstyle = array_shift($parts);
```

```
    $module = array_shift($parts);
```

```
    $image = array_pop($parts);
```

```
    $subdir = join('/', $parts);
```

```
    $dir = core_component::get_component_directory($frankenstyle);
```

```
// For shifted YUI modules, we need the YUI module name in
frankenstyle format.
```

```
$frankenstylemodulename = join('-', array($version,
$frankenstyle, $module));
```

```
// By default, try and use the /yui/build directory.
```

```
$imagepath = $dir . '/yui/build/' . $frankenstylemodulename .
'/assets/skins/sam/' . $image;
```

Продолжение приложения A

```
// If the shifted versions don't exist, fall back to the non-
shifted file.
    if (!file_exists($imagepath) or !is_file($imagepath)) {
        $imagepath = $dir . '/yui/' . $module .
'/assets/skins/sam/' . $image;
    }
} else if ($version == 'gallery' && count($parts) >= 3) {
    list($revision, $module, , , , $image) = $parts;
    $imagepath = "$CFG-
>dirroot/lib/yuilib/gallery/$module/assets/skins/sam/$image";
} else {
    // Allow support for revisions on YUI between official
releases.
    // We can just discard the subrevision since it is only used
to invalidate the browser cache.
    $yuipatchedversion = explode('_', $version);
    $yuiversion = $yuipatchedversion[0];
    if (count($parts) == 1 && ($yuiversion == $CFG->yui3version ||
$yuiversion == $CFG->yui2version)) {
        list($image) = $parts;
        if ($yuiversion == $CFG->yui3version) {
            $imagepath = "$CFG->dirroot/lib/yuilib/$CFG-
>yui3version/assets/skins/sam/$image";
        } else {
            $imagepath = "$CFG->dirroot/lib/yuilib/2in3/$CFG-
>yui2version/build/assets/skins/sam/$image";
        }
    } else {
        yui_image_not_found();
    }
}

if (!file_exists($imagepath)) {
    yui_image_not_found();
}

$pathinfo = pathinfo($imagepath);
$imagename = $pathinfo['filename'].'.'.$pathinfo['extension'];
switch($pathinfo['extension']) {
    case 'gif' : $mimetype = 'image/gif'; break;
    case 'png' : $mimetype = 'image/png'; break;
    case 'jpg' : $mimetype = 'image/jpeg'; break;
    case 'jpeg' : $mimetype = 'image/jpeg'; break;
    case 'ico' : $mimetype = 'image/vnd.microsoft.icon'; break;
    default: $mimetype = 'document/unknown';
}

// if they are requesting a revision that's not -1, and they have
supplied an
```

Продолжение приложения А

```
// If-Modified-Since header, we can send back a 304 Not Modified
since the
// content never changes (the rev number is increased any time the
content changes)
if (strpos($path, '/-1/') === false and
(!empty($_SERVER['HTTP_IF_NONE_MATCH']) ||
!empty($_SERVER['HTTP_IF_MODIFIED_SINCE']))) {
    $lifetime = 60*60*24*360; // 1 year, we do not change YUI
versions often, there are a few custom yui modules
    header('HTTP/1.1 304 Not Modified');
    header('Last-Modified: '. gmdate('D, d M Y H:i:s',
filemtime($imagepath)) .' GMT');
    header('Expires: '. gmdate('D, d M Y H:i:s', time() +
$lifetime) .' GMT');
    header('Cache-Control: public, max-age='.$lifetime.', no-
transform');
    header('Content-Type: '.$mimetype);
    header('Etag: "'.$etag.'");
    die;
}
```

```
yui_image_cached($imagepath, $imagename, $mimetype, $etag);
```

```
function yui_image_cached($imagepath, $imagename, $mimetype,
$etag) {
    global $CFG;
    require("$CFG->dirroot/lib/xsendfilelib.php");

    $lifetime = 60*60*24*360; // 1 year, we do not change YUI
versions often, there are a few custom yui modules

    header('Content-Disposition: inline;
filename="'.$imagename.'");
    header('Last-Modified: '. gmdate('D, d M Y H:i:s',
filemtime($imagepath)) .' GMT');
    header('Expires: '. gmdate('D, d M Y H:i:s', time() +
$lifetime) .' GMT');
    header('Pragma: ');
    header('Cache-Control: public, max-age=315360000, no-
transform');
    header('Accept-Ranges: none');
    header('Content-Type: '.$mimetype);
    header('Content-Length: '.filesize($imagepath));
    header('Etag: "'.$etag.'");

    if (xsendfile($imagepath)) {
        die;
    }
}
```

Продолжение приложения A

```
// no need to gzip already compressed images ;-)  
  
readfile($imagepath);  
die;  
}  
  
function yui_image_not_found() {  
    header('HTTP/1.0 404 not found');  
    die('Image was not found, sorry.');}  
define('NO_DEBUG_DISPLAY', true);  
  
define('ABORT_AFTER_CONFIG', true);  
require('../config.php');  
if ($slashargument = min_get_slash_argument()) {  
    $slashargument = ltrim($slashargument, '/');  
    if (substr_count($slashargument, '/') < 3) {  
        font_not_found();  
    }  
    list($themename, $component, $rev, $font) = explode('/',  
$slashargument, 4);  
    $themename = min_clean_param($themename, 'SAFEDIR');  
    $component = min_clean_param($component, 'SAFEDIR');  
    $rev        = min_clean_param($rev, 'INT');  
    $font       = min_clean_param($font, 'RAW');  
} else {  
    $themename = min_optional_param('theme', 'standard',  
'SAFEDIR');  
    $component = min_optional_param('component', 'core',  
'SAFEDIR');  
    $rev       = min_optional_param('rev', -1, 'INT');  
    $font      = min_optional_param('font', '', 'RAW');}  
  
if (!$font) {  
    font_not_found();  
}  
  
if (empty($component) or $component === 'univer' or $component ===  
'core') {  
    $component = 'core';  
}  
  
if (preg_match('/^[a-z0-9_-]+\.woff$/i', $font, $matches)) {  
    // This is the real standard!  
    $font = $matches[0];  
    $mimetype = 'application/font-woff';  
}
```

Продолжение приложения A

```
} else if (preg_match('/^[a-z0-9_-]+\.ttf$/i', $font, $matches)) {
    $font = $matches[0];
    $mimetype = 'application/x-font-ttf';

} else if (preg_match('/^[a-z0-9_-]+\.otf$/i', $font, $matches)) {
    $font = $matches[0];
    $mimetype = 'application/x-font-opentype';

} else if (preg_match('/^[a-z0-9_-]+\.eot$/i', $font, $matches)) {
    // IE8 must die!!!
    $font = $matches[0];
    $mimetype = 'application/vnd.ms-fontobject';
} else if (preg_match('/^[a-z0-9_-]+\.svg$/i', $font, $matches)) {
    $font = $matches[0];
    $mimetype = 'image/svg+xml';

} else {
    font_not_found();
}

if (file_exists("$CFG->dirroot/theme/$themenamename/config.php")) {
    // Normal theme exists.
} else if (!empty($CFG->themedir) and file_exists("$CFG->
>themedir/$themenamename/config.php")) {
    // Theme exists in alternative location.
} else {
    font_not_found();
}

$candidatelocation = "$CFG->
>localcachedir/theme/$rev/$themenamename/fonts/$component";
$etag = sha1("$rev/$themenamename/$component/$font");

if ($rev > 0) {
    if (file_exists("$candidatelocation/$font.error")) {
        font_not_found();
    }

    if (file_exists("$candidatelocation/$font")) {
        if (!empty($_SERVER['HTTP_IF_NONE_MATCH']) ||
!empty($_SERVER['HTTP_IF_MODIFIED_SINCE'])) {
            // We do not actually need to verify the etag value
because our files
            // never change in cache because we increment the rev
parameter.
            $lifetime = 60*60*24*60; // 60 days only - the
revision may get incremented quite often.
            header('HTTP/1.1 304 Not Modified');
            header('Expires: '. gmdate('D, d M Y H:i:s', time() +
$lifetime) .' GMT');
```

Продолжение приложения А

```
header('Cache-Control: public, max-age='.$lifetime);
    header('Content-Type: '.$mimetype);
    header('Etag: "'.$etag.'");
    die;
}
    send_cached_font("$candidatelocation/$font", $etag, $font,
$mimetype);
}
}

// Ok, now we need to start normal univer script, we need to load
all libs and $DB.
define('ABORT_AFTER_CONFIG_CANCEL', true);

define('NO_UNIVER_COOKIES', true); // Session not used here.
define('NO_UPGRADE_CHECK', true); // Ignore upgrade check.

require("$CFG->dirroot/lib/setup.php");

$theme = theme_config::load($themename);
$themerev = theme_get_revision();

$fontfile = $theme->resolve_font_location($font, $component);

if ($themerev <= 0 or $rev != $themerev) {
    // Do not send caching headers if they do not request current
revision,
    // we do not want to pollute browser caches with outdated
fonts.
    if (empty($fontfile) or !is_readable($fontfile)) {
        font_not_found();
    }
    send_uncached_font($fontfile, $font, $mimetype);
}

make_localcache_directory('theme', false);

if (empty($fontfile) or !is_readable($fontfile)) {
    if (!file_exists($candidatelocation)) {
        @mkdir($candidatelocation, $CFG->directorypermissions,
true);
    }
    // Make note we can not find this file.
    $cachefont = "$candidatelocation/$font.error";
    $fp = fopen($cachefont, 'w');
    fclose($fp);
    font_not_found();
}
```


Продолжение приложения A

```
$cachefont = cache_font($font, $fontfile, $candidatelocation);
if (connection_aborted()) {
    die;
}
// Make sure nothing failed.
clearstatcache();
if (file_exists($cachefont)) {
    send_cached_font($cachefont, $etag, $font, $mimetype);
}

send_uncached_font($fontfile, $font, $mimetype);

// Utility functions.

function send_cached_font($fontpath, $etag, $font, $mimetype) {
    global $CFG;
    require("$CFG->dirroot/lib/xsendfilelib.php");

    $lifetime = 60*60*24*60; // 60 days only - the revision may
    get incremented quite often.

    header('Etag: "'. $etag. '"');
    header('Content-Disposition: inline; filename="'. $font. '"');
    header('Last-Modified: '. gmdate('D, d M Y H:i:s',
filetime($fontpath)) . ' GMT');
    header('Expires: '. gmdate('D, d M Y H:i:s', time() +
$lifetime) . ' GMT');
    header('Pragma: ');
    header('Cache-Control: public, max-age='. $lifetime);
    header('Accept-Ranges: none');
    header('Content-Type: '. $mimetype);
    header('Content-Length: '. filesize($fontpath));

    if (xsendfile($fontpath)) {
        die;
    }

    // No need to gzip already compressed fonts.
    die;
}

function send_uncached_font($fontpath, $font, $mimetype) {
    header('Content-Disposition: inline; filename="'. $font. '"');
    header('Last-Modified: '. gmdate('D, d M Y H:i:s', time()) . '
GMT');
    header('Expires: '. gmdate('D, d M Y H:i:s', time() + 15) . '
GMT');
    header('Pragma: ');
}
```

Продолжение приложения А

```
readfile($fontpath);

header('Accept-Ranges: none');
header('Content-Type: '.$mimetype);
header('Content-Length: '.filesize($fontpath));

readfile($fontpath);
die;
}

function font_not_found() {
    header('HTTP/1.0 404 not found');
    die('font was not found, sorry.');
```

```
}

/**
 * Caches a given font file.
 *
 * @param string $font The name of the font that was requested.
 * @param string $fontfile The location of the font file we want
to cache.
 * @param string $candidatelocation The location to cache it in.
 * @return string The path to the cached font.
 */
function cache_font($font, $fontfile, $candidatelocation) {
    global $CFG;
    $cachefont = "$candidatelocation/$font";

    clearstatcache();
    if (!file_exists($candidatelocation)) {
        @mkdir($candidatelocation, $CFG->directorypermissions,
true);
    }

    // Prevent serving of incomplete file from concurrent request,
// the rename() should be more atomic than copy().
    ignore_user_abort(true);
    if (@copy($fontfile, $cachefont.'.tmp')) {
        rename($cachefont.'.tmp', $cachefont);
        @chmod($cachefont, $CFG->filepermissions);
        @unlink($cachefont.'.tmp'); // Just in case anything
fails.
    }
    return $cachefont;
}
define('NO_DEBUG_DISPLAY', true);

// we need just the values from config.php and minlib.php
define('ABORT_AFTER_CONFIG', true);
```

Продолжение приложения А

```
require('../config.php'); // this stops immediately at the
beginning of lib/setup.php

if ($slashargument = min_get_slash_argument()) {
    $slashargument = ltrim($slashargument, '/');
    if (substr_count($slashargument, '/') < 3) {
        image_not_found();
    }
    if (strpos($slashargument, '_s/') === 0) {
        // Can't use SVG
        $slashargument = substr($slashargument, 3);
        $usesvg = false;
    } else {
        $usesvg = true;
    }
    // image must be last because it may contain "/"
    list($themename, $component, $rev, $image) = explode('/',
$slashargument, 4);
    $themename = min_clean_param($themename, 'SAFEDIR');
    $component = min_clean_param($component, 'SAFEDIR');
    $rev       = min_clean_param($rev, 'INT');
    $image     = min_clean_param($image, 'SAFEPATH');

} else {
    $themename = min_optional_param('theme', 'standard',
'SAFEDIR');
    $component = min_optional_param('component', 'core',
'SAFEDIR');
    $rev       = min_optional_param('rev', -1, 'INT');
    $image     = min_optional_param('image', '', 'SAFEPATH');
    $usesvg    = (bool)min_optional_param('svg', '1', 'INT');
}

if (empty($component) or $component === 'univer' or $component ===
'core') {
    $component = 'core';
}

if (empty($image)) {
    image_not_found();
}

if (file_exists("$CFG->dirroot/theme/$themename/config.php")) {
    // exists
} else if (!empty($CFG->themedir) and file_exists("$CFG->
themedir/$themename/config.php")) {
    // exists
} else {
    image_not_found();
}
}
```

Продолжение приложения A

```
$candidatelocation = "$CFG-
>localcachedir/theme/$rev/$themename/pix/$component";
$etag = sha1("$rev/$themename/$component/$image");

if ($rev > 0) {
    if (file_exists("$candidatelocation/$image.error")) {
        // This is a major speedup if there are multiple missing
images,
        // the only problem is that random requests may pollute
our cache.
        image_not_found();
    }
    $cacheimage = false;
    if ($usesvg && file_exists("$candidatelocation/$image.svg")) {
        $cacheimage = "$candidatelocation/$image.svg";
        $ext = 'svg';
    } else if (file_exists("$candidatelocation/$image.png")) {
        $cacheimage = "$candidatelocation/$image.png";
        $ext = 'png';
    } else if (file_exists("$candidatelocation/$image.gif")) {
        $cacheimage = "$candidatelocation/$image.gif";
        $ext = 'gif';
    } else if (file_exists("$candidatelocation/$image.jpg")) {
        $cacheimage = "$candidatelocation/$image.jpg";
        $ext = 'jpg';
    } else if (file_exists("$candidatelocation/$image.jpeg")) {
        $cacheimage = "$candidatelocation/$image.jpeg";
        $ext = 'jpeg';
    } else if (file_exists("$candidatelocation/$image.ico")) {
        $cacheimage = "$candidatelocation/$image.ico";
        $ext = 'ico';
    }
    if ($cacheimage) {
        if (!empty($_SERVER['HTTP_IF_NONE_MATCH']) ||
!empty($_SERVER['HTTP_IF_MODIFIED_SINCE'])) {
            // we do not actually need to verify the etag value
because our files
            // never change in cache because we increment the rev
parameter
            $lifetime = 60*60*24*60; // 60 days only - the
revision may get incremented quite often
            $mimetype = get_contenttype_from_ext($ext);
            header('HTTP/1.1 304 Not Modified');
            header('Expires: '. gmdate('D, d M Y H:i:s', time() +
$lifetime) .' GMT');
            header('Cache-Control: public, max-age='.$lifetime.',
no-transform');
            header('Content-Type: '.$mimetype);
            header('Etag: "'.$etag.'");
            die;
        }
    }
}
```

Продолжение приложения А

```
    }
    send_cached_image($cacheimage, $etag);
}
}

//=====
// ok, now we need to start normal univer script, we need to load
// all libs and $DB
define('ABORT_AFTER_CONFIG_CANCEL', true);

define('NO_UNIVER_COOKIES', true); // Session not used here
define('NO_UPGRADE_CHECK', true); // Ignore upgrade check

require("$CFG->dirroot/lib/setup.php");

$theme = theme_config::load($themename);
$themerev = theme_get_revision();

if ($themerev <= 0 or $rev != $themerev) {
    // Do not send caching headers if they do not request current
    // revision,
    // we do not want to pollute browser caches with outdated
    // images.
    $imagefile = $theme->resolve_image_location($image,
    $component, $usesvg);
    if (empty($imagefile) or !is_readable($imagefile)) {
        image_not_found();
    }
    send_uncached_image($imagefile);
}

make_localcache_directory('theme', false);

// We're not using SVG and there is no cached version of this file
// (in any format).
// As we're going to be caching a format other than svg, and
// because svg use is conditional we need to ensure that at the same
// time we cache a version of the SVG if it exists. If we don't do
// this other users who ask for SVG would not ever get it as
// there is a cached image already of another format.
// Remember this only gets run once before any candidate exists,
// and only if we want a cached revision.
if (!$usesvg) {
    $imagefile = $theme->resolve_image_location($image,
    $component, true);
    if (!empty($imagefile) && is_readable($imagefile)) {
        $cacheimage = cache_image($image, $imagefile,
    $candidatelocation);
        $pathinfo = pathinfo($imagefile);
    }
}
```

Продолжение приложения A

```
// There is no SVG equivalent, we've just successfully cached an
image of another format.
    if ($pathinfo['extension'] !== 'svg') {
        // Serve the file as we would in a normal request.
        if (connection_aborted()) {
            die;
        }
        // Make sure nothing failed.
        clearstatcache();
        if (file_exists($cacheimage)) {
            send_cached_image($cacheimage, $etag);
        }
        send_uncached_image($imagefile);
        exit;
    }
}

// Either SVG was requested or we've cached a SVG version and are
ready to serve a regular format.
$imagefile = $theme->resolve_image_location($image, $component,
$usesvg);
if (empty($imagefile) or !is_readable($imagefile)) {
    if (!file_exists($candidatelocation)) {
        @mkdir($candidatelocation, $CFG->directorypermissions,
true);
    }
    // Make note we can not find this file.
    $cacheimage = "$candidatelocation/$image.error";
    $fp = fopen($cacheimage, 'w');
    fclose($fp);
    image_not_found();
}

$cacheimage = cache_image($image, $imagefile, $candidatelocation);
if (connection_aborted()) {
    die;
}
// Make sure nothing failed.
clearstatcache();
if (file_exists($cacheimage)) {
    send_cached_image($cacheimage, $etag);
}

send_uncached_image($imagefile);

//=====
=====
//=== utility functions ===
```

Продолжение приложения А

```
// we are not using filelib because we need to fine tune all
header
// parameters to get the best performance.

function send_cached_image($imagepath, $etag) {
    global $CFG;
    require("$CFG->dirroot/lib/xsendfilelib.php");

    $lifetime = 60*60*24*60; // 60 days only - the revision may
get incremented quite often
    $pathinfo = pathinfo($imagepath);
    $imagename = $pathinfo['filename'].'.'.$pathinfo['extension'];

    $mimetype = get_contenttype_from_ext($pathinfo['extension']);

    header('Etag: "'.$etag.'"');
    header('Content-Disposition: inline;
filename="'.$imagename.'"');
    header('Last-Modified: '. gmdate('D, d M Y H:i:s',
filetime($imagepath)) .' GMT');
    header('Expires: '. gmdate('D, d M Y H:i:s', time() +
$lifetime) .' GMT');
    header('Pragma: ');
    header('Cache-Control: public, max-age='.$lifetime.', no-
transform');
    header('Accept-Ranges: none');
    header('Content-Type: '.$mimetype);
    header('Content-Length: '.filesize($imagepath));

    if (xsendfile($imagepath)) {
        die;
    }

    // no need to gzip already compressed images ;- )

    readfile($imagepath);
    die;
}

function send_uncached_image($imagepath) {
    $pathinfo = pathinfo($imagepath);
    $imagename = $pathinfo['filename'].'.'.$pathinfo['extension'];

    $mimetype = get_contenttype_from_ext($pathinfo['extension']);

    header('Content-Disposition: inline;
filename="'.$imagename.'"');
    header('Last-Modified: '. gmdate('D, d M Y H:i:s', time()) .'
GMT');
```

Продолжение приложения А

```
header('Expires: '. gmdate('D, d M Y H:i:s', time() + 15) .'
GMT');
header('Pragma: ');
header('Accept-Ranges: none');
header('Content-Type: '.$mimetype);
header('Content-Length: '.filesize($imagepath));

readfile($imagepath);
die;
}

function image_not_found() {
    header('HTTP/1.0 404 not found');
    die('Image was not found, sorry.');
```

```
}

function get_contenttype_from_ext($ext) {
    switch ($ext) {
        case 'svg':
            return 'image/svg+xml';
        case 'png':
            return 'image/png';
        case 'gif':
            return 'image/gif';
        case 'jpg':
        case 'jpeg':
            return 'image/jpeg';
        case 'ico':
            return 'image/vnd.microsoft.icon';
    }
    return 'document/unknown';
}

/**
 * Caches a given image file.
 *
 * @param string $image The name of the image that was requested.
 * @param string $imagefile The location of the image file we want
to cache.
 * @param string $candidatelocation The location to cache it in.
 * @return string The path to the cached image.
 */
function cache_image($image, $imagefile, $candidatelocation) {
    global $CFG;
    $pathinfo = pathinfo($imagefile);
    $cacheimage =
"$candidatelocation/$image.".$pathinfo['extension'];

    clearstatcache();
    if (!file_exists(dirname($cacheimage))) {
```


Продолжение приложения A

```
        @mkdir(dirname($cacheimage), $CFG->directorypermissions,
true);
    }

    // Prevent serving of incomplete file from concurrent request,
    // the rename() should be more atomic than copy().
    ignore_user_abort(true);
    if (@copy($imagefile, $cacheimage.'.tmp')) {
        rename($cacheimage.'.tmp', $cacheimage);
        @chmod($cacheimage, $CFG->filepermissions);
        @unlink($cacheimage.'.tmp'); // just in case anything
fails
    }
    return $cacheimage;
}
$THEME->name = 'more';
$THEME->parents = array('clean', 'bootstrapbase');

$THEME->doctype = 'html5';
$THEME->sheets = array('custom');
$THEME->lessfile = 'univer';
$THEME->parents_exclude_sheets = array('bootstrapbase' =>
array('univer'), 'clean' => array('custom'));
$THEME->lessvariablescallback = 'theme_more_less_variables';
$THEME->extralesscallback = 'theme_more_extra_less';
$THEME->supportscssoptimisation = false;
$THEME->yuicssmodules = array();
$THEME->enable_dock = true;
$THEME->editor_sheets = array();

$THEME->rendererfactory = 'theme_overridden_renderer_factory';
$THEME->csspostprocess = 'theme_more_process_css';

$THEME->blockrtlmanipulations = array(
    'side-pre' => 'side-post',
    'side-post' => 'side-pre'
);
function theme_more_extra_less($theme) {
    $content = '';
    $imageurl = $theme->setting_file_url('backgroundimage',
'backgroundimage');
    // Sets the background image, and its settings.
    if (!empty($imageurl)) {
        $content .= 'body { ';
        $content .= "background-image: url('$imageurl');";
        if (!empty($theme->settings->backgroundfixed)) {
            $content .= 'background-attachment: fixed;';
        }
        if (!empty($theme->settings->backgroundposition)) {
```

Продолжение приложения А

```
        $content .= 'background-position: ' . str_replace('_',
' ', $theme->settings->backgroundposition) . ';';
    }
    if (!empty($theme->settings->backgroundrepeat)) {
        $content .= 'background-repeat: ' . $theme->settings-
>backgroundrepeat . ';';
    }
    $content .= ' }';
}
// If there the user wants a background for the content, we
need to make it look consistent,
// therefore we need to round its borders, and adapt the
border colour.
if (!empty($theme->settings->contentbackground)) {
    $content .= '
        #region-main {
            .well;
            background-color: ' . $theme->settings-
>contentbackground . ';
            border-color: darken(' . $theme->settings-
>contentbackground . ', 7%);
        }';
}
return $content;
}

/**
 * Returns variables for LESS.
 *
 * We will inject some LESS variables from the settings that the
user has defined
 * for the theme. No need to write some custom LESS for this.
 *
 * @param theme_config $theme The theme config object.
 * @return array of LESS variables without the @.
 */
function theme_more_less_variables($theme) {
    $variables = array();
    if (!empty($theme->settings->bodybackground)) {
        $variables['bodyBackground'] = $theme->settings-
>bodybackground;
    }
    if (!empty($theme->settings->textcolor)) {
        $variables['textColor'] = $theme->settings->textcolor;
    }
    if (!empty($theme->settings->linkcolor)) {
        $variables['linkColor'] = $theme->settings->linkcolor;
    }
    if (!empty($theme->settings->secondarybackground)) {
```

Продолжение приложения A

```
        $variables['wellBackground'] = $theme->settings-
>secondarybackground;
    }
    return $variables;
}

/**
 * Parses CSS before it is cached.
 *
 * This function can make alterations and replace patterns within
the CSS.
 *
 * @param string $css The CSS
 * @param theme_config $theme The theme config object.
 * @return string The parsed CSS The parsed CSS.
 */
function theme_more_process_css($css, $theme) {

    // Set the background image for the logo.
    $logo = $theme->setting_file_url('logo', 'logo');
    $css = theme_more_set_logo($css, $logo);

    // Set custom CSS.
    if (!empty($theme->settings->customcss)) {
        $customcss = $theme->settings->customcss;
    } else {
        $customcss = null;
    }
    $css = theme_more_set_customcss($css, $customcss);

    return $css;
}

/**
 * Adds the logo to CSS.
 *
 * @param string $css The CSS.
 * @param string $logo The URL of the logo.
 * @return string The parsed CSS
 */
function theme_more_set_logo($css, $logo) {
    $tag = '[[setting:logo]]';
    $replacement = $logo;
    if (is_null($replacement)) {
        $replacement = '';
    }

    $css = str_replace($tag, $replacement, $css);

    return $css;
}
```

Продолжение приложения А

```
}

/**
 * Serves any files associated with the theme settings.
 *
 * @param stdClass $course
 * @param stdClass $cm
 * @param context $context
 * @param string $filearea
 * @param array $args
 * @param bool $forcedownload
 * @param array $options
 * @return bool
 */
function theme_more_pluginfile($course, $cm, $context, $filearea,
    $args, $forcedownload, array $options = array()) {
    if ($context->contextlevel == CONTEXT_SYSTEM && ($filearea ===
'logo' || $filearea === 'backgroundimage')) {
        $theme = theme_config::load('more');
        return $theme->setting_file_serve($filearea, $args,
$forcedownload, $options);
    } else {
        send_file_not_found();
    }
}

/**
 * Adds any custom CSS to the CSS before it is cached.
 *
 * @param string $css The original CSS.
 * @param string $customcss The custom CSS to add.
 * @return string The CSS which now contains our custom CSS.
 */
function theme_more_set_customcss($css, $customcss) {
    $tag = '[[setting:customcss]]';
    $replacement = $customcss;
    if (is_null($replacement)) {
        $replacement = '';
    }

    $css = str_replace($tag, $replacement, $css);

    return $css;
}
defined(SKT) || die;

if ($ADMIN->fulltree) {

    // @textColor setting.
    $name = 'theme_more/textcolor';
```

Продолжение приложения А

```
$title = get_string('textcolor', 'theme_more');
$description = get_string('textcolor_desc', 'theme_more');
$default = '#333366';
$setting = new admin_setting_configcolourpicker($name, $title,
$description, $default, null, false);
$setting->set_updatedcallback('theme_reset_all_caches');
$settings->add($setting);

// @linkColor setting.
$name = 'theme_more/linkcolor';
$title = get_string('linkcolor', 'theme_more');
$description = get_string('linkcolor_desc', 'theme_more');
$default = '#FF6500';
$setting = new admin_setting_configcolourpicker($name, $title,
$description, $default, null, false);
$setting->set_updatedcallback('theme_reset_all_caches');
$settings->add($setting);

// @bodyBackground setting.
$name = 'theme_more/bodybackground';
$title = get_string('bodybackground', 'theme_more');
$description = get_string('bodybackground_desc',
'theme_more');
$default = '';
$setting = new admin_setting_configcolourpicker($name, $title,
$description, $default, null, false);
$setting->set_updatedcallback('theme_reset_all_caches');
$settings->add($setting);

// Background image setting.
$name = 'theme_more/backgroundimage';
$title = get_string('backgroundimage', 'theme_more');
$description = get_string('backgroundimage_desc',
'theme_more');
$setting = new admin_setting_configstoredfile($name, $title,
$description, 'backgroundimage');
$setting->set_updatedcallback('theme_reset_all_caches');
$settings->add($setting);

// Background repeat setting.
$name = 'theme_more/backgroundrepeat';
$title = get_string('backgroundrepeat', 'theme_more');
$description = get_string('backgroundrepeat_desc',
'theme_more');;
$default = 'repeat';
$choices = array(
    '0' => get_string('default'),
    'repeat' => get_string('backgroundrepeatrepeat',
'theme_more'),
```

Продолжение приложения А

```
'repeat-x' => get_string('backgroundrepeatrepeatx',
'theme_more'),
'repeat-y' => get_string('backgroundrepeatrepeaty',
'theme_more'),
'no-repeat' => get_string('backgroundrepeatnorepeat',
'theme_more'),
);
$setting = new admin_setting_configselect($name, $title,
$description, $default, $choices);
$setting->set_updatedcallback('theme_reset_all_caches');
$settings->add($setting);

// Background position setting.
$name = 'theme_more/backgroundposition';
$title = get_string('backgroundposition', 'theme_more');
$description = get_string('backgroundposition_desc',
'theme_more');
$default = '0';
$choices = array(
    '0' => get_string('default'),
    'left_top' => get_string('backgroundpositionlefttop',
'theme_more'),
    'left_center' =>
get_string('backgroundpositionleftcenter', 'theme_more'),
    'left_bottom' =>
get_string('backgroundpositionleftbottom', 'theme_more'),
    'right_top' => get_string('backgroundpositionrighttop',
'theme_more'),
    'right_center' =>
get_string('backgroundpositionrightcenter', 'theme_more'),
    'right_bottom' =>
get_string('backgroundpositionrightbottom', 'theme_more'),
    'center_top' => get_string('backgroundpositioncentertop',
'theme_more'),
    'center_center' =>
get_string('backgroundpositioncentercenter', 'theme_more'),
    'center_bottom' =>
get_string('backgroundpositioncenterbottom', 'theme_more'),
);
$setting = new admin_setting_configselect($name, $title,
$description, $default, $choices);
$setting->set_updatedcallback('theme_reset_all_caches');
$settings->add($setting);

// Background fixed setting.
$name = 'theme_more/backgroundfixed';
$title = get_string('backgroundfixed', 'theme_more');
$description = get_string('backgroundfixed_desc',
'theme_more');
```

Продолжение приложения A

```
$setting = new admin_setting_configcheckbox($name, $title,
$description, 0);
$setting->set_updatedcallback('theme_reset_all_caches');
$settings->add($setting);

// Main content background color.
$name = 'theme_more/contentbackground';
$title = get_string('contentbackground', 'theme_more');
$description = get_string('contentbackground_desc',
'theme_more');
$default = '#FFFFFF';
$setting = new admin_setting_configcolourpicker($name, $title,
$description, $default, null, false);
$setting->set_updatedcallback('theme_reset_all_caches');
$settings->add($setting);

// Secondary background color.
$name = 'theme_more/secondarybackground';
$title = get_string('secondarybackground', 'theme_more');
$description = get_string('secondarybackground_desc',
'theme_more');
$default = '#FFFFFF';
$setting = new admin_setting_configcolourpicker($name, $title,
$description, $default, null, false);
$setting->set_updatedcallback('theme_reset_all_caches');
$settings->add($setting);

// Invert Navbar to dark background.
$name = 'theme_more/invert';
$title = get_string('invert', 'theme_more');
$description = get_string('invertdesc', 'theme_more');
$setting = new admin_setting_configcheckbox($name, $title,
$description, 1);
$setting->set_updatedcallback('theme_reset_all_caches');
$settings->add($setting);

// Logo file setting.
$name = 'theme_more/logo';
$title = get_string('logo', 'theme_more');
$description = get_string('logodesc', 'theme_more');
$setting = new admin_setting_configstoredfile($name, $title,
$description, 'logo');
$setting->set_updatedcallback('theme_reset_all_caches');
$settings->add($setting);
```

Приложение Б

Каталог функций программного обеспечения

№ п/п	Наименование (содержание) функций	Объем функций (строк исходного кода)		
		С использованием среды разработки приложений		
		Delphi (Borland)	C++ Builder (Borland)	Visual C++ (Microsoft)
1. Ввод, анализ входной информации, генерация кодов и процессор входного языка				
101	Организация ввода информации	100	110	150
102	Контроль, предварительная обработка и ввод информации	550		
103	Анализ входного языка (синтаксический и семантический)	630	660	980
104	Преобразование операторов входного языка и команды другого языка	1050	1050	980
105	Обработка входного заказа и формирование таблиц	750	900	1340
106	Преобразование входного языка в машинные команды (транслятор, препроцессор, макрогенератор)	1300	4300	5100
107	Синтаксический и семантический анализ входного языка и генерация кодов команд	8700	5400	5400
108	Процессор языка	3000	2300	2300
109	Организация ввода/вывода информации в интерактивном режиме	220	220	320
110	Организация ввода/вывода информации с сети терминалов	3680	3340	3200
110	Управление вводом/выводом	2700	2900	2400
2. Формирование, введение и обслуживание баз данных				
201	Генерация структуры базы данных	3450	3950	4300
202	Генерация структуры базы данных	1540	1610	2060
203	Формирование баз данных	1700	1750	2180
204	Обработка наборов и записей базы данных	2050	2350	2670
205	Обслуживание базы данных в пакетном режиме	1030	1100	1260
206	Обслуживание базы данных в интерактивном режиме	3800	4400	6950
207	Манипулирование данными	8400	8670	9550
208	Организация поиска и поиск в базе данных	5230	5460	5480
209	Реорганизация базы данных	130	190	220
210	Загрузки базы данных	3150	2950	2780
3. Формирование и обработка файлов				
301	Формирование последовательного файла	340	360	290
302	Сортировка файла	340	360	290
303	Автоматическая сортировка файлов	1040	1150	930

304	Обслуживание файлов	520	540	420
305	Обработка файлов	750	800	720
306	Обработка файлов в диалоговом режиме	2400	2600	3050
307	Совместная обработка группы файлов	4900	5300	6180
308	Управление файлами	5130	5380	5750
309	Формирование файла	1100	1080	1020
4. Генерация программ и ПО, а также настройка ПО				
401	Генерация рабочих программ	3680	4120	3360
402	Генерация программ по описанию пользователей	10870	12330	9880
403	Формирование служебных таблиц	570	620	1070
404	Система генерации ПО	5120	5340	4980
405	Система генерации ПО	250	300	370
5. Управление ПО, компонентами ПО и внешними устройствами				
501	Монитор ПО (управление работой компонентов)	350	360	740
502	Монитор системы (управление работой комплекса ПО)	3750	3880	7740
503	Управление внешними устройствами и объектами	6850	7340	5900
504	Обработка прерываний	890	730	540
505	Управление внешней памятью	250	210	200
506	Обработка ошибочных и сбойных ситуаций	430	410	410
507	Обеспечение интерфейса между компонентами	730	750	970
6. Отладка прикладных программ, обмен информацией между магнитным диском и магнитной лентой, вспомогательные программные функции				
601	Отладка прикладных программ в интерактивном режиме	4500	4700	4300
602	Обмен информацией между магнитным диском и магнитной лентой	*	*	*
603	Копирование наборов данных на магнитной ленте и восстановление	*	*	*
604	Справка и обучение	680	680	720
605	Вспомогательные и сервисные программы	460	490	580
7. Расчетные задачи, формирование и вывод на внешние носители (АЦПУ) документов сложной формы и файлов				
701	Матстатистика и прогнозирование	8370	9570	9320
702	Расчетные задачи (расчет режимов обр-ки)	12600	15300	14800
703	Расчет показателей	410	500	460
704	Процессор отчетов	1070	1230	3200
705	Формирование и вывод на внеш.носители	2650	2850	3500
706	Предварит.обработка и печать файлов	540	560	470
707	Графический вывод результатов	300		480
708	Интерактивный редактор текста	3800	3910	4540

Приложение В

Укрупненные нормы времени на разработку ПО в зависимости от уточненного объема ПО и группы сложности ПО

Объем ПО(строки исходного кода, LOC)	Категории сложности ПО			Категории сложности ПО
	1-я	2-я	3-я	
<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>
200	—	—	21	1
300	—	—	23	2
400	—	—	25	3
500	—	—	27	4
600	—	33	28	5
700	—	36	30	6
800	—	38	32	7
900	—	40	34	8
1000	51	43	36	9
1200	54	45	38	10
1400	57	48	40	11
1600	60	50	42	12
1800	64	54	45	13
2000	68	57	48	14
2200	73	61	51	15
2400	76	64	54	16
2600	81	68	57	17
2800	86	72	60	18
3000	91	76	64	19
3200	97	81	68	20
3400	103	86	72	21
3600	110	92	77	22
3800	117	98	82	23
4000	124	104	87	24
4200	133	111	93	25
4400	141	118	99	26
4600	151	126	105	27
4800	160	134	112	28
5000	170	142	119	29
5500	182	152	127	30
6000	194	162	135	31
6500	206	172	144	32
7000	220	184	154	33
7500	235	196	164	34
8000	252	210	175	35

Приложение Г

Дополнительные коэффициенты сложности ПО

Характеристика ПО	Значения K_c
1. Функционирование ПО в расширенной операционной среде (связь с другими ПО)	0,08
2. Интерактивный доступ	0,06
3. Обеспечение хранения, ведения и поиска данных в сложных структурах	0,07
4. Наличие у ПО одновременно нескольких характеристик по табл.Г4.1, приложение Г	
4.1 2 характеристики	0,12
4.2 3 характеристики	0,18
4.3 Свыше 3-х характеристик	0,26

Приложение Д

Значения поправочного коэффициента, учитывающего использование стандартных модулей типовых программ и ПО (K_T)

Степень охвата реализуемых функций разрабатываемого ПО стандартными модулями, типовыми программами и ПО	Значения K_T
1. От 60 % и выше	0,6
2. От 40 % до 60	0,7
3. От 20 % до 40 %	0,8
4. До 20 %	0,9
5. Типовые программы и ПО не используемые для реализации функций разрабатываемого ПО	1,0

Приложение Ж

Поправочные коэффициенты, учитывающие новизну ПО (K_H)

Категория Новизны	Степень новизны	Использование		Значение K_H
		На основе нового типа ПК	В среде новой ОС	
А	Принципиально новые ПО, не имеющие доступных аналогов	+	+	1,75
		-	+	1,6
		+	-	1,2
		-	-	1,0
Б	ПО, являющиеся развитием определенного параметрического ряда ПО	+	+	1,0
		-	-	0,9
		+	-	0,8
В	ПО, являющиеся развитием определенного параметрического ряда ПО, разработанных для ранее освоенных типов конфигурации ПК и ОС	-	-	0,7

Приложение 3

Оценка значений среднего машинного времени на отладку 100 строк исходного кода без применения ПО

Наименование подсистемы АС и СОД	Средний расход машинного времени, ч/100 строк кода
1. Общесистемные задачи: ведение линейных файлов, информационно-поисковые системы и информационно-справочные системы, сбор информации, ввод информации, расширение возможностей средств обработки данных, организация вычислительного процесса	12
2. Задачи расчетного характера	15
3. Оперативное управление производством, расчеты по Ценообразованию	7
4. Техническая подготовка производства, транспортное, ремонтное, энергетическое и инструментальное обслуживание производства	8
5. Бухгалтерский учет, финансовые расчеты, учет пенсий и пособий, учет страховых операций, качество продукции	13
6. Управление кадрами	13
7. Техничко-экономическое планирование	13
8. Материально-техническое снабжение, реализация и сбыт готовой Продукции	13