

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ
им. ГУМАРБЕКА ДАУКЕЕВА»
Кафедра IT – инжиниринг

«ДОПУЩЕН К ЗАЩИТЕ»
Зав. кафедрой PhD, доцент Досжанова А.А
_____ «____» _____ 2020 г.

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Разработка системы анализа поведения посетителей веб сайтов
(человек в веб среде)

Специальность 5В070400 – Вычислительная техника и программное
обеспечение

Выполнила: Макеев Д.М. Группа ВТ-16-2

Научный руководитель: PhD, профессор Картбаев Т.С.

Консультанты:

по экономической части: к.э.н., профессор Габелашвили К.Р

(учёная степень, звание, Ф.И.О.)

_____ «____» _____ 2020 г.

по безопасности жизнедеятельности: доцент Приходько Н.Г

(учёная степень, звание, Ф.И.О.)

_____ «____» _____ 2020 г.

по программному обеспечению: ст.преп. Майкотов М.Н

(учёная степень, звание, Ф.И.О.)

_____ «____» _____ 2020 г.

Нормоконтролер: ст.преп. Абсатарова Б.Р

(учёная степень, звание, Ф.И.О.)

_____ «____» _____ 2020 г.

Рецензент: _____

(учёная степень, звание, Ф.И.О.)

_____ «____» _____ 2020 г.

Алматы 2020

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ
им. ГУМАРБЕКА ДАУКЕЕВА»

Институт систем управления и информационных технологий

Кафедра IT-инжиниринг

Специальность 5В070400 – «Вычислительная техника и программное обеспечение»

ЗАДАНИЕ

на выполнение дипломного проекта

Студенту Макееву Данияру Мухтарұлы

Тема проекта: Разработка системы анализа поведения посетителей веб-сайтов (человек в веб-среде)

Утверждена приказом по университету № ___ от «___» _____ 2020 г.

Срок сдачи законченного проекта «___» _____ 2020 г.

Исходные данные к проекту (требуемые параметры результатов исследования (проектирования) и исходные данные объекта): руководство по программированию на PHP, интегрированная среда Plesk, данные преддипломной практики.

Перечень вопросов, подлежащих разработке в дипломном проекте, или краткое содержание дипломного проекта:

- а) анализ и исследование предметной области;
- б) проектирование приложения;
- в) реализация приложения;
- г) экономическое обоснование эффективности;
- д) вопросы безопасности жизнедеятельности и охраны труда.

Перечень графического материала (с точным указанием обязательных чертежей): имеется 13 таблиц, 34 иллюстрации и 1 диаграмма.

Основная рекомендуемая литература:

1 Никсон, Робин Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript и CSS / Робин Никсон. - М.: "Издательство "Питер", 2013. - 560 с.

2 Прохоренок, Николай HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера (+ CD-ROM) / Николай Прохоренок. - М.: БХВ-Петербург, 2012. - 912 с.

3 Дронов В. PHP, MySQL и Dreamweaver. Разработка интерактивных Web-сайтов; БХВ-Петербург - М., 2016. - 480 с.

Консультация по проекту с указанием относящихся к ним разделов проекта

Раздел	Консультант	Сроки	Подпись
Экономическая часть	Габелашвили К.Р	21.04.2020	
Безопасности жизнедеятельности	Приходько Н.Г	30.04.2020	
Программная часть	Майкотов М.Н	15.05.2020	
Нормконтролер	Абсатарова Б.Р	13.05.2020 – 18.05.2020	

ГРАФИК
подготовки дипломной работы (проекта)

Наименования разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечания
Анализ и исследование предметной области	17.02.2020 - 26.02.2020	
Проектирование приложения	27.02.2020 - 23.03.2020	
Программная реализация	24.03.2020 - 30.04.2020	

Дата выдачи задания « ____ » _____ 2020 г.

Заведующий кафедрой _____ А.А.Досжанова
(подпись)

Научный руководитель проекта _____ Т.С. Картбаев
(подпись)

Задание принял к исполнению студент _____ Д.М.Макеев
(подпись)

АҢДАТПА

Дипломдық жоба тақырыбы: " Веб-сайттарға кірушілердің (веб-ортадағы адам) мінез-құлқын талдау жүйесін құру."

Осы дипломдық жобада веб-ортадағы адамды талдау, веб сайттарда адамның мінез-құлқтарының мүдделерімен әдістерін зерттейтін формулаларын (сұлбаларын) тауып шығаратын ақпараттық жүйені әзірлеу жүзеге асқан.

Дипломдық жоба скрипт немесе блок-схема тілінде C++ және Java-script форматында орындалды. Дипломдық жобаны толық қамтылы дайындау үшін C++, Java script, HTML технологиялар қолданылды.

Сондай-ақ, бағдарламаны әзірлеу шығындары мен құнын экономикалық орындылығы мен тиімділігін жүргізіліп, өндірістік жарықтандыруды жақсарту бойынша іс-шаралар ұсынылды.

АННОТАЦИЯ

Тема дипломного проекта: «Разработка системы анализа поведения посетителей веб сайтов (человек в веб среде)»

В данном дипломном проекте реализована разработка информационной системы анализа человека в веб-среде, нахождение и выведение формул (схем) для изучения интересов и методов поведения человека на веб сайтах.

Дипломный проект был выполнен в формате скрипта или блок схемы на языке C++ и Java script. Для достижения цели разрабатываемого дипломного проекта были применены такие технологии, как C++, Java script, HTML.

Также был проведен экономический расчет затрат и стоимости разработки программы, оценка экономической целесообразности разрабатываемого проекта и были предложены мероприятия по улучшению производственного освещения.

ANNOTATION

The theme of the diploma project: "Development of a system for analyzing the behavior of website visitors (a person in a web environment)".

In the given diploma project development of information system of analysis of the person in a web environment, finding and output of formulas (schemes) for studying of interests and methods of behaviour of the person on web sites is realized.

The diploma project was carried out in the format of a script or a block of schemes in C++ and Java script. Such technologies as C++, Java script, HTML were used to achieve the goal of the diploma project being developed.

There was also an economic calculation of costs and expenses of program development, assessment of economic feasibility of the project under development and proposed measures to improve production lighting.

Содержание

Введение	11
1 Аналитика	12
1.1 Представление о web-приложении	12
1.2 Классификация web-приложений и их уникальности.....	14
2 Описание проекта.....	23
2.1 Язык гипертекстовой разметки, как основной язык Web-приложения	23
2.2 Язык PHP и его роль в разработке проекта	24
2.3 JavaScript.....	28
2.4 Notepad++	28
3 Проектирование и разработка программного продукта.....	32
3.1 Проектирование и разработка формул алгоритмов	32
3.2 Этапы разработки веб-приложения.....	34
3.3 Разработка дизайна Web-сайта	40
3.5 Проектирование и разработка кода сайта	43
4 Экономический расчет	51
4.1 Цели и задачи, решаемые в экономической части	51
4.2 Расчет трудоемкости разработки ПП	51
4.3 Расчет затрат на разработку ПП	51
4.7 Расчет эксплуатационных затрат	59
4.8 Расчет результатов от создания и использования ИС	61
4.9 Расчет основных показателей экономической эффективности	63
5 Безопасность жизнедеятельности.....	65
5.1 Анализ потенциально опасных и вредных факторов в офисе, воздействующих на персонал.	65
5.2 Методика расчета установок водяного и пенного пожаротушения	68
Заключение	77
Список литературы	78
Приложение А	80
Приложение Б.....	82
Приложение В	92

Введение

Современные маркетинговые кампании немислимы без рекламы в сети Интернет. Огромное количество информации, размещаемое на новостных и развлекательных порталах, на сайтах известных издательств и телевизионных компаний привлекает массы посетителей. И почти всегда сопровождается рекламой важных событий, новостей, товаров, услуг.

Любая рекламная кампания становится результативной только в том случае, если она соответствует поставленной перед ней цели. Основными же целями любого рекламного проекта являются внедрение на рынок новых товаров и услуг, а также вовлечение потенциальных потребителей и заключение с ним сделки [4].

Существующие на данный момент сервисы доставки рекламы, такие как Google DoubleClick, 24/7 Realmedia OAS, AdTech Helios, Google AdSense, Yandex.Direct, Begun и пр. специализируются на выполнении целей рекламных кампаний, стараясь использовать все имеющиеся ресурсы интернет страниц. Эти сервисы предоставляют оператору определённый набор инструментов, позволяющих управлять рекламными кампаниями и принимать решения по оптимизации их хода. Имеющиеся статистические данные дают возможность проводить детальный анализ сложившейся ситуации.

Для достижения целей рекламных кампаний сервисы доставки рекламы позволяют настраивать их на различные поведенческие характеристики посетителей сайтов. В связи с этим возникает необходимость оценки и анализа поведенческих характеристик пользователя, просматривающих рекламу, а также их прогнозирование при изменении рекламной сети. Однако наличие человеческого фактора, зашумлённости данных, а также высокая степень зависимости хода рекламных кампаний друг от друга не позволяет операторам вручную принимать эффективные решения по управлению и снижению рисков.¹ Обзор и аналитика современного состояния вопроса.

1 Аналитика

1.1 Представление о web-приложении

Web-приложение является приложение которое используется в интернете с непосредственным подключением к нему. Веб-приложение - клиент-серверное приложение, в котором клиент взаимодействует с сервером при помощи браузера, а за сервер отвечает - веб-сервер. Логика веб-приложения распределена между сервером и клиентом, хранение данных осуществляется, преимущественно, на сервере, обмен информацией происходит по сети. Одним из преимуществ такого подхода является тот факт, что клиенты не зависят от конкретной операционной системы пользователя, поэтому веб-приложения являются межплатформенными службами.

Веб-приложения стали широко использоваться в конце 1990-х - начале 2000-х годов.

Веб-приложение - это любой сайт с элементами интерактива. Это значит, что посетитель может взаимодействовать с материалом, функциями: нажимать кнопки, заполнять формы, запрашивать прайс, совершать покупки.

Практически любой интернет-ресурс входит в их число. Это поисковые системы, видео сервисы типа youtube, соцсети, любые веб-сайты с функциями аутентификации пользователя, покупки, заказа, бронирования, калькуляторы кредитов.

Web-приложения (веб-приложения, Web-application) - вспомогательные программные средства, предназначенные для автоматизированного выполнения действий на Web-серверах. Они используют в качестве пользовательских интерфейсов Web-браузеры. Обычно Web-приложения создаются в разных вариантах архитектуры клиент-сервер. К числу средств создания Web-приложений относятся ISAPI, CGI, ASP, JSP, WAP. За годы существования WWW состав Web-приложений, выполняемые ими функции, принципы и архитектура их построения претерпели значительные изменения - от простейших средств хранения HTML-страниц до решений, ориентированных на поддержку работы корпоративными информационными систем и их партнеров. Виды Веб-приложений и технологий их создания:

AJAX (Asynchronous JavaScript and XML) - подход к построению пользовательских интерфейсов Web-приложений, при котором в ответ на каждое действие пользователя Web-страница на его браузере, не перезагружается полностью - с Web-сервера только догружаются нужные ему данные. Этим обеспечивается оперативная работа как одного, так и групп пользователей с приложениями. AJAX представляет собой не одну, а группу технологий и базируется на принципах использования DHTML для динамического изменения содержания страницы и использования

XMLHttpRequest для обращения к серверу (альтернативный вариант - динамическая подгрузка JavaScript с использованием объектной модели документа - DOM). С учетом этих принципов можно создавать удобные Web-интерфейсы на тех страницах сайтов, где необходимо активное взаимодействие с пользователями. Популярность AJAX приобрела после того, как компания Google начала применять его при создании Gmail, Google maps, Google suggest.

ASP (Active Server Pages) - технология создания Web-приложений, использующая объектную модель интерфейса, созданного на основе ISAPI-фильтра. ASP упростила задачи генерации HTML-страниц и позволила производить обращение к компонентам баз данных. Принцип, заложенный в основу интерфейса приложения, заключается в том, что на Web-странице присутствуют фрагменты кода, который интерпретируется Web-сервером и предоставляет пользователю готовый результат выполнения выбранных фрагментов кода. Web-страница, созданная с использованием технологии ASP, имеет расширение «.asp».

CGI (Common Gateway Interface, общий шлюзовой интерфейс) - программа поиска в удаленных БД, переадресации ссылок, использования графических меню, связи с базами данных (путем запуска программы преобразования форматов баз данных в формат языка HTML).

CRM (Customer Relationship Management) - Web-приложения для автоматизации и повышения эффективности процессов, связанных с бизнесом (обработка заказов, маркетинг, обслуживание клиентов). CRM используются в специализированных операторских «контакт-центрах». Первая версия программного продукта Microsoft CRM появилась в 2002 году. Web-сервисы Microsoft CRM реализуются на основе использования SQL-сервера и предусматривают создание основного хранилища данных Microsoft CRM, БД метаданных, БД для построения отчетности и дистрибуционной БД, предназначенной для отслеживания взаимодействия автономных пользователей клиента Outlook с основной БД Microsoft CRM. Использование XML позволяет интегрировать Microsoft CRM с приложениями подобного назначения независимо от языка программирования и операционной системы, под управлением которой работает стороннее приложение (например, SAP R/3). Система предусматривает ограничения доступа и проверку прав доступа клиентов.

ERP (Enterprise Resource Planning) - Web-приложения, предназначенные для автоматизации процессов управления внутрихозяйственной деятельностью корпорации, включая управление производством, финансами, снабжением, персоналом.

ISAPI (Internet Server Application Programming Interface) - интерфейс к серверу Интернета фирмы Microsoft, предназначен для программного управления сервером. ISAPI поддерживается большинством производителей программных средств. ISAPI-программы представляют собой специальный вид

приложений, обрабатывающих пользовательские запросы и отображающих их вывод в виде потока HTML, который поступает непосредственно в браузер клиента.

ITRP (IT Resources Planning) - класс Web-приложений, предназначен для поддержки управления корпоративными ИТ-ресурсами и сервисами.

JSP (Java Server Pages) - технология создания Web-приложений, основанная на однократной компиляции Java-кода (сервлета) при первом обращении к нему с последующим выполнением методов этого сервлета и помещением полученных результатов в набор данных, которые отправляются в браузер.

OSS (Operation Support Systems) - вид Web-приложений, предназначен для обеспечения работы операторов распределенных вычислительных сетей. OSS обеспечивает управление сетью, производительность, ликвидацию сбоев в работе, создание и учет сервисов, планирование сетевых ресурсов, мониторинг процессов, контроль за безопасностью, качество услуг и уровень обслуживания клиентов, сбором статистических данных. Разновидностью OSS является система поддержки бизнеса - BSS (Business Support Systems). К ним относятся биллинговые системы, системы управления взаимоотношениями с клиентами, управления сетями, заказами, качеством услуг.

PHP (Personal Home Page) - сценарный язык и программное средство для создания Web-страниц. В его состав входит CGI-интерфейс, интерпретатор языка и набор функций для доступа к базам данных и различным объектам WWW. PHP позволяет формировать страницы в режиме интерактивного взаимодействия в системах «клиент-сервер».

1.2 Классификация web-приложений и их уникальности

Распределенная архитектура, как наиболее подходящая для Business Web Application (BWA).

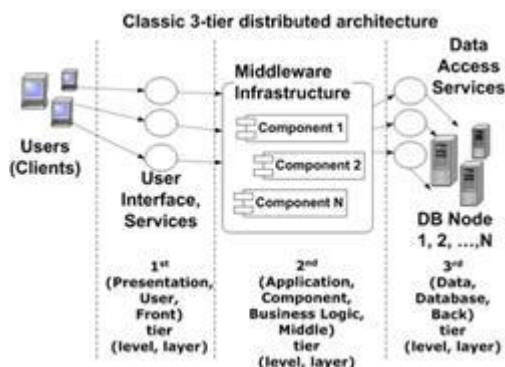


Рисунок 1.1 – Архитектура BWA

Три уровня (с позиции программирования) - это *хранение, обработка и представление* информации. Идея заключается в том, чтобы не смешивать эти три составляющие.

Виды распределенной архитектуры:

1) 2-х уровневая архитектура (Клиент-Сервер)

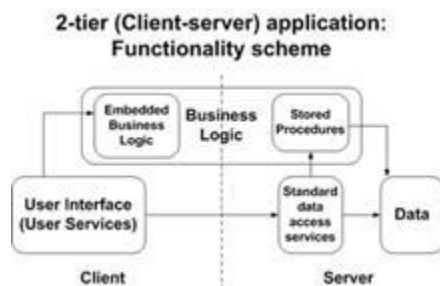


Рисунок 1.2 - 2-х уровневая архитектура

В классической архитектуре клиент-сервер, бизнес-логика может располагаться как у клиента, так и на сервере. В результате существует тенденция смешивания бизнес-логики с интерфейсом пользователя и/или со структурой БД.

2) 3-х уровневая архитектура (идеальный вариант)

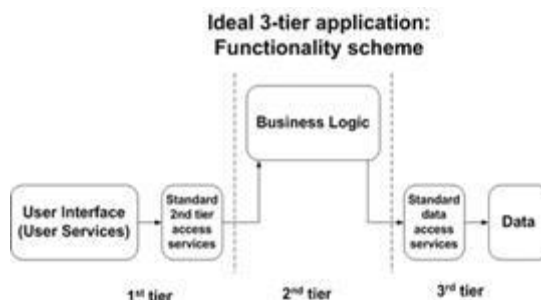


Рисунок 1.3 - 3-х уровневая архитектура (идеальный вариант)

3-х уровневая архитектура подразумевает четкое выделение бизнес-логики. Сервисы для вызова бизнес-логики (2nd tier), относятся к 1st tier. А интерфейс с базой данных (или любым источником данных) - сразу к 3rd tier. Данный подход позволяет четко очертить границы уровня бизнес-логики и отличие 3-х уровневого подхода от 2-х уровневого (клиент-сервер).

3) 3-х уровневая архитектура (фактический вариант)

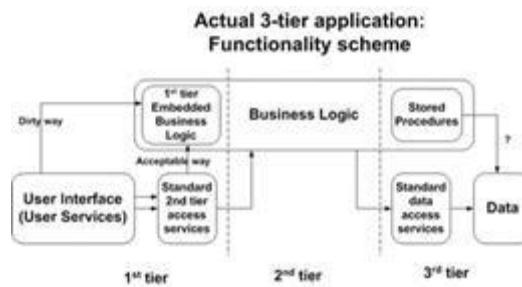


Рисунок 1.4 - 3-х уровневая архитектура (фактический вариант)

На практике по различным причинам бизнес-логика остается на 1st и 3rd tier. Это может быть оправдано, если не приведет к перегрузке уровня, на котором находится ненормированный код. И если этот код легко можно перенести в 2nd tier объекты. Например, поместив бизнес-логику в хранимые процедуры в 3rd tier, можно получить большой выигрыш в производительности. Если же эти хранимые процедуры написаны на Java, то перенести их в 2nd tier будет несложно. А вот бизнес-логика на 1st tier распространена повсеместно, и, обычно, связана с неудачной архитектурой.

Основные критерии идеальной распределенной архитектуры:

1) Каждый уровень распределенного приложения может взаимодействовать только со смежным уровнем. Это означает, что: 1st tier не должен иметь прямого доступа к 3rd tier и наоборот; 2nd и 3rd tiers не должны иметь прямого интерфейса с пользователем; обращение к источнику данных из 1st tier происходит только через объекты 2-nd tier;

2) Вся сложная бизнес-логика находится в объектах 2nd tier. Как уже упоминалось, это условие часто нарушается, снижая возможность масштабирования. Хотя подобные решения иногда оправданы.

3) Взаимодействие уровней организовано так, чтобы они могли взаимодействовать по сети, находясь физически на различных компьютерах. Это означает, что распределенная архитектура не зависит от способа развертывания приложения - все уровни могут быть размещены физически как одном компьютере, так и на разных, в условиях заданной сетевой структуры.

4) Сущности данных независимы от способа их хранения, уникально идентифицируемы по какому-либо ключу, и независимы от способа и места хранения других сущностей. Это, в частности, иногда означает отсутствие в некоторых таблицах связей по внешним ключам, что позволяет хранить группы таблиц, имеющих отношение к одной сущности, в разных базах данных. Однако тогда необходимо отслеживать целостность данных дополнительным кодом.

Нативную (native) разработку можно назвать «родной» для операционных систем - Android, IOS, Win Phone и т.д. Такие мобильные приложения пишутся на языках программирования, утвержденных

разработчиками программного обеспечения под каждую конкретную платформу, а потому органично встраиваются в сами операционные системы. Приложения загружаются через магазины приложений (App Store, Google Play и т.д.) и соответствуют требованиям этих магазинов.

Главное преимущество нативных приложений - то, что они оптимизированы под конкретные операционные системы, а значит работают корректно и быстро. Также они имеют доступ к аппаратной части устройств, то есть могут использовать в своем функционале камеру смартфона, микрофон, акселерометр, геолокацию, адресную книгу, плеер и т.д. Можно настроить получение push-уведомлений. Еще один плюс - экономный расход ресурсов телефона (батарея, память).

Нативные приложения могут полностью или частично работать и при отсутствующем интернет-соединении, поэтому пользователи менее зависят от качества связи и могут пользоваться приложением там и тогда, когда им это удобно.

Разумеется, написание такого продукта требует от разработчика владение специальными знаниями и умениями для работы в конкретной среде разработки (xCode для iPhone, Android Studio для устройств на Android). Как следствие стоимость таких приложений гораздо выше в силу их трудоемкости и того, что под каждую платформу приходится писать отдельное приложение на другом языке.

Генераторы мобильных приложений позволяют создавать кроссплатформенные приложения приближенные по функционалу и качеству к нативным приложениям. Это нечто среднее между нативными и веб-приложениями. Такие приложения устанавливаются через официальные магазины, имеют ограниченный доступ к аппаратной части смартфонов и планшетов, в них можно настраивать push-уведомления. А также они, как правило, дешевле нативных приложений.

Качество и возможности гибридных приложений зависят от самого фреймворка, которым пользовался разработчик. Есть более дешевые и более дорогие (приближенные к нативным по стоимости и срокам) варианты.

Основные недостатки кроссплатформенных приложений:

- ограничения объема хранимых в мобильном приложении данных, от которых зависит как часто приложению придется что-то докачивать через интернет (html5 – 50 Мб);
- невозможность использовать общий поиск мобильного телефона;
- невозможность узнать тип сетевого соединения (GPRS, 3G, LTE, WiFi - зависит от фреймворка);
- ограничения, накладываемые браузером операционной системы на доступ к акселерометру, гироскопу, геопозиционированию, видеозахвату;
- невозможность явной работы с файловой системой (создание и управление файлами и папками);

- ненативный интерфейс, который выглядит одинаково на всех операционных системах, или его приходится реализовывать отдельно под каждую из платформ;

- необходимость создавать различные компоненты с нуля (к примеру, выезжающее меню или анимацию), тогда как в нативных приложениях быстрее и проще использовать готовые компоненты;

- сложнее процесс оптимизации под различные размеры экранов устройств, чем у нативных приложений.

Выбирайте тип мобильного приложения исходя из целей такой разработки. Для начала решите зачем вам мобильное приложение, что его функционал даст вашей компании. Не забывайте, что если вы создаете приложение для ваших клиентов, то важно сделать его максимально удобным для использования, а не для того, чтобы оно было. Полагаем, что вы не хотите, чтобы ваши клиенты получали негативные впечатления и удаляли его вскоре после скачивания, поскольку даже если вы переделаете его или напишете заново уже более качественным, вам будет сложно убедить их скачать ваше мобильное приложение снова.

Если вам нужно емкое, скоростное и высококачественное приложение, функции которого довольно сложны и используют мощности устройств – вам к разработчику нативных приложений либо к разработчику, использующему фреймворк Xamarin. По стоимости это будет примерно одинаково.

Если у ваших клиентов всегда есть доступ к высокоскоростному интернету, а требуемый функционал приложения минимален, к примеру только просмотр данных, то вы можете обойтись и веб-приложением. По сути это будет адаптированный под экраны мобильных устройств интерфейс для просмотра информации с вашего сайта.

Для создания качественного web-приложения лучше всего будет приносить людям пользу. В интернете присутствует очень большое количество приложений обучающих чему-либо и когда разработчики таких приложений делают это качественно, такие приложения пользуются спросом, потому что это людям приносит пользу. По средствам различных программ пользователи в нынешнее время извлекают пользу в различных сферах. Это может быть, как обучение математики, проверка орфографии, правописания, обучение новому языку или освоению грамматики этого языка. Если поддерживать такое приложение своевременно устранять неполадки и обновлять информацию, то такое приложение как правило будет пользоваться долгим и большим спросом.

Правилами создания web-приложения можно назвать:

- удобный интерфейс;
- высокоскоростной отклик программы;
- информативность;
- поддержка программного продукта.

При соблюдении этих правил можно добиться высокого спроса на свой программный продукт.

Видами web-приложений могут быть как сайты, так и мобильные приложения. В большинстве случаев в Казахстане образовательные центры используют сайты ознакомительного характера. Для предоставления полной информации они используют систему оставления заявки и через дополнительные сервисы по средствам электронной почты они предоставляют свою полную информацию. Так же присутствуют сайты информативного характера. Такие сайты используются в основном для того, чтобы дать небольшую информацию, либо же совет по сдаче экзамена. Все больше и больше набирают популярность у образовательных центров языкового характера мобильные веб-приложения. Создав продукт, они обучают в большинстве случаев английскому языку. Чаще всего это заранее записанные произношения слов, написание транскрипций и т.д., но никто не использует свои приложения, чтобы передавать к примеру задания, проводить онлайн-уроки.

По сути это мобильная версия сайта только с расширенным функционалом. Но разница между веб-приложением и адаптивной версткой сайта не велика, поскольку и там и там применяются стандартные веб-технологии, а скорость работы ограничена качеством интернет-соединения. При этом веб-приложения не размещаются в специализированных магазинах приложений и обычно используют браузер телефона для работы.

Бывает, что такое приложение даже скачивается через официальные магазины (оболочка написана в нативной среде разработки и представляет собой встроенный браузер, а контент и интерфейс реализованы с помощью веб-технологии) и имеет свою иконку на экран смартфона. Однако для пользователя всегда будет очевидно, что его качество не соответствует полноценной нативной разработке. Срок жизни таких приложений на устройствах пользователей не велик, а отзывы чаще всего отрицательные. Когда речь идет об имидже компании, то уж лучше не иметь приложение совсем, чем такое, которое отвратит клиентов от пользования услугами данной компании.

Есть еще пара нюансов – веб приложения не всегда безопасны, поскольку не могут шифровать файловую систему, а если в дальнейшем потребуются обновления и расширение функционала, то придется писать приложение заново.

Из основных плюсов такого решения – кроссплатформенность, быстрые сроки реализации и низкая стоимость.

Давайте для простоты будем пока считать, что веб-приложение и веб-сайт - одно и то же. Не всегда веб-приложение является сайтом, и наоборот, но пока нет задачи вдаваться в подробности, поскольку рассматривать будем самый простой вариант.

Чаще всего веб-приложения состоят как минимум из трёх основных компонентов:

Клиентская часть веб приложения - это графический интерфейс. Это то, что вы видите на странице. Графический интерфейс отображается в браузере. Пользователь взаимодействует с веб-приложением именно через браузер, кликая по ссылкам и кнопкам.

Серверная часть веб-приложения - это программа или скрипт на сервере, обрабатывающая запросы пользователя (точнее, запросы браузера). Чаще всего серверная часть веб-приложения программируется на PHP. При каждом переходе пользователя по ссылке браузер отправляет запрос к серверу. Сервер обрабатывает этот запрос, вызывая некоторый PHP-скрипт, который формирует веб-страничку, описанную языком HTML, и отправляет клиенту по сети. Браузер тут же отображает полученный результат в виде очередной веб-страницы.

База данных (БД, или система управления базами данных, СУБД) - программное обеспечение на сервере, занимающееся хранением данных и их выдачей в нужный момент. В случае форума или блога, хранимые в БД данные - это посты, комментарии, новости, и так далее. База данных располагается на сервере. Серверная часть веб-приложения (то есть, PHP скрипт) обращается к базе данных, извлекая данные, которые необходимы для формирования страницы, запрошенной пользователем.

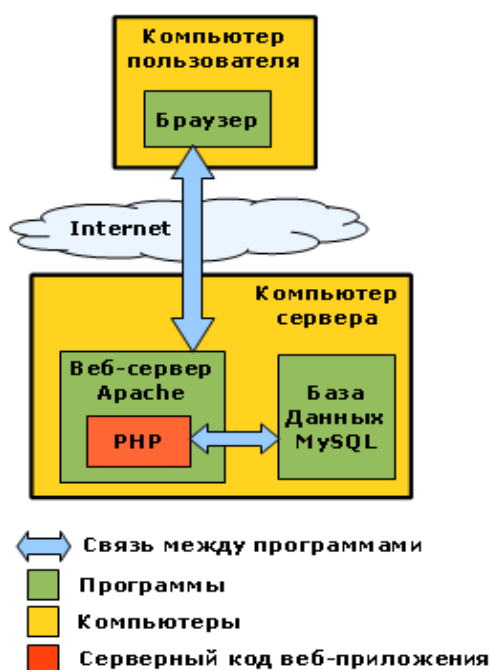


Рисунок 1.5 - Схема взаимодействия веб приложений

Это, пожалуй, основные компоненты большинства веб-приложений. Графически схему их взаимодействия можно представить так:

- Браузер через Интернет отправляет HTTP-запросы веб-серверу
- Веб-сервер вызывает PHP-скрипт, написанный разработчиком веб-приложения
- PHP-скрипт обращается к базе данных, если это нужно
- В результате PHP-скрипт возвращает клиенту веб-страницу, которую и отображает браузер.

Язык программирования для серверной части

Для программирования серверной части веб приложения может использоваться язык PHP, как в описанном примере, но не обязательно он. Для разработки веб-приложений можно использовать практически любые современные языки программирования:

- PHP, Perl;
- Ruby;
- Java;
- платформа .NET (языки VB.NET, C# и другие, поддерживаемые .NET);
- C/C++.

Независимо от языка, на котором написана серверная часть веб-приложения, способы обработки запросов и взаимодействия с пользователем остаются те же.

В дальнейшем в ходе освоения веб-технологий будем опираться на язык PHP, поскольку он является самым распространённым и самым простым среди всех перечисленных. Конфигурация сервера, включающая себя операционную систему Linux, язык программирования PHP, веб-сервер Apache и СУБД MySQL считается стандартом де-факто для среднестатистического сервера и даже носит название **LAMP** (от сокращения Linux, Apache, MySQL, PHP).

Язык программирования клиентской части

Основной язык, которым описывается графический интерфейс веб-приложения - это HTML. Этот язык описывает структуру веб-страницы. Художественное оформление веб страниц описывается таблицами стилей - CSS.

Для "оживления" графического интерфейса используются дополнительные технологии: скрипты JavaScript, а также встроенные в веб-страницу компоненты, созданные на Flash, Java или Silverlight. Все эти элементы веб-страницы могут взаимодействовать между собой: программа, написанная на JavaScript и выполняющаяся на веб-странице, может управлять встроенными в страницу компонентами, тем самым реализуя пользовательский интерфейс с богатыми возможностями.

Самым большим плюсом клиентских языков программирования является то, что нет необходимости отправлять веб-документ для обработки на сервер. Эта особенность значительно повышает скорость чтения скрипта. Уменьшается

трафик, что экономит деньги и снижается нагрузка на сервер. Так же нет надобности ставить дополнительное программное обеспечение для обработки скриптов, т.к. их обрабатывают браузеры.

Недостатки клиентских языков программирования также вытекают из особенностей. Разные браузеры могут интерпретировать скрипты, исходя из своих индивидуальных предпочтений. Поэтому, результаты показа одних и тех же документов могут отличаться. Хотя, последнее время работа различных браузеров всё более унифицируется. Так же к недостаткам можно отнести и открытость кода таких языков для пользователей.

2 Описание проекта

Одной из важнейших задач оптимизации хода рекламных кампаний является прогнозирование динамики поведения пользователя в сети Интернет. Прогнозирование может быть проведено различными методами, так или иначе базирующихся на моделировании аспектов поведения пользователя и влияющих на него факторов. Это не только поведенческие характеристики пользователя, но и содержимое рекламной сети в виде совокупности связанных интернет страниц, а также набор рекламных кампаний и алгоритмы по доставке рекламных кампаний до пользователя. Моя дипломная работа будет представлять из себя веб-сайт в котором изменены алгоритмы и модели анализа поведения человека на веб сайтах, результатом которой будет более точная рекламная аналитика.

2.1 Язык гипертекстовой разметки, как основной язык Web-приложения

Язык разметки гипертекста (Hypertext Markup Language), или, как его больше именуют, HTML, - это компьютерный язык, находящийся в базе World Wide Web (Мировой Сети). Благодаря языку HTML каждый текст можно разметить, перестроив его в гипертекст с дальнейшей публикацией в Web.

Язык HTML обладает собственным набором символов, благодаря которым Web-браузеры воспроизводят страницу. Эти символы, именуемые дескрипторами, содержат в себе компоненты, требуемые для формирования гиперссылок.

Одной из характерных отличительных черт HTML-документов считается то, что сам документ содержит только текст, а все остальные объекты внедряются в документ в момент его отображения Браузером с поддержкой специализированных тэгов и хранятся в отдельности. При сохранении HTML-файла в месте размещения документа создается папка, в которую помещаются сопутствующие ему графические компоненты дизайна.

Новая версия языка HTML, с новейшими компонентами, свойствами и новым поведением. Комплект технологий, позволяющий создавать всевозможные сайты и Web-приложения.

HTML - это язык разметки гипертекста, разработанный для создания сайтов, которые в последствии имеют возможность собирать данные о всех, кто подключен к тому или иному сайту.

Он регулярно улучшается и модифицируется, для того чтобы соответствовать требованиям возрастающей онлайн-аудитории под управлением «W3C», организации, что разрабатывает и поддерживает язык.

HyperText - это метод, с помощью которого вы перемещаетесь по сети нажав на специальный текст, называемый гиперссылками, который переведет

к другой странице. Суть этого метода в том, что щелкнув по ссылкам, можно перейти на любой адрес в Интернете, когда захотите.

Разметка - это то, что HTML-теги делают с текстом внутри них. Они отмечают его как определенный тип текста (например, курсивом). HTML - это язык, поскольку он имеет кодовые слова и синтаксис, как любой другой язык. HTML состоит из серии коротких кодов-тегов, введенных автором текста в текстовый файл. Затем текст сохраняется как html-файл и просматривается через браузер, например Internet Explorer или любой другой браузер. Этот браузер читает файл и переводит содержимое в видимую форму. Написание собственного HTML предполагает правильное использование тегов для создания вашего видения. Вы можете использовать графический редактор для создания HTML-страницы

Теги - это отдельные обычные слова в HTML-коде. Они выглядят как текст между <угловыми скобками>. Они отвечают за мультимедийное содержимое, таблицы и прочее. Различные теги будут выполнять разные функции. Сами теги не отображаются, когда вы просматриваете свою страницу через браузер, но их видно в редакторе.

Диапазон навыков, которое человек узнает в результате запуска собственного веб-сайта, впечатляет. Он узнает о тонкостях графического дизайна, типографии и компьютерного программирования.

Язык разметки гипертекстовых страниц HTML представляет собой язык, разработанный специально для создания web-документов. С помощью него определяется размещение специальных тегов. Они указывают браузеру отображение на странице элементов сайта.

Редакторы, работающие с кодом позволяют проверить правильность документа, и другие задачи. Преимуществами использования такого редактора будет то, что весь код находится перед глазами, и тем самым обеспечивается полный контроль над содержимым web-страницы.

2.2 Язык PHP и его роль в разработке проекта

В первую очередь PHP применяется с целью формирования скриптов, функционирующих в стороне сервера, в целях этого его, собственно говоря, и разработали. PHP горазд регулировать эти же задачи, как и всевозможные прочие CGI-скрипты, в том числе подвергать обработке данные html-форм, динамически производить html страницы и тому аналогичное. Однако имеются и прочие сферы, в которых имеет возможность применяться PHP.

Вторая сфера - это разработка скриптов, выполняющихся в командной строке. То есть с помощью PHP возможно разрабатывать подобные скрипты, которые станут реализовываться, вне зависимости от web-сервера и браузера, на определенном устройстве.

И заключительная сфера - это формирование GUI-приложений (графических интерфейсов), выполняющихся на стороне клиента. Несмотря на то PHP и далеко не очень распространён в этой сфере, его имеется возможность применять и с целью формирования GUI-приложений.

Для формирования кроссплатформенных приложений предназначаются пакеты PHP-GTK и PHP-Qt, показывающие собою обёртки в целях определения распространенных библиотек виджетов. Кроме того имеется среда разработки кроссплатформенных приложений Devel Next.

Для формирования графических приложений на Windows имеются свободные пакеты WinBinder (прописан на C, практически — обёртка для WinAPI), PQBuilder(прописан в PHP с внедрением библиотеки PHPQt5), а кроме того предшественник Devel Next - среда быстрой разработки Devel Studio.

Кроме этого имеется реализация PHP на .NET/Mono - Phalanger и на JVM - JPHP, итогом компиляции PHP-кода в Phalanger может являться любое .NET-приложение, в тот же момент JPHP поддерживает расширение Swing, практически в полном объеме импортированное с среды Java.

В сфере веб-программирования, в частности серверной части, PHP - единственный из пользующихся популярностью сценарных языков (наравне с JSP, Perl и языками, используемыми в ASP.NET).

Популярность в сфере построения веб-сайтов обуславливается присутствием значительного комплекта интегрированных средств ради разработки веб-приложений. Ключевые из них:

- автоматическое получение POST и GET-характеристик, а кроме того переменных окружения веб-сервера в предопределённые массивы;
- взаимодействие с значительным числом всевозможных систем управления базами данных в нашем случае MySQL;
- автоматизированное отправление HTTP-заголовков;
- работа с HTTP-авторизацией;
- работа с cookies и сессиями;
- работа с локальными и удалёнными файлами, сокетами;
- обработка файлов, загружаемых на сервер;
- работа с XForms.

На сегодняшний день PHP применяется сотнями тысяч разработчиков. Согласно рейтингу компании TIOBE, базирующемуся на данных поисковых систем, в мае 2016 года PHP находился на 6 месте среди языков программирования. К крупнейшим сайтам, использующим PHP, причисляются Facebook, Wikipedia и др.

PHP-интерпретатор осуществляет исключительно PHP-код в собственных разделителях. Все, что располагается за пределами собственных разделителей, никак не обрабатывается PHP, хотя текст, который никак не принадлежащий к PHP, по-прежнему покоряется структурам управления,

описанным в PHP-коде. Преимущественно распространенными разделителями являются `<? Php` в целях раскрытия и `>` в целях закрытия разделов PHP. Укороченная форма `<? >` вдобавок имеется. Этот короткий разделитель делает файлы сценариев меньше портативными, таким образом их поддержка имеет возможность быть отключенной в локальной конфигурации PHP, и по этой причине она никак не рекомендуется. Однако отсутствуют советы относительно применения короткого тега `echo <? =`. Вплоть до PHP 5.4.0 данный короткий синтаксис для `echo ()` функционирует только лишь с включенным параметром конфигурации `short_open_tag`, в то время как для PHP 5.4.0 и более поздних версий он всегда доступен. Цель всех сих разделителей состоит в том, чтобы разъединить PHP-код от не-PHP-контента, например кода JavaScript или разметки HTML.

Переменные обладают префикс с символом доллара, и тип никак не нужно указывать заблаговременно. В PHP 5 установлен вид знака, какой даёт возможность функциям принудительно передавать собственные характеристики объектам предопределенного класса, массивов, интерфейсов или функций обратного вызова. Между тем до PHP 7.0 типы подсказок не имеют все шансы использоваться со скалярными типами, подобными как `integer` либо `string`.

В отличии от имен функций и классов имена переменных чувствительны к регистру. Обе строки с двойными кавычками (`"`) и `heredoc` предоставляют возможность интерполировать значение переменной в строку. PHP обрабатывает символы новой строки как пробельные символы в виде языка свободной формы, а операторы заканчиваются точкой с запятой. PHP имеет три типа синтаксиса комментариев: `/* */` / `mark block` и встроенные комментарии; `//` а также `#` используются для однострочных комментариев. Оператор `echo` является одним из нескольких объектов, которые PHP предоставляет для вывода текста, например, в веб-браузер.

С точки зрения синтаксиса ключевых слов и языка PHP похож на синтаксис стиля C. Условия, циклы `for` и `while` и возвращаемые функции похожи в синтаксисе на такие языки, как C, C ++, C #, Java и Perl. PHP хранит целые числа в зависящем от платформы диапазоне, либо 64-битное, либо 32-разрядное целое число, эквивалентное длинному типу Сязыка. Целочисленные целые числа преобразуются в знаковые значения в определенных ситуациях. Это поведение отличается от поведения других языков программирования. Целочисленные переменные могут быть назначены с использованием десятичных (положительных и отрицательных), восьмеричных, шестнадцатеричных и двоичных обозначений. Нулевой тип данных представляет собой переменную, которая не имеет значения; `NULL` является единственным допустимым значением для этого типа данных.

Переменные типа «ресурс» представляют ссылки на ресурсы из внешних источников. Они обычно создаются функциями определенного расширения и могут обрабатываться только функциями из одного и того же расширения. Примеры включают файлы, изображения и базы данных.

Массивы могут содержать элементы любого типа, с которыми может работать РНР, включая ресурсы, объекты и другие массивы. Заказ сохраняется в списках значений и в хэшах с двумя ключами и значениями, и оба могут быть смешаны. РНР также поддерживает строки, которые могут использоваться с одинарными кавычками, двойными кавычками, синтаксисом `powdoc` или `heredoc`.

Стандартная библиотека РНР (SPL) пытается решить стандартные проблемы и реализует эффективные интерфейсы и классы доступа к данным.

РНР определяет большой набор функций на основном языке, и многие из них также доступны в различных расширениях. Эти функции хорошо документированы в онлайн-документации РНР. Однако встроенная библиотека имеет множество соглашений об именах, как описано в предыдущей истории.

Пользовательские функции могут быть определены разработчиком.

Вместо указателей на функции в РНР могут ссылаться на строку, содержащую их имя. Таким образом, обычные функции РНР могут использоваться, например, как обратные вызовы или внутри таблиц функций. Пользовательские функции могут быть созданы в любое время. Функции могут быть определены внутри кодовых блоков, что позволяет принимать вовремя решение о том, следует ли определять функцию. Существует функция `function_exists`, которая определяет, была ли определена функция с заданным именем. Вызовы функций должны использовать круглые скобки, за исключением функций конструктора классов с нулевым аргументом, вызванных с новым оператором РНР, и в этом случае скобки являются необязательными.

Основные функциональные возможности объектно-ориентированного программирования были добавлены в РНР 3 и улучшены в РНР 4. Это позволило РНР получить дополнительную абстракцию, упрощая креативные задачи для программистов, использующих язык. Обработка объектов полностью переписана для РНР 5, расширяя набор функций и повышая производительность. В предыдущих версиях РНР объекты обрабатывались как типы значений. Недостатком этого метода было то, что код должен был использовать «ссылочные» переменные РНР, если он хотел изменить объект, который был передан, а не создавать его копию. В новом подходе объекты ссылаются на дескриптор, а не по значению.

В РНР 5 введены частные и защищенные переменные-члены и методы, а также абстрактные классы, конечные классы, абстрактные методы и конечные методы. Он также представил стандартный способ объявления конструкторов

и деструкторов, аналогичных тем, что и для других объектно-ориентированных языков, таких как C ++, и стандартной модели обработки исключений. Кроме того, PHP 5 добавил интерфейсы и позволил реализовать несколько интерфейсов. Существуют специальные интерфейсы, которые позволяют объектам взаимодействовать с системой времени выполнения. Объекты, реализующие ArrayAccess, могут использоваться с синтаксисом массива, а объекты, реализующие Iterator или IteratorAggregate, могут пользоваться конструкцией языка foreach. Отсутствуют функции виртуальной таблицы, поэтому статические переменные связаны с именем вместо ссылки во время компиляции.

В PHP 7.1 были добавлены новые функции, классы, интерфейсы, библиотеки. Если сравнивать с более старыми версиями данная версия намного производительней. Ведь как говорят в простом народе PHP 7 и более заточены на производительность. В нашей разработке будем использовать именно эту версию.

2.3 JavaScript

Язык программирования JavaScript разработан фирмой Netscape для создания интерактивных HTML-документов. Это объектно-ориентированный язык разработки встраиваемых приложений, выполняющихся как на стороне клиента, так и на стороне сервера. Синтаксис языка очень похож на синтаксис Java - поэтому его называют - Java-подобным.

Основные области применения JavaScript делятся на следующие категории:

- динамическое создание документа с помощью сценария;
- оперативная проверка достоверности заполняемых пользователем полей форм HTML до передачи их на сервер;
- создание динамических HTML-страниц совместно с каскадными таблицами стилей и объектной моделью документа;
- взаимодействие с пользователем при решении "локальных" задач, решаемых приложением JavaScript, встроенном в HTML-страницу.

2.4 Notepad++

Notepad++ — свободный текстовый редактор с открытым исходным кодом для Windows с подсветкой синтаксиса большого количества языков программирования и разметки. Поддерживает открытие более 100 форматов. Базируется на компоненте Scintilla, написан на C++ с использованием STL, а также Windows API и распространяется под лицензией GNU General Public License. Базовая функциональность программы

может быть расширена как за счёт плагинов, так и сторонних модулей, таких как компиляторы и препроцессоры.

Notepad ++ распространяется как бесплатное программное обеспечение. Проект дважды выиграл награду SourceForge Community Choice Award за лучший инструмент для разработчиков.

Блокнот ++ был разработан Дон Хо в сентябре 2003 года. Разработчик ранее использовал JEXT (текстовый редактор на основе Java), но был недоволен низкой производительностью, и начал разрабатывать текстовый редактор, написанный на C ++ с помощью Scintilla. Он разработал его, но проект был отвергнут. Notepad ++ был создан как приложение Microsoft Windows. Notepad ++ был впервые выпущен на SourceForge 25 ноября 2003 года в качестве приложения для Windows. Он основан и помощью редактора Scintilla и написан на C ++ только с вызовами Win32 API с использованием STL для повышения производительности и уменьшения размера программы.

В январе 2010 года правительство США обязало основателей проекта с открытым исходным кодом США отказать в доступе в Кубе, Иране, Северной Корее, Судане и Сирии в соответствии с законодательством США. В ответ на это в июне 2010 года Notepad ++ вышла из территориальной юрисдикции США, выпустив версию на TuxFamily во Франции.

В 2011 году Liferhacker описал Notepad ++ как «Лучший текстовый редактор для Windows», заявив, что «если вы предпочитаете простой, легкий и расширяемый программный текстовый редактор, наш первый выбор - это бесплатный Notepad ++ с открытым исходным кодом». Liferhacker раскритиковал пользовательский интерфейс, заявив, что «на самом деле это довольно не красиво. К счастью, вы можете многое сделать, чтобы настроить его внешний вид, но это компенсирует функциональность».

В 2015 году Stack Overflow провела всемирный опрос разработчиков, и Notepad ++ был признан наиболее используемым текстовым редактором во всем мире, и 34,7% из 26 086 респондентов заявили, что ежедневно его используют.

Notepad ++ поддерживает несколько языков программирования и естественные языки.

Особенности Notepad ++:

- поддерживает символы окончания строки в Microsoft Windows, Unix и Classic Mac OS;
- автоматическое резервное копирование несохраненных файлов;
- форматы текстового кодирования, включая ASCII, Unicode, UTF-8 и несколько кодировок UTF-16;
- поиск и замена регулярных выражений;
- маркировка / сводка событий;
- сравнение данных;
- запись и выполнение макросов;

- закладки;
- одновременное редактирование;
- редактирование разделенного экрана и синхронизированная прокрутка;
- интерфейс с вкладками;
- масштабирование;
- синтез речи (текст в речь);
- несколько буферов обмена;
- проверка орфографии;
- FTP-браузер;
- программирование;

Notepad ++ является одним из самых популярных редакторов исходного кода в мире и поддерживает подсветку синтаксиса и сворачивание кода для более чем 50 языков программирования, сценариев и разметки. Его функции, связанные с программированием, включают:

- автозаполнение для подмножества API некоторого языка программирования;
- выделение синтаксиса;
- рекомендации по отступу для файлов с вкладками;
- подгонка подсветки;
- соответствие выделения тегов (для тегов HTML и XML);
- различные инструменты, такие как сортировка строк, преобразование текста;
- автообнаружение состояния файла;
- настраиваемое сочетание клавиш;
- список функций;
- автоматическое определение языка программирования, который использует данный файл, с помощью изменяемого списка;
- ручная установка языка программирования текущего файла;
- пользовательские языки программирования (для подсветки синтаксиса и автозаполнения);
- настраиваемая подсветка синтаксиса (стили шрифтов и цвета).

Notepad ++ поддерживает следующие языки программирования:

- Ada, asp, Assembly, AutoIt;
- C, C ++, C #, Caml, Cmake, COBOL, CoffeeScript, CSS;
- D, Diff;
- Flash ActionScript, Fortran;
- Gui4CLI;
- Haskell, HTML;
- INNO;
- Java, Javascript, JSP;
- KiXtart;
- LISP, Lua;

- Makefile, Matlab, MS-DOS, INI;
- NSIS, обычный текстовый файл;
- Objective-C;
- Pascal, Perl, PHP.

Notepad ++ поддерживает макросы и плагины, также был отмечен своей надежной архитектурой плагина, которая позволила включить в программу различные новые функции. В настоящее время для Notepad ++ разработаны более 140 совместимых плагинов, из которых 10 по умолчанию включены в программу. Первым плагином, который должен быть включен в программу, был «TextFX», который включает проверку W3C для HTML и CSS, сортировку текста, изменение символов и обработку цитат.

3 Проектирование и разработка программного продукта

Основной идеей предлагаемого подхода является разработка модели поведения пользователя наиболее близкой к реальному поведению, и дальнейшее использование её для прогнозирования поведенческих характеристик

Frontend это та часть, которая отображается пользователю, выполняется в веб-браузере и взаимодействует визуально с пользователем. На этой стороне работают такие языки разметки, стилей и программирования как HTML, CSS и JavaScript.

Backend не имеет собственного визуального представления и взаимодействует с пользователем через веб-браузер. На этой части мы будем использовать такие программные языки как Java, PHP, C# и т. д. Данный веб сайт разрабатывается на языке программирования PHP.

3.1 Проектирование и разработка формул алгоритмов

В модели имеется ряд допущений. В первую очередь предполагается, что пользователь посещает случайные страницы. Поэтому пользователь характеризуется законом распределения посещаемых страниц. Вторым допущением будет являться то, что на определённом наборе страниц существует несколько групп пользователей, имеющих свои определённые характеристики. В предлагаемой модели предлагается рассматривать

- $B(u)$ - начало первой сессии в течение дня;
- $E(u)$ - окончание последней сессии;
- $C(u)$ - количество сессий в течение дня;
- $V(u)$ - продолжительность сессии;
- $A(u)$ - количество просматриваемой рекламы за час.

Здесь u - это пользователь.

Такая модель позволяет прогнозировать частотные характеристики непосредственно с прогнозированием временных характеристик - плотности прихода пользователей на сайт.

Процесс показа рекламы группе пользователей является случайным процессом. Пусть $F(x)$ - плотность распределения пользователей по числу показов рекламы за день на пользователя, получаемая из вышеописанных параметров модели поведения пользователя и статистических данных отчётов, предоставляемых AdServer'ом. Расчёт производится градиентным методом с использованием эвристических методов поиска начального приближения.

Лимит числа просмотров кампании i для одного пользователя для кампаний равны L_i . Будем полагать, что список кампаний упорядочен по возрастанию ограничений.

В силу того, что количество показов рекламы достаточно велико и может достигать десятков миллионов показов в день, можно перейти к рассмотрению типичных последовательностей. Перейдем также к рассмотрению мат. ожиданий. Это допустимо, поскольку для задачи прогнозирования представляет интерес в первую очередь математическое ожидание числа показов кампании, а не точное распределение, получить которое невозможно вследствие отсутствия необходимой информации. При таком рассмотрении число выпавших на кампанию показов будем считать равным его математическому ожиданию. Другими словами, если g показов разыгрывается между кампаниями, то количество показов равно:

$$I_i(r) = r \frac{w_i}{\sum_k w_k}, \quad (1)$$

где i - номер кампании, w_i - вес кампании.

Формула (1) применима для большинства случаев, и используется во всех системах динамики прогнозирования поведения пользователя. Однако, наличие хотя бы одной кампании с частотным ограничением меняет реакцию группы пользователей.

Рассмотрим поведение отдельного пользователя. Пользователь просматривает страницы содержащие рекламу. При этом каждая кампания, показываемая к моменту просмотра пользователем g реклам, характеризуется набранным количеством показов $I(r)$, а так же возможностью показываться данному пользователю, определяемой неравенством

$$I_i(r) \leq L_i, \quad (2)$$

Тогда количество показов кампании к моменту показа дополнительных r_i :

$$I_i(r + \Delta r) = I_i(r) + \Delta r \cdot \frac{w_i}{\sum_{k: R_k \leq L_k} w_k}, \quad (3)$$

Заметим, что в любой момент времени по отношению к отдельному пользователю существует два вида кампаний: достигшие предела L_i и продолжающие делить оставшиеся показы пользователя пропорционально весам.

При переходе на математическое ожидание из (1) получаем:

$$MI_i = \int_0^{+\infty} I_i(r) f(r) dr, \quad (4)$$

После разделения на отрезки, на которых набор кампаний, достигших предела не изменяется, из (4) получим следующее:

$$MI_i = \sum_{k=0}^N \int_{r_k}^{r_{k+1}} I_i(r) f(r) dr, \quad (5)$$

где N - количество отрезков, r_k - количество показов в начале отрезка k . Отметим, что в общем случае количество отрезков равно количеству кампаний. Однако в случае, когда более одной кампании в один момент выбирают свой лимит, отрезок вырождается в точку.

С учётом (3) формулу (5) можно переписать в виде:

$$MI_i = \sum_{k=0}^i MI_i^k + L_i \int_{r_i}^{\infty} f(r) dr, \quad (6)$$

где r_i - количество показов рекламы пользователю, при котором останавливается кампания i .

При этом:

$$MI_i^k = \frac{w_i}{\sum_{j=k}^N w_j} \int_{r_k}^{r_{k+1}} (r - \sum_{j=1}^{k-1} L_j) f(r) dr, \quad (7)$$

Для нахождения r_k предлагается следующий итеративный алгоритм:

1) Полагаем, что на предыдущих шагах вычислены точки r_k для $k < k_0$, где k_0 - номер текущего шага.

2) r_k - минимальное количество показов, когда сменяется набор кампаний. Другими словами - минимальное количество показов, когда одна из кампаний выберет свой лимит L_k . Таким образом

$$(r - \sum_{j=1}^{k-1} L_j) \frac{w_i}{\sum_{j=k}^N w_j} = L_i, i \geq k_0, \quad (8)$$

3) Так как кампании должны быть отсортированы по возрастанию L_i , кампанию, для которой был получен этот минимум, меняем местами с кампанией k .

Полученные в итоге величины MI_i представляют собой среднее число показов на пользователя для кампании i .

3.2 Этапы разработки веб-приложения

А начну я свою работу с концептуального проектирования, и точнее с построения ER диаграммы. Концептуальное проектирование – это построение семантической модели предметной области и ее описание. Процесс такого

проектирования создаётся без направленности на какую-либо конкретную систему управления базами данных.

ER-модель или ER-диаграмма – это модель данных, которая дает возможность описать концептуальные схемы предметной области. Моделью данных является графическое описание предметной области при помощи стандартизированного набора обозначений. Для применения в выбранной СУБД строится логическая модель основе ER-модели.

Логическое модель отражает процесс проектирования информационной модели, которая выполняется на основе созданной модели данных, но без учета определенной СУБД и других физических ограничений. [9]

Для перевода логической модели данных в физическую необходимо руководствоваться следующими правилами:

- Перевод объектов в таблицы может быть выполнен только в физической базе данных;

- Для того чтобы атрибуты превратились в поля необходима физическая база данных, также необходимо выбрать тип данных для атрибута;

- любой идентификатор, тип которого уникален, становится первичным ключом, то есть полем, не допускающим нулевое значение.

Отношение в базе данных или другими словами связь – это графически изображаемая ассоциация, устанавливаемая между двумя разными сущностями. Эта ассоциация всегда является бинарной и может существовать между двумя разными сущностями или между сущностью и ей же самой.

Основные элементы, входящие в состав ER-моделей:

- объекты предметной области (сущности);

- непосредственно связи между объектами;

- свойства объектов (атрибуты).

В каждой ассоциации можно выделить два конца (согласно паре сущностей, связанных между собой), на каждом из которых устанавливается степень конца связи (количество связываемых экземпляров сущности).

Моделирование связей между объектами определяется типом связи и классом принадлежности. Основные типы связей:

- «один к одному» (1:1);

- «один ко многим» (1:∞);

- «многие ко многим» (∞:∞).

Класс принадлежности является обязательным, если все объекты одного класса участвуют в связи с объектами другого класса или необязательным, если объекты одного класса не участвуют в связи с объектами другого класса. ER- диаграмма для разрабатываемого сайта представлена на рисунке 3.3.1:

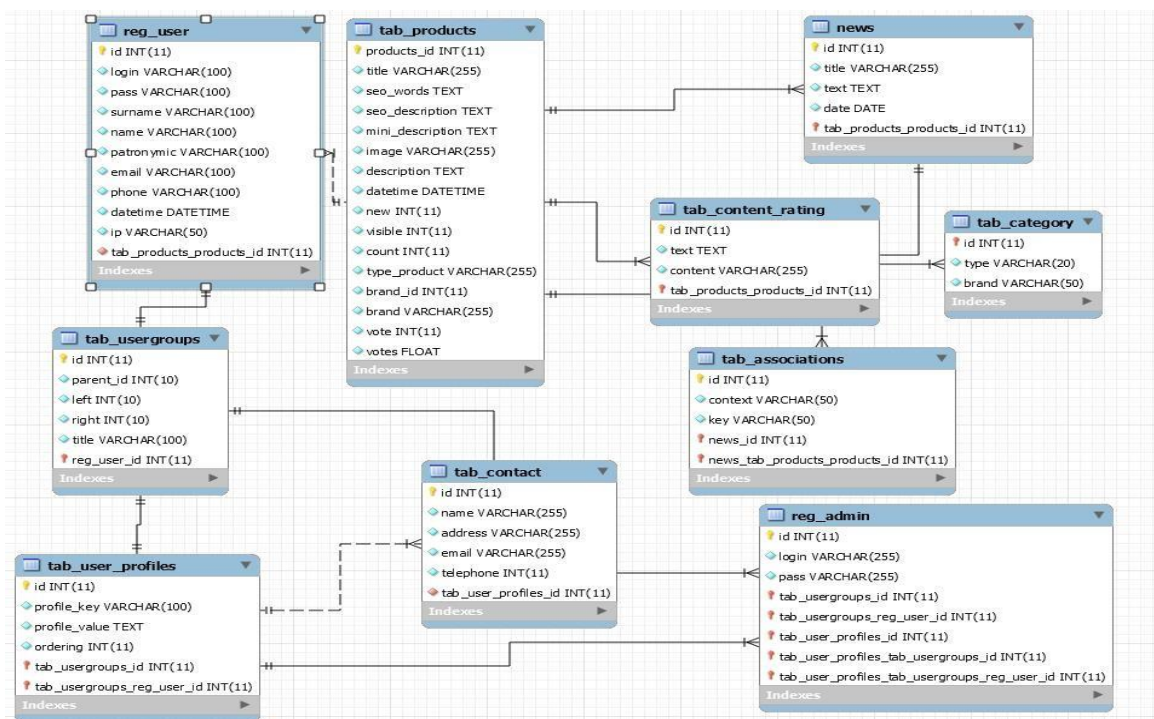


Рисунок 3.3.1 – ER-диаграмма

База данных для Web-сайта «Kupiltraktor» состоит из следующих таблиц:

Таблица 3.1 – Контент

Имя поля	Назначение поля	Тип данных
products_id	Первичный ключ	Счетчик
title	Название	Текстовый
mini_description	Мини описание	Текстовый
image	Ссылка на изображение	Текстовый
description	Полное описание	Текстовый
datetime	Дата добавления	Дата/время
new	Новинки	Числовой
visible	Отображение на сайте	Числовой
count	Количество просмотров	Числовой
type_product	Тип	Текстовый
brand_id	Номер категории	Числовой
brand	Категория	Текстовый
vote	Отзыв	Числовой

Таблица 3.2 – Категории

Имя поля	Назначение поля	Тип данных
id	Первичный ключ	Счетчик
type	Тип	Текстовый
brand	Название категории	Текстовый

Таблица 3.3 – Пользователи

Имя поля	Назначение поля	Тип данных
id	Первичный ключ	Счетчик
login	Логин	Текстовый
pass	Пароль	Текстовый
surname	Фамилия	Текстовый
name	Имя	Текстовый
patronymic	Отчество	Текстовый
email	Электронная почта	Текстовый
phone	Телефон	Числовой
datetime	Дата	Дата/время

Таблица 3.4 - Администраторы

Имя поля	Назначение поля	Тип данных
id	Первичный ключ	Счетчик
login	Логин	Текстовый
pass	Пароль	Текстовый

Таблица 3.5 – Группы пользователей

Имя поля	Назначение поля	Тип данных
id	Первичный ключ	Счетчик
parent_id	Номер родительской	Числовой

	страницы	
title	Название	Текстовый

Таблица 3.6 – Контакты

Имя поля	Назначение поля	Тип данных
id	Первичный ключ	Счетчик
name	Имя	Текстовый
address	Адрес	Текстовый
email	Электронная почта	Текстовый
telephone	Телефон	Числовой

Таблица 3.7 – Новости

Имя поля	Назначение поля	Тип данных
id	Первичный ключ	Счетчик
title	Название	Текстовый
text	Описание	Текстовый
date	Дата добавления	Дата/время

Таблица 3.8 – Профили

Имя поля	Назначение поля	Тип данных
id	Первичный ключ	Счетчик
profile_key	Ключ профиля пользователя	Текстовый
profile_value	Значение профиля пользователя	Текстовый

Таблица 3.9 – Связи

Имя поля	Назначение поля	Тип данных
id	Первичный ключ	Счетчик
context	Данные контактов	Текстовый
key	Ключ контактов	Числовой

Важной частью нашего проекта является интегрирование формул для анализа. В данном разделе описывается работа веб-сайта. Блок-схема работы рекламного взаимодействия представлена на рисунке 3.3.1.



Рисунок 3.3.1 – Зависимость составляющих интернет рекламы

Как видно на рисунке 3.3.1 пользователь заходит на страницу интернет страницу, после этого пользователю присваивается временной штамп(отметка) которая фиксирует время его сеанса на сайте. Так же пользователю присваивается ID номер, который позволяет нам фиксировать каждого посетителя сайта и выдавать более обширный обхват аналитике.

```

<script>
var wixBiSession = {
  initialTimestamp: Date.now()
  , sssRequestTimestamp: 1588069048572
  , requestId: publicModel.requestId
  , viewerSessionId: 'XXXXXXXX-XXXX-4XXX-XXXX-XXXXXXXXXXXX'.replace(/[/\x]/g,function(c){var r=Math.random()*16|0,v=c=='x'?r:(r&0x3|0x8);return v.toString(16);})
  , sessionId: '8eca9595-65a5-4fda-ae63-374a174bac74'
  , initialRequestTimestamp: performance.timeOrigin ? performance.timeOrigin : Date.now() - performance.now()

  , is_rollout: 0
  , is_platform_loaded: 1
  , suppressibi: false
  , dc: '84'
  , renderType: 'bolt'
  , siteRevision: '867'
  , siteCacheRevision: '1588893427971'
  , wixBoltExclusionReason: ''
  , wixBoltExclusionReasonMoreInfo: ''
  , checkVisibility: (function () {
    var alwaysVisible = document.hidden !== true;
    function checkVisibility() {
      alwaysVisible = alwaysVisible && document.hidden !== true;
      return alwaysVisible;
    }
    document.addEventListener('visibilitychange', checkVisibility, false);
    return checkVisibility;
  })()
  , cacheCookie: document.cookie.match(/ssr-caching="?cache[,\#\s]*desc=(\w+)(?:[,\#\s]*varnish=(\w+))?(?:[,\#\s]*dc[,\#\s]*desc=(\w+))?(?:!|:|;)/)
  , setCaching: function (bodyCacheable) {
    if (!bodyCacheable) {
      wixBiSession.caching = undefined;
    }
  }

  var parts = wixBiSession.cacheCookie;
  if (parts) {
    set(parts);
  }

  if (window.PerformanceServerTiming) {
    var serverTiming = performance.getEntriesByType('navigation')[0].serverTiming;
    if (serverTiming && serverTiming.length) {
      var names = ['cache', 'varnish', 'dc'];
      parts = [];
      serverTiming.forEach(function (entry) {
        var i = names.indexOf(entry.name);
        if (i > 0) {
          parts[i] = entry.description;
        }
      });
    }
    set(parts);
  }
}

```

Рисунок 3.3.2 – Фиксация времени сессии и номера пользователя

Теперь перейдем к основной части это создание веб-сайта.

3.3 Разработка дизайна Web-сайта

Согласно разработанной структуре была спроектирована главная страница Web-сайта. В ее состав входят все основные структурные элементы, переход по которым осуществляется с помощью гиперссылок.

В разработку сайта входит процесс создания макета Web-страниц, на который впоследствии будут нанизываться все остальные элементы. При этом происходит формирование так называемых структурных блоков сайта - обособленных модулей, каждый из которых играет определенную роль и отвечает за определенный функционал ресурса.

Состав Web-сайта включает в себя:

1) Шапка сайта.

Самый верхний блок часто называют шапкой сайта, либо заголовок сайта, либо хедер от англ. header. Место, в котором обычно располагается хедер - верхняя часть страницы. Ориентация хедера – альбомная (рисунок 3.4).



Рисунок 3.4 - Шапка разрабатываемого сайта «kupiltraktor»

2) Основная область сайта (область основного контента).

Слово «Контент» происходит от английского «content» - содержание. Это самая большая по площади и наиболее важная для посетителей сайта часть страницы.

В данном блоке размещается:

- текстовый контент;
- графический контент.

Ширина области основного контента может варьироваться в довольно широких пределах - тут все зависит от типа макета сайта (жесткий или резиновый). Резиновый макет позволяет менять ширину блока в зависимости от разрешения монитора, с которого пользователь просматривает сайт. Жесткий макет этого сделать не позволяет (рисунок 3.5):



Рисунок 3.5 - Контент разрабатываемого сайта «kupiltraktor»

3) Сайдбар.

Понятие «Сайдбар» происходит от английского термина "sidebar", где "side" - сторона, "bar" - полоса. Обычно сайдбаром называют боковую колонку сайта, которая размещается справа или слева от области основного контента. В некоторых случаях на сайте могут располагаться сразу два сайдбара (первый - слева от основной области, а второй - справа).

Как правило, содержимое сайдбара не меняется от одной страницы сайта к другой, в отличие от содержимого области основного контента. Поэтому в сайдбаре обычно размещают блоки со ссылками, а также важную служебную информацию.

В сайдбар разрабатываемого сайта входит:

- Главная;
- КХ Джантагулова;
- Продукция
- Видео галерея
- О компании
- Фотогалерея
- Новости
- Вопросы и ответы
- Контакты
- Форум

Ширина сайдбара, как правило, четко фиксирована и не зависит от типа сайта (рисунок 3.6):

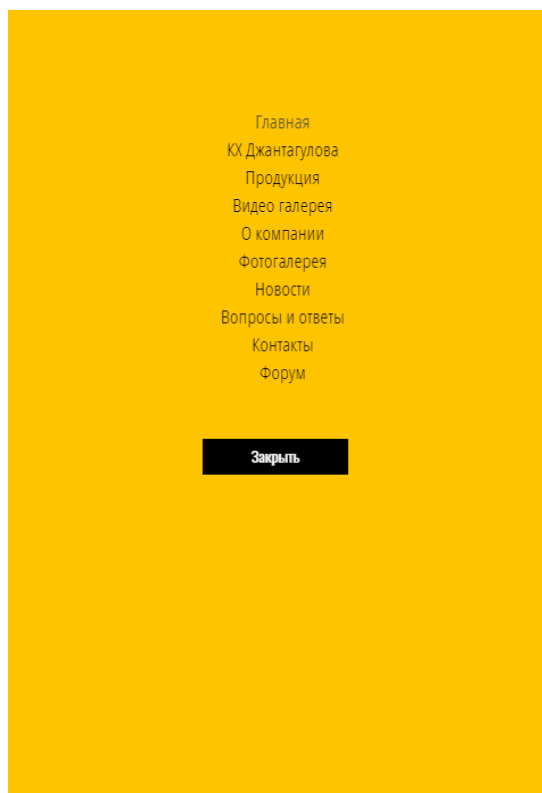


Рисунок 3.6 – сайдбар (меню сайта)

4) Футер сайта (или подвал).

Слово «Футер» произошло от англ. «Footer» - подвал. Обычно футером называют область сайта, которая расположена под всеми остальными блоками. По аналогии с шапкой сайта, футер также имеет альбомную ориентацию.

Ширина футера также может меняться в зависимости от разрешения монитора пользователя (рисунок 3.7).



Рисунок 3.7. – футер сайта

5) Фон сайта (фоновая область).

Фоновая область сайта не занята никакими элементами (является полностью свободной). Размер фоновой области зависит от типа макета сайта. При использовании резинового макета фона может и не быть, если макет жесткий, то размеры фоновой области будут напрямую зависеть от разрешения монитора, с которого пользователь просматривает сайт.

Чем больше будет разрешение, тем больше места будет занимать фон для сайта. Фоновая область может заполняться определенным цветом (посредством CSS-атрибута `background-color`), а также одной большой или несколькими маленькими симметричными изображениями (которые будут равномерно заполнять собой все свободное пространство).

3.5 Проектирование и разработка кода сайта

Все документы веб-сайта, которые мы видим в браузере, написаны на языке HTML. HTML (Hyper Text Markup Language) это язык, являющийся стандартом для разработки документов, которые будут отражаться в интернет-браузерах. Ключевым смыслом языка HTML в разметке текста, для чего используются специальные команды, называемые теги, графически они отображаются в угловых скобках. HTML – документы хороши тем, что для их просмотра подходит практически любой браузер. Документ, написанный на языке HTML, будет правильно отображаться на различных устройствах, имеющих различное расширение экрана и операционную систему. Практически все документы содержат в себе стандартные элементы, такие, как титульная надпись, заголовки или списки. С помощью тегов HTML можно обозначать данные элементы, обеспечивая веб-браузеры минимальной информацией для отображения данных элементов, сохраняя в целом общую структуру и информационную полноту документов. Браузер – это окно, через которое мы просматриваем интернет веб-сайты. То есть все странички должны быть написаны на языке HTML.

Создать документы, написанные на языке HTML, которые станут основой вашего веб-сайта, можно несколькими способами.

Основным способом является текстовый набор HTML-кода в редакторе. Это достаточно трудоемкий процесс, который подразумевает глубокое знание принципов построения документов в HTML, умение работать с каскадными таблицами стилей CSS, а также как минимум с JavaScript.

Вторым способом является использование CMS. Это движок сайта, который позволяет пользователю почти автоматически создавать веб-сайты, подразумевает меньший набор знаний для создания и требует гораздо меньше времени на разработку. Также при использовании CMS гораздо проще осуществлять дальнейшее обслуживание сайта, добавление или удаление новой информации.

Также есть возможность создавать веб-сайты с помощью специальных редакторов, которые объединяются под общим названием WYSIWYG-«What You See Is What You Get», который можно перевести как «что ты видишь, то ты и получаешь». Создание веб-сайтов в этих программах напоминает работу в графических и текстовых редакторах, но на выходе мы получим не

изображение, а документ на языке HTML. Этот Метод позволяет достаточно быстро получить документ с достаточно сложным оформлением.

Для написания внешнего вида элементов применяется каскадные таблицы стилей CSS.

CSS довольно элементарный формальный язык, который придумали для описания внешнего вида документов. Это говорит о том, что он довольно прост и состоит из самобытных примитивных конструкций, которые не так сложны для изучения. Самое сложное не синтаксис, не правила написания конструкций, а огромное количество CSS свойств, для запоминания, которые выполняют различные задачи. Благо, все правила англоязычные с соответствующей смысловой нагрузкой. Простой перевод на наш язык дает понятие о том, что это правило делает и наоборот – при переводе того, что мы хотим добиться определенным свойством на английский язык, велика вероятность того, что мы получим правильное свойство. Это значительно упрощает запоминание CSS правил на интуитивном уровне.

CSS имеет довольно простой синтаксис. Правила объявлений настолько просты, что их можно описать в одном предложении. Сначала пишется селектор, который выбирает конкретный элемент на странице после фигурных скобок пишутся свойства со значениями после двоеточия, а сами свойства отделяются друг от друга точкой с запятой. Самое сложное в CSS объявлении - селектор.

CSS селектор - (от слова select – выбирать) – это конструкция, с которой начинается каждый блок объявлений, и которая служит для выборки элемента или однотипных элементов на странице для дальнейшей стилизации. Чаще всего в качестве селектора используется определенный класс тега.

Первое, на что обращает внимание человек, когда заходит на какой-либо веб-сайт, это его дизайн. Именно от того, насколько удачно сделан дизайн веб-сайта зависит захочет ли пользователь просматривать страницу дальше или закроет её. Поэтому очень важно при проектировании веб-сайта хорошо понимать, на какую аудиторию он рассчитан, какие потребности возникают у пользователей. Всеми этими вопросами занимается UX/UI.

UX или User Experience (дословно: «опыт пользователя») - дисциплина изучающая опыт взаимодействия пользователя с продуктом, восприятие и реакцию, возникающую в результате использования.

UI – это User Interface (дословно: «пользовательский интерфейс») - то, как выглядит интерфейс и то, какие физические характеристики приобретает. Определяет какого цвета будет ваше «изделие», удобно ли будет человеку попадать пальцем в кнопки, читабельным ли будет текст и так далее.

На нашем сайте интерфейс должен быть простым и понятным, так как данным сайтом будут пользоваться все население в возрасте от 10 до 70 лет.

Сайт будет состоять из трех страниц: информативная часть, страница конвертера текстов и страница конвертера файлов.

Основные теги HTML это head, body, footer. Head это тег в котором происходит установка кодировки страницы, а также ссылки на стили применяемые для данной страницы(рисунок 3.8).

```
<script type="text/javascript">

    var serviceTopology = {"cacheKillerVersion": "1", "staticServerUrl": "https://static.parastorage.com/", "usersScriptsRoot": "//static.parastorage.com/services"};
    var santaModels = true;
    var isStreaming = true;
    var rendererModel = {"metaSiteId": "618e077d-c7dd-4c32-b032-963f172f5981", "siteInfo": {"documentType": "UGC", "applicationType": "HtmlWeb", "siteId": "7e7979d8-"}, "publicModel": {"domain": "kupiltraktor.kz", "externalBaseUrl": "https://www.kupiltraktor.kz/", "unicodeExternalBaseUrl": "https://www.kupiltraktor.kz"};

    var isSAVRollout = "false";
    var googleAnalytics = "";
    var ipAnonymization = false;

    var googleRemarketing = "";
    var googleTagManager = "";
    var facebookRemarketing = "";
    var yandexMetrika = "";

</script>
```

Рисунок 3.8 – Тег head

```
</defs><image width="359" height="310" x="0" y="0" transform="" preserveAspectRatio="xMidMid slice" id="comp-jwsz18zimimage" data-type="image" xlink:href="https://static.wix.com/mask-comp-jwsz18yimimg" data-bbox="150 325 880 350"/>
<mask id="mask-comp-jwsz18yimimg">
  <use id="mask-comp-jwsz18yimimg-svg-use" xlink:href="#mask-comp-jwsz18yimimg-svg" width="100%" height="100%" x="0" y="0"/></use>
</mask>
<svg preserveAspectRatio="none" id="mask-comp-jwsz18yimimg-svg" data-bbox="150 350 880 380"/>
  <g>
    <path d="M180 20v160H20v20h160z"/></path>
  </g>
</svg>
</defs><image width="428" height="348" x="0" y="0" transform="" preserveAspectRatio="xMidMid slice" id="comp-jwsz18yimimg" data-type="image" xlink:href="https://static.wix.com/mask-comp-jwsz18yimimg" data-bbox="150 380 880 400"/>
<li><p class="font_8"><span style="font-family:play,sans-serif;">Трактора класса<span> 0.6-0.9<span> активно используется.</span></p></li>
<li><p class="font_8"><span style="font-family:play,sans-serif;">в растениеводстве<span> на<span> площадях до 50 га.</span></p></li>
<li><p class="font_8"><span style="font-family:play,sans-serif;">в животноводстве</p></li>
<li><p class="font_8"><span style="font-family:play,sans-serif;">на коммунальных<span> предприятиях</p></li>
</ul></div><div data-packed="true" style="width: 276px; pointer-events: none; opacity: 0;" class="txtNew" id="comp-jwituy83"><p class="font_7" style="font-size:22px; text-align: center;"><span style="font-family:play,sans-serif;">Трактора класса 1.4<span> активно используется.</span></p></li>
<li><p class="font_8"><span style="font-family:play,sans-serif;">в растениеводстве<span> на<span> площадях до 200<span> га.</span></p></li>
<li><p class="font_8"><span style="font-family:play,sans-serif;">в животноводстве</p></li>
<li><p class="font_8"><span style="font-family:play,sans-serif;">на коммунальных<span> предприятиях</p></li>
</ul></div><div data-packed="true" style="width: 276px; pointer-events: none; opacity: 0;" class="txtNew" id="comp-jwituy9k"><p class="font_7" style="font-size:22px; text-align: center;"></p></div>
```

Рисунок 3.9 – Тег header

На рисунке 3.5.2 показан тег body. Body - тег с которого начинается разметка, которую мы видим на веб – страницах. В свою очередь тег body делится на секции, так как данный тег может быть очень большим. Деление на секции также удобно для прописывания стилей, которые будут применены к данным секциям. Результат кода который, представлен на рисунке 3.5.2 виден на рисунке 3.10.



Рисунок 3.10 – главная страница

Далее мы вставим аналитические формулы для вывода анализа. На рисунке 3.11 будет представлена часть кода работы аналитического строения.

```

<script type="text/javascript" src="/e9947f1ab0"/></script><script src="/nr-1167.min.js.see.nazammi"/></script><script type="text/javascript">window.NREUM||(NREUM=[]);NREUM.info = {agent:"","beacon":{}};NREUM.info = {agent:"","beacon":{"data-url":"data","type":
<link rel="stylesheet" href="/app.min.css">
<link rel="stylesheet" href="/osa">
<style rel="stylesheet">
  body { font-family: 'Roboto' }
</style>
<link id="osatom-css" rel="stylesheet" href="/7885084-5779-402b-8210-0b0027abb46f.osa">
</script>
<script type="text/javascript">
(function () {
  var appName = 'answers-widget';
  window.fedops = window.fedops || {};
  window.fedops.apps = window.fedops.apps || [];
  window.fedops.apps[appName] = { staticUrl: performance.now() };
  window.fedops.sessionId = window.localStorage.getItem('fedops_logger_sessionId') || 'xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx'.replace(/(x)/g, function(c) { var z = Math.random()*16|0, v = c == 'x' ? z : (z&0x3|0&8); return
  (new Image()).src = '//log.wix.com/fed?appName=' + appName + '&src=72e618b-850-402b-a27e-1e8c8e86c97c','parentId':'00000000-0000-0000-0000-000000000000','creationId:
  });
</script>
<link type="text/css" rel="stylesheet" href="/fontFace.osa"><link type="text/css" rel="stylesheet" href="/iframe.osa"></head>
<body class="dashboard-answers-widget-business-manager-dashboard" style="display: block;">
<div id="root"><div class="router-container" data-reactroot=""><div class="module-container slide-down-page module-10 search-only"><div class="module"><span><div class="widget-module kb-container kb-home-page" data-item-id"
<script charset="UTF-8">
window._STATIC_URL = '//static.parastorage.com/services/answers-widget/1.844.0/';
window._BASE_PATH = '/apps/widget/v1/v1a/b88ed9c-2e9b-42ca-9402-a2d3c3ce439/view/rw/';
window._INITIAL_STATE = {locale:"ru","appId":"https://data.wixanswers.com/api/v1","categoryTree":{"children":[{"id":"70b7e18b-850-402b-a27e-1e8c8e86c97c","parentId":"00000000-0000-0000-0000-000000000000","creationId:
</script>
<script src="/answers.ru.js.see.nazammi"/></script>
<script src="/app.bundle.min.js.see.nazammi"/></script>
<script id="custom-js" src="/file.js.see.nazammi"/></script>
</body></html>

```

Рисунок 3.11 - Код аналитики

Результатом этого кода является подвязанная аналитика сайта к платформе WIX, которая выводит нам статистику посещения сайта, с какого устройства заходил пользователь и т.д. А так же будет представлен временной график посещения на сайт, тем самым мы будем знать когда сайт пользуется большим спросом а когда наименьшим. Таким образом мы сможем более экономично тратить деньги на рекламу. Вы это можете видеть на рисунках 3.12, 3.13, 3.14.

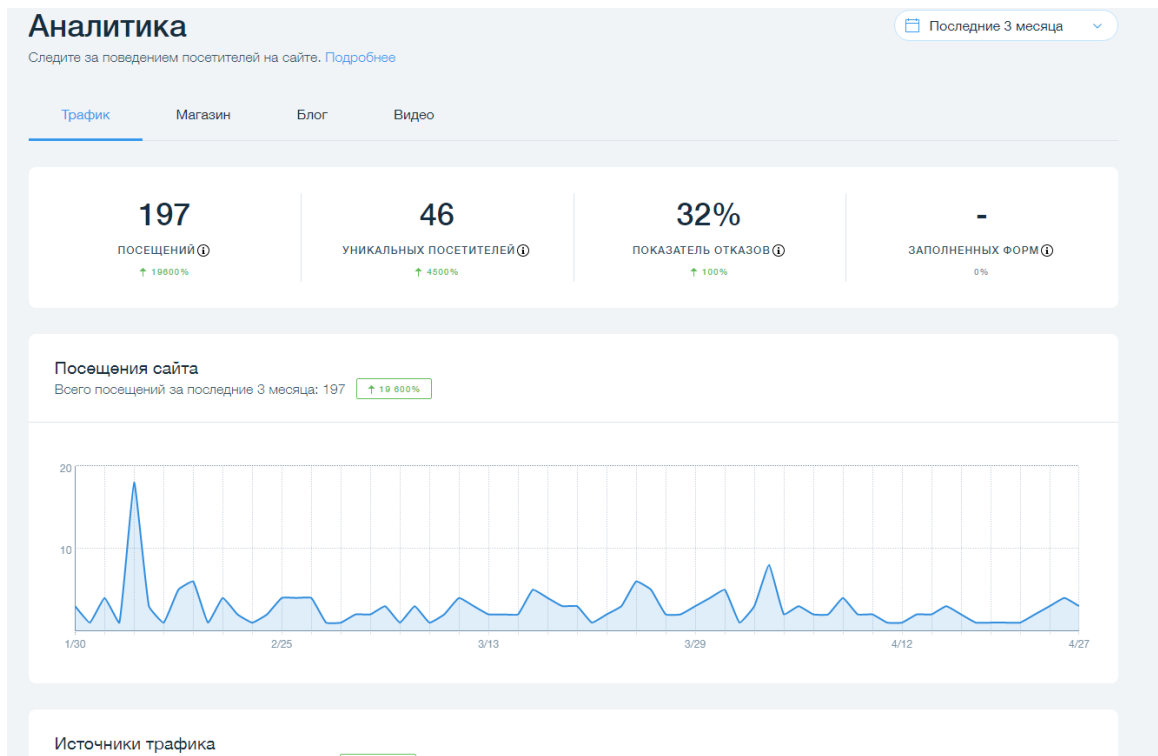


Рисунок 3.12 - Общая аналитика Сайта.

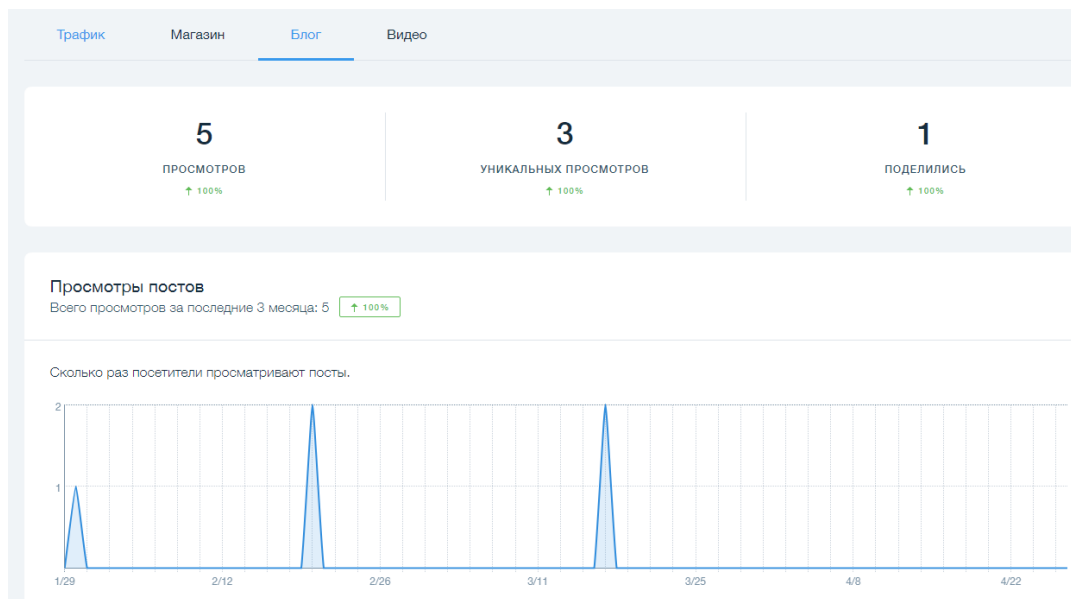


Рисунок 3.13 - Аналитика разных блоков сайта

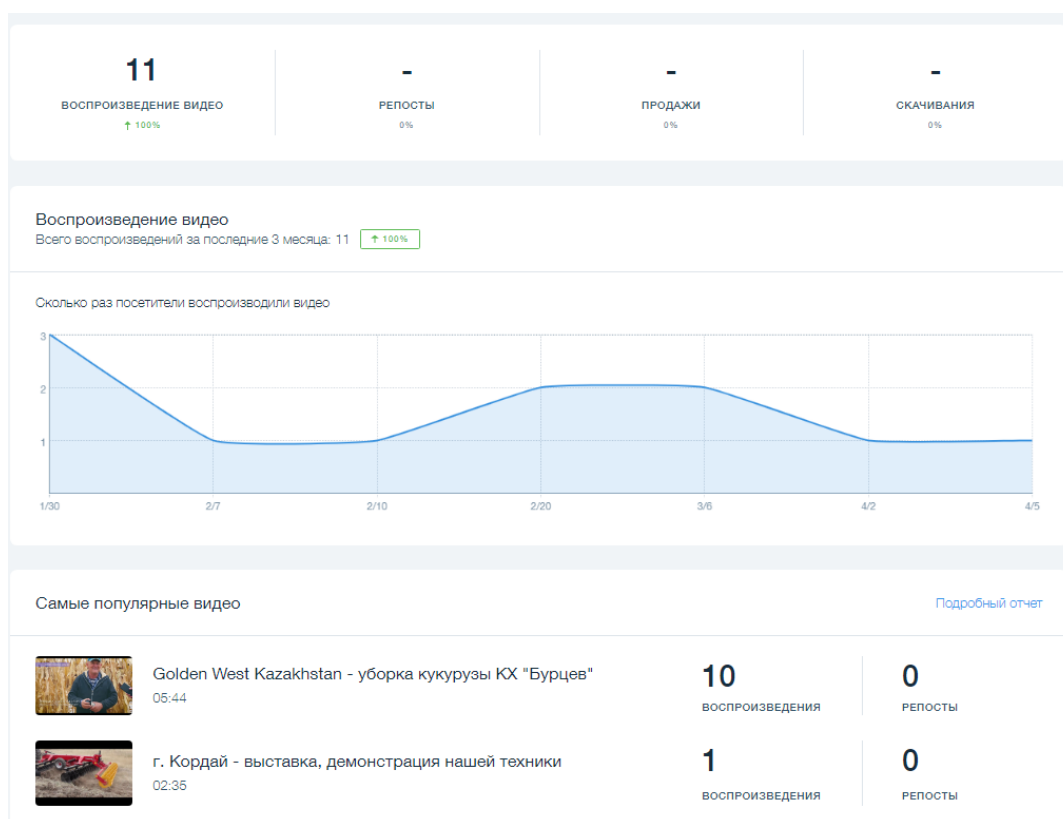


Рисунок 3.14 – Аналитика видео контента

На этих рисунках Вы видите что сайт успешно анализирует все данные которые поступают в код программы. На рисунка 3.15, 3.16 Вы сможете увидеть аналитику источника посещения сайта. То есть посещали ли его напрямую, или через гугл, ссылки и т.д. А так же с мобильного устройства или с компьютера.

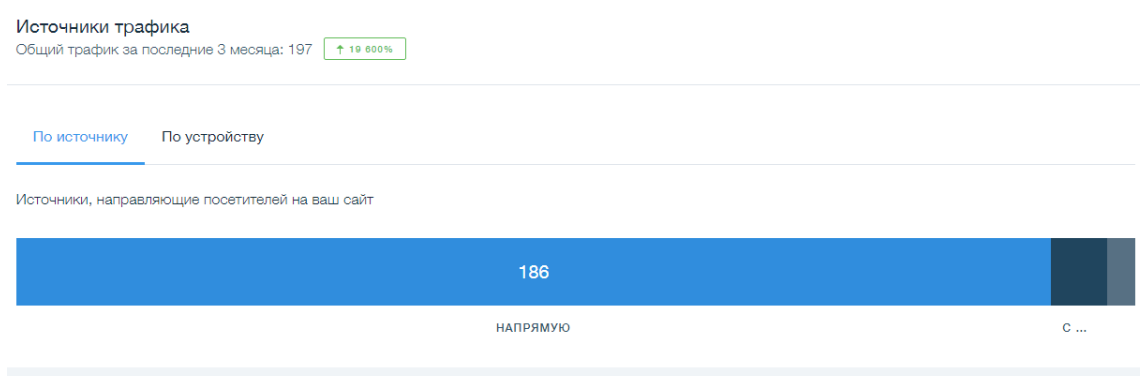


Рисунок 3.15 – Аналитика источника трафика

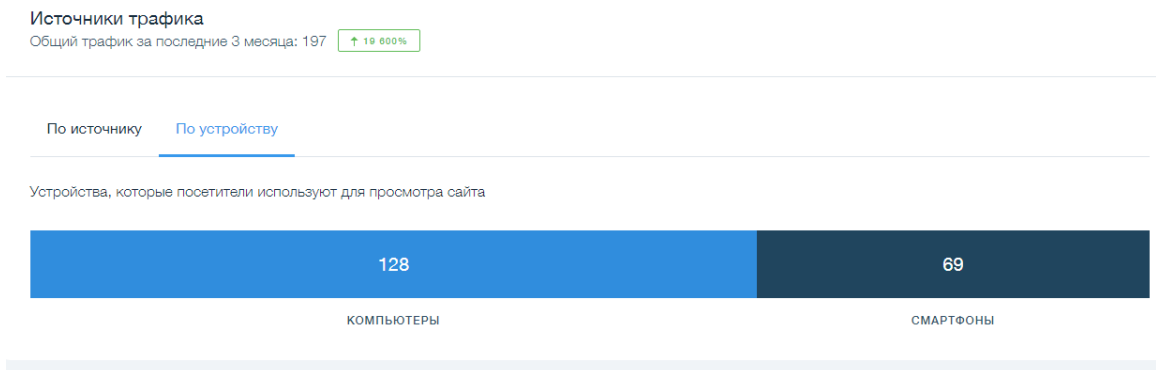


Рисунок 3.16 – Аналитика устройства посещения

Далее мне предстояло подвязать весь этот трафик и аналитику к сервисам Google, Yandex и т.д. так как моя дипломная работа является новым дополнением к мировой аналитики, и чтобы проверить их совместимость мне нужно было проверить на своем сайте. На рисунке 3.17 представлен фрагмент когда отвечающий за подвязку к сервисам.

```

44 <!-- META DATA -->
45 <script type="text/javascript">
46
47
48     var serviceTopology = {"cacheKillerVersion":"1","staticServerUrl":"https://static.parastorage.com/","
49     var santaModels = true;
50     var isStreaming = true;
51     var rendererModel = {"metaSiteId":"618e077d-c7dd-4c32-b032-963f172f5981","siteInfo":{"documentType":
52     var publicModel = {"domain":"kupiltraktor.kz","externalBaseUrl":"https://www.kupiltraktor.kz/","u
53
54     var isSAVRollout = "false";
55     var googleAnalytics = "";
56     var ipAnonymization = false;
57
58     var googleRemarketing = "";
59     var googleTagManager = "";
60     var facebookRemarketing = "";
61     var yandexMetrika = "";
62 </script>
63

```

Рисунок 3.17 – Код подвязки к сервисам

Эти сервисы помогают нам получать дополнительный трафик с интернета и анализировать его. Выдавать более подходящую рекламу для пользователя а так же узнавать его поведение, которое позволяет в будущем прогнозировать его поведения на веб-сайтах.

В своем сайте я использовал CSS кода который позволяет адаптивно проводить код анализа и других кодов на сайте. В каждом подразделе моего сайта есть по несколько блоков CSS кода, пример такого вы увидите на Рисунках 3.18, 3.19.

Рабочий стол > Diplom > Сайт > Аналитика_files

Поиск: Аналитика_fi

Имя	Дата изменения	Тип	Размер
module.bundle.min.js.Без названия	28.04.2020 21:46	Файл	158 КБ
module-registry-components.bundle.mi...	28.04.2020 21:46	Файл	1 735 КБ
module-registry-components.bundle.mi...	28.04.2020 21:46	Файл	1 188 КБ
module-registry-components.bundle.mi...	28.04.2020 21:46	Файл	1 611 КБ
module-registry-components.min(1)	28.04.2020 21:46	CSS-документ	30 КБ
module-registry-components.min(2)	28.04.2020 21:46	CSS-документ	41 КБ
module-registry-components.min	28.04.2020 21:46	CSS-документ	48 КБ
nr-1167.min.js.Без названия	28.04.2020 21:46	Файл	27 КБ
omni-setup-sidebar-component.bundle....	28.04.2020 21:46	Файл	232 КБ
omni-setup-sidebar-component.min	28.04.2020 21:46	CSS-документ	1 КБ
omni-setup-widjet.bundle.min.js.Без на...	28.04.2020 21:46	Файл	415 КБ
omni-setup-widjet.min	28.04.2020 21:46	CSS-документ	29 КБ
partners-team-module.bundle.min.js.Без...	28.04.2020 21:46	Файл	131 КБ
ping-settings-client-module.bundle.min....	28.04.2020 21:46	Файл	8 КБ
plugins.js.Без названия	28.04.2020 21:46	Файл	60 КБ
...

Рисунок 3.18 – фрагмент CSS файлов

Рабочий стол > Diplom > Сайт > О компании_files

Поиск: О компании_files

Имя	Дата изменения	Тип	Размер
focus-visible.min.js.Без названия	29.04.2020 13:59	Файл	3 КБ
frame-listener.bundle.min.js.Без названия	29.04.2020 13:59	Файл	13 КБ
galleriesCommon.min.js.Без названия	29.04.2020 13:59	Файл	5 КБ
gsap.min.js.Без названия	29.04.2020 13:59	Файл	57 КБ
handler(1)	29.04.2020 13:59	Chrome HTML Do...	11 КБ
handler	29.04.2020 13:59	Chrome HTML Do...	11 КБ
imageClientApi.js.Без названия	29.04.2020 13:59	Файл	30 КБ
imageZoom.min.js.Без названия	29.04.2020 13:59	Файл	38 КБ
languages(1)	29.04.2020 13:59	CSS-документ	146 КБ
languages(2)	29.04.2020 13:59	CSS-документ	148 КБ
languages	29.04.2020 13:59	CSS-документ	138 КБ
layout.min.js.Без названия	29.04.2020 13:59	Файл	136 КБ
linkBar.min.js.Без названия	29.04.2020 13:59	Файл	7 КБ
lodash.min.js.Без названия	29.04.2020 13:59	Файл	72 КБ
main-r.min.js.Без названия	29.04.2020 13:59	Файл	172 КБ
...

Рисунок 3.19 – фрагмент CSS файлов

4 Экономический расчет

4.1 Цели и задачи, решаемые в экономической части

В данном дипломном проекте описывается разработка системы анализа поведения посетителей веб сайтов (человек в веб среде) Программа разрабатывается для внедрения в сайт компании, чтобы рассчитать эффективность рекламы в интернете. В результате расчетов находится себестоимость прикладной программы.

Для нахождения себестоимости необходимо учесть:

- трудоемкость разработки приложения;
- материальные затраты;
- затраты на оплату труда;
- социальный налог;
- амортизация основных фондов;
- прочие затраты.

4.2 Расчет трудоемкости разработки ПП

Таблица 4.2.1 - Распределение работ по этапам и видам и оценка их трудоемкости

Этап разработк и	Вид работы	Трудоемкость разработки, чел.×ч.
1	Анализ и описание задачи	45
2	Разработка алгоритма конвертации текста	35
3	Разработка алгоритма конвертации файла	70
4	Реализация клиентской части	150
5	Разработка админ части проекта	160
6	Отладка и тестирование программы	110
7	Оформление документации, инструкции пользователю, пояснительной записки.	65
ИТОГО		635

4.3 Расчет затрат на разработку ПП

Определение затрат на разработку ПП производится путем составления соответствующей сметы, которая включает следующие статьи:

- материальные затраты;

- затраты на оплату труда;
- социальный налог;
- амортизация основных фондов;
- прочие затраты.

Таблица 4.3.1 - Затраты на материальные ресурсы

Наименование материального ресурса	Единица измерения	Количество	Цена за единицу, тг	Сумма, тг
Асер SP314-52 NX	шт.	1	230000.00	230000.00
ИТОГО				230000.00

Общая сумма затрат на материальные ресурсы (Z_M) определяется по формуле:

$$Z_M = \sum_{i=1}^n P_i * C, \quad (4.1)$$

- где, P_i - расход i -го вида материального ресурса, натуральные единицы;
 C_i - цена за единицу i -го вида материального ресурса, тг;
 i - вид материального ресурса;
 n - количество видов материальных ресурсов.

Если для разработки ПП используется электрооборудование, то необходимо рассчитать затраты на электроэнергию по форме, приведенной в таблице 4.3.2.

Общая сумма затрат на электроэнергию (Z_E) рассчитывается по формуле:

$$Z_E = \sum_{i=1}^n M_i * K_i * T_i * C, \quad (4.2)$$

- где, M_i - паспортная мощность i -го электрооборудования, кВт;
 K_i - коэффициент использования мощности i -го электрооборудования (принимается $K_i=0.8$);
 T_i - время работы i -го оборудования за весь период разработки ПП ч;
 C - цена электроэнергии, тг/кВт×ч;
 i - вид электрооборудования;
 n - количество электрооборудования.

Затраты на электроэнергию находятся исходя из продолжительности периода разработки ПП, количества кВт/ч, затраченных на проектирование ПП и тарифа за 1 кВт/ч. Тариф по городу Алматы для юридических лиц в 2020 году составляет 19,18 тенге за 1 кВт/ч с учетом НДС (согласно данным представленным на официальном сайте ТОО «АлматыЭнергоСбыт»).

$$Z_3 = 0,8 \cdot 0,8 \cdot 635 \cdot 19,18 \approx 7\,794,75 \text{ тг}$$

$$Z_3 = 0,4 \cdot 0,6 \cdot 635 \cdot 19,18 \approx 2\,923,03$$

Таблица 4.3.2 - Затраты на электроэнергию

Наименование оборудования	Паспортная мощность, кВт	Коэффициент использования мощности	Время работы оборудования для разработки ПП, ч	Цена электроэнергии, тг/кВт·ч	Сумма, тг
Асер SP314-52 NX	0,8	0,8	635	19,18	7 794,75
Освещение	0,4	0,6	635	19,18	2 923,03
ИТОГО					10 717,78

4.4 Расчет затрат на оплату труда

Зарботная плата Middle Full stack разработчика в 2020 составляет 240000 тенге (для города Алматы).

Рабочие часы сотрудника за месяц определяются по формуле:

$$Ч_м = N_м \cdot Ч_{рд}, \quad (4.3)$$

где, $Ч_м$ — рабочие часы сотрудника за месяц;

$N_м$ — количество рабочих дней за месяц;

$Ч_{рд}$ — количество рабочих часов в день.

$$Ч_м = 21 \cdot 7 = 147 \text{ ч}$$

Ставка работника за час работы рассчитана по формуле:

$$ЧС_i = \frac{ЗП_i}{ФРВ_i}, \quad (4.4)$$

Инженер-разработчик:

$$ЧС_i = \frac{240000}{147} = 1\,632,65 \text{ тг}$$

где, Z_{Pi} - заработная плата в месяц i -го работника, тг;
 ΦP_{Vi} - фонд рабочего времени в месяц i -го работника, час.

Для определения трудоемкости разработки ПП используются данные из таблицы 4.2.1.

Трудоемкость разработки Middle Full stack разработчика

$$T=45+35+70+150+160+110+65= 635 \text{ чел.} \times \text{ч.}$$

Затраты на оплату труда в сумме ($Z_{тр}$) определяется по формуле:

$$Z_{тр} = \sum_{i=1}^n ЧС_i * T_i, \quad (4.5)$$

где, $ЧС_i$ - часовая ставка i -го работника, тг;
 T_i - трудоемкость разработки ПП, чел.×ч;
 i - категория работника;
 n - количество работников, занятых разработкой ПП.

Инженер-разработчик:

$$Z_{тр} = 1\ 632,65 \cdot 635 = 1\ 036\ 732,75$$

Таблица 4.4.1 - Затраты на оплату труда

Квалификация	Трудоемкость разработки ПП, чел.×ч	Часовая ставка, тг/ч	Сумма, тг
Middle Full stack разработчик	635	1 632,65	1 036 732,75
ИТОГО			1 036 732,75

Дополнительная заработная плата:

$$Z_{доп} = Z_{тр} \cdot 10\%, \quad (4.6)$$

$$Z_{доп} = 1\ 036\ 732,75 \cdot 0.1 = 1\ 036\ 73,275$$

Фонд заработной платы:

$$\Phi_{зп} = Z_{тр.о} + Z_{доп}, \quad (4.7)$$

$$\Phi_{зп} = 1\ 036\ 732,75 + 1\ 036\ 73,275 = 1\ 140\ 406,02 \text{ тг}$$

4.5 Расчет затрат по социальному налогу

Согласно Налоговому кодексу Республики Казахстан социальный налог составляет 9,5% от фонда оплаты труда. Налоги уплаченные за работников юридическим лицом можно рассчитать по следующей формуле:

Таблица 4.5.1 – Формулы расчета социального налога

Уплаченные налоги юридическим лицом	10,46	ФОТ	1 140 406
СО (Социальные отчисления)	3,5	(ЗП - ОПВ)*3,5%	35 922,79
ВОСМСЮ (Отчисления на ВОСМСЮ)	2,0	ЗП*2%	22 808,12
СН (Социальный налог)	9,5	(ЗП - ОПВ - ВОСМСФ)*9,5%-СО	60 498,54
Всего уплаченные налоги			119 229,45

Расчет амортизационных основных фондов

Полная сумма амортизационных отчислений определяется по формуле:

$$Z_{AM} = \sum_{i=1}^n \frac{\Phi_i * N_{Ai} * T_{НИРi}}{100 * T_{Э\Phi i}}, \quad (4.9)$$

где, Φ_i - стоимость i -го ОФ, тг;

N_{Ai} - годовая норма амортизации i -го ОФ, %;

$T_{НИРi}$ - время работы i -го ОФ за весь период разработки ПП, ч;

$T_{Э\Phi i}$ - эффективный фонд времени работы i -го ОФ за год, ч/год;

i - вид ОФ;

n - количество ОФ.

Расчет годовой нормы амортизации ОФ:

$$N_{Ai} = \frac{100}{4} = 25$$

$$N_{Ai} = \frac{100}{T_{Ni}}, \quad (4.10)$$

где, T_{Ni} - возможный срок использования i -го ОФ, год;

Для определения времени работы ПО для разработки ПП используются данные из таблицы 1.1.

Время работы ПО Sublime text и Хаmpp для разработки ПП составляет 613 часов (реализация клиентской части проекта, реализация администраторской части проекта, отладка и тестирование программы).

$$T_i = 40 + 73 + 140 + 180 + 130 = 563 \text{ ч.}$$

Оборудование:

$$З_{AM} = \frac{230000 \cdot 25 \cdot 635}{100 \cdot 1920} = 19\,016,92 \text{ тг}$$

Таблица 4.5.2 - Амортизация основных фондов (ОФ)

Наименование оборудования и ПО	Стоимость оборудования и ПО, тг	Годовая норма амортизации, %	Эффективный фонд времени работы оборудования и ПО, ч/год	Время работы оборудования и ПО для разработки ПП, ч	Сумма, тг
Acer SP314-52 NX	230000	25	1920	635	19 016,92
ОС Windows 10	Бесплатно	-	1920	635	-
ПО Sublimetext	Бесплатно	-	1920	563	-
ПО Хамpp	Бесплатно	-	1920	563	-
ИТОГО					19 016,92

Таблица 4.5.3 - Смета затрат на разработку ПП

Статьи затрат	Сумма, тг	%
1. Материальные затраты, в том числе:		
- материалы	230 000,00	15,20
- электроэнергия	10 717,78	0,71
2. Затраты на оплату труда	1 140 406,02	75,37
3. Отчисления на социальные нужды	112 900,19	7,46
4. Амортизация основных фондов	19 016,92	1,26
5. Прочие затраты	0	
ИТОГО	1 513 040,91	100%



Диаграмма 1 – Смета затрат на разработку проекта

4.6 Определение возможной (договорной) цены ПП

Размер вероятной (договорной) стоимости ПП обязана устанавливаться с учетом производительности, свойства и сроков ее выполнения на уровне, отвечающем финансовым заинтересованностям клиента (потребителя) и исполнителя.

Договорная цена (C_d) для прикладных ПП рассчитывается по формуле:

$$C_d = Z_{\text{НИР}} \left(1 + \frac{P}{100} \right), \quad (4.11)$$

где, $Z_{\text{НИР}}$ - затраты на разработку ПП (из таблицы 1.6), тг;

P - средний уровень рентабельности ПП - 20 %

$$C_d = 1\,513\,040,91 * \left(1 + \frac{20}{100} \right) = 1\,513\,040,91 + 302\,608,18 = 1\,815\,649,09 \text{ тг}$$

Дальше ориентируется стоимость реализации с учетом налога на добавленную цену (НДС), ставка НДС уточняется законодательно Налоговым Кодексом РК. На 2020 год ставка НДС установлена в объеме 12%.

Цена реализации с учетом НДС рассчитывается по формуле:

$$Ц_p = Ц_d + Ц_d \cdot \text{НДС}, \quad (4.12)$$

$$Ц_p = 1\,815\,649,09 + 1\,815\,649,09 \cdot 0,12 = 2\,033\,527 \text{ тг}$$

В заключении нужно отметить что, цена реализации с учетом всех возможных затрат составляет 2 033 527 тенге, себестоимость разработки равна 1 513 040,91, прибыль разработки равна 302 608,18

4.7 Расчет эксплуатационных затрат

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

$$C = ЗП + ОТ + А + М + НР \quad (4.13)$$

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:

$$ЗП = \frac{O_c \cdot Ч_c \cdot 12}{\Phi_{p.v.}} \cdot t_{\text{общ}} \cdot 12 \cdot (1 + K_d) \quad (4.14)$$

где O_c - оклад специалиста, тенге/мес.;

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

$\Phi_{p.v.}$ - годовой фонд рабочего времени, час;

$t_{\text{общ}}$ - 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:

$$t_{\text{общ}} = \sum_{\beta=1}^n t_{\beta} \cdot K_{\beta} \quad (4.15)$$

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

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

В ходе эксплуатации системы предполагается появление серверных ошибок, на обработку одной ошибки в среднем уходит 2 часа, на протяжении месяца эксплуатации может возникать до 60 ошибок, помимо этого ожидается внесение изменений в интерфейс и доработка компонентов по требованию руководства. Всего компонентов в системе 30, на доработку одного уходит в среднем 2 часов.

$$t_{\text{общ}} = 2 \cdot 60 + 2 \cdot 30 = 120 + 60 = 180 \text{ часов}$$
$$ЗП = \frac{240\,000 \cdot 1 \cdot 12}{1920} \cdot 180 \cdot 12 \cdot 1 = 3\,240\,000 \text{ тенге}$$

Материальные затраты определяются по формуле:

$$M = \sum_{\mu=1}^n C_{\mu} \cdot N_{\mu} \quad (4.16)$$

где C_{μ} - цена μ -го вида единицы материальных затрат в условиях функционирования ИТ, тенге.

N_{μ} - используемое количество μ -го вида материальных затрат в месяц, ед.

В условиях функционирования разрабатываемой ИС, не предполагается материальных затрат.

Амортизационные отчисления производятся по установленным нормам амортизации, выражаются в процентах к балансовой стоимости оборудования и рассчитываются по формуле:

$$A = C_{\text{обор}} \cdot H_a \quad (4.17)$$

где H_a — норма амортизации (25 %);

$C_{\text{обор}}$ — первоначальная стоимость оборудования;

$$A = (230\,000 \cdot 0,25) = 57\,500$$

Отчисления по социальному налогу и накладные расходы рассчитываются так же, как и при разработке информационных технологий.

$$СН = 3\,240\,000 \cdot 10,46\% = 338\,904 \text{ тенге.}$$

$$НР = ЗП \cdot \frac{H_{\text{нр}}}{100} = 3\,240\,000 \cdot 0,7 = 2\,268\,000 \text{ тенге}$$

Таблица 4.7 – Эксплуатационные затраты

Затраты на эксплуатацию	Условное обозначение	Значение, тенге	В процентах от общей суммы
Заработная плата	ЗП	3 240 000	54,87%
Социальные отчисления и налоги	ОТ	338 904	5,74%,
Материальные затраты	М	0	0%
Амортизационные отчисления	А	57 500	0,97%
Накладные расходы	НР	2 268 000	38,42%
Итого		5 904 404	

C = 5 904 404 тенге

4.8 Расчет результатов от создания и использования ИС

Информационная система для управления проектированием предназначена в главную очередь для контроля работы проектной группы. Она упрощает процесс принятия решений по управлению, а также способна повысить эффективность и скорость работы сотрудников, за счет тайм-менеджмента.

Для оценки экономии от использования разрабатываемой ИС необходимо сравнить эксплуатационные расходы с ее применением и без.

Статьи затрат при применении ИС включают в себя:

- заработная плата специалиста, осуществляющего поддержку и сопровождение системы;
- износ оборудования;

- накладные расходы.

Данная система не предполагает расхода каких-либо материалов.

В организации имеется два куратора проектов (месячная заработная плата – 210 000 тенге), основной задачей которых является осуществление контроля работы проектной группы, управление проектированием.

ЗП = 5 040 000 тенге

СО = 527 184

В их распоряжении находятся два персональных компьютера (стоимость одного ~ 300 000 тенге), принтер (150 000 тенге) и прочая компьютерная периферия (~50 000 тенге); итого затрат на оборудование – 500 000 тенге.

Износ оборудования рассчитывается исходя из 25 % амортизационных отчислений за год.

$A = (300\,000 \cdot 0,25) + (150\,000 \cdot 0,25) + (50\,000 \cdot 0,25) = 125\,000$ тенге

На заправку картриджей, бумагу прочую канцелярию в месяц уходит порядка 30 000 тенге.

Общие накладные расходы составят:

$$НР = ЗП \cdot \frac{Н_{нр}}{100} = 5\,040\,000 \cdot 0,7 = 3\,528\,000 \text{ тенге}$$

Статьи затрат без применения ИС включают в себя:

- заработная плата специалистов – кураторов проектов, которые осуществляют управление проектами;
- износ используемого ими оборудования;
- расход материалов (к примеру канцелярия – бумага, картридж и т.д.)
- накладные расходы.

Таблица 5.8 – Годовые эксплуатационные затраты

Статьи	Без применения ИС (2 куратора проекта)	С применением ИС (1 сотрудник – программист, сопровождающий систему)
Годовая заработная плата	5 040 000	3 240 000
Социальные отчисления и налоги	527 184	338 904
Расходуемые материалы	500 000	0
Амортизационные отчисления	125 000	57 500
Накладные расходы	3 528 000	2 268 000
Всего	9 720 184	5 904 404

Ожидаемая условно-годовая экономия определяется по формуле:

$$\mathcal{E}_{\text{уг}} = C_1 - C_2 + \sum \mathcal{E}_i \quad (4.18)$$

где $\mathcal{E}_{\text{уг}}$ - величина экономии, тенге;

C_1 и C_2 – показатели текущих затрат по базовому и внедряемому вариантам, тенге;

$\sum \mathcal{E}_i$ – ожидаемый дополнительный эффект от различных факторов, тенге.

$$\mathcal{E}_{\text{уг}} = C_1 - C_2 = 3\,815\,780 \text{ тенге}$$

4.9 Расчет основных показателей экономической эффективности

Так как разработанная информационная система несет более социальный эффект, чем экономический, целесообразно оценивать его эффективность за счет экономии в сравнении с предыдущим периодом работы без использования ИС.

Величина ожидаемого годового экономического эффекта от внедрения ИС рассчитывается по формуле:

$$\mathcal{E}_{\text{г}} = \mathcal{E}_{\text{уг}} - K \cdot E_{\text{н}} \quad (4.19)$$

где $\mathcal{E}_{\text{г}}$ - ожидаемый годовой экономический эффект, тенге;

$\mathcal{E}_{\text{уг}}$ — ожидаемая условно-годовая экономия, тенге;

K — капитальные вложения, тенге;

$E_{\text{н}}$ - нормативный коэффициент экономической эффективности капитальных вложений.

Нормативный коэффициент экономической эффективности капитальных вложений определяется по формуле:

$$E_{\text{н}} = \frac{1}{T_{\text{н}}} \quad (4.20)$$

где $T_{\text{н}}$ — нормативный срок окупаемости капитальных вложений, лет.

Нормативный срок окупаемости капитальных вложений принимается исходя из срока морального старения -технических средств и проектных решений ИС ($T_{\text{н}}=1,2,3\dots n$), для программных продуктов срок окупаемости принимаем равным 4 года.

$$E_{\text{н}} = \frac{1}{4} = 0,25$$

$$\text{Э}_{\text{уг}} = 2\,033\,527 - 3\,815\,780 \cdot 0,25 = 1\,079\,582 \text{ тенге}$$

Расчетный коэффициент экономической эффективности капитальных вложений составляет:

$$E_p = \frac{\text{Э}_g}{K} \quad (4.21)$$

где E_p - расчетный коэффициент экономической эффективности капитальных вложений;

Э_g — ожидаемая годовая экономия, тенге;

K — капитальные вложения на создание системы, тенге.

$$E_p = \frac{3\,815\,780}{2\,033\,527} = 1,87$$

Расчетный срок окупаемости капитальных вложений составляет:

$$T_p = \frac{1}{E_p} \quad (4.13)$$

где E_p - коэффициент экономической эффективности капитальных вложений.

$$T_p = \frac{1}{1,87} = 0,53 \text{ года} \approx 6,3 \text{ месяцев}$$

Таблица 5.9 – Показатели сравнительной экономической эффективности от внедрения программного продукта

Наименование показателей	Значение
Условная годовая экономия затрат, тенге	1 079 582
Коэффициент экономической эффективности капитальных вложений (E_p)	1,87
Срок окупаемости капитальных вложений (T_p)	0,53

Таким образом, разработанная информационная система позволила не только упростить процесс управления работой проектной группы организации и процесс принятия решений руководством, но и значительно сэкономить на различных факторах. Во-первых, автоматизация процесса управления позволит больше не использовать услуги дополнительных кураторов проектов по филиалам, так как теперь отсутствует необходимость осуществлять контроль непосредственно на месте филиала, вся работа организации сосредоточена в ИС. Во-вторых, значительно снизится расход материалов, за счет того, что назначение на задания и все изменения по заданиям можно сделать прямоком в системе, не прибегая к формированию документов по назначению и их распечатке. Ожидаемый годовой экономический эффект составил 1 079 582 тенге. Приложение окупится в первые 5,3 месяцев использования.

5 Безопасность жизнедеятельности

В данном дипломном проекте описывается разработка системы анализа поведения посетителей веб сайтов (человек в веб среде). Программа разрабатывается специально для ТОО «УТО Agro Kazakhstan» в целях изучения поведения посетителей сайта компании, для более эффективного внедрения рекламы. Целью данной работы является разработка моделей и методов прогнозирования поведения людей в веб-среде.

Автоматизирование вышеизложенных услуг позволяет работникам рекламных компании больше времени уделять улучшению качества самих программ. Данное web-приложение будет использоваться пользователями через их мобильные устройства или через их персональные компьютеры. Чаще всего его используют через мобильные устройства при плохом освещении. Разработка данного проекта производилась в бизнес центре.

5.1 Анализ потенциально опасных и вредных факторов в офисе, воздействующих на персонал

В рассматриваемом помещении работают разработчик, дизайнер, руководитель и менеджеры которые имеют свое рабочее место. При разработке программного продукта важную роль играет правильная организация условий труда в рабочем помещении.

Рабочее место персонала должно отвечать техническим нормам предназначенным для данного помещения, с учетом наличия всех защитных правил, процедур и критериев во избежание воздействия на здоровье персонала производственных факторов способных нанести вред как в краткосрочной, так и в долгосрочной перспективе.

Показатели данных факторов должны находиться в интервале, не превышая допустимые значения предусмотренные правовыми, техническими и санитарно-техническими нормами. Нормативные правовые акты в области безопасности и охраны труда устанавливают организационные, технические, технологические, санитарно-эпидемиологические, медицинские и иные нормы, правила, процедуры и критерии, направленные на сохранение жизни и здоровья работников в процессе их трудовой деятельности.

Наиболее характерными факторами риска для работников офиса (согласно закону N 528-2 «о безопасности и охране труда») являются перенапряжение зрения и внимания; превышение норм электромагнитного, рентгеновского, ультрафиолетового и инфракрасного излучения; чрезвычайно высокий уровень статического электричества и показатель запыленности воздуха рабочей зоны, длительное нахождение в сидячем положении, монотонность труда, повышенное содержание в воздухе рабочей зоны двуокиси углерода, озона, аммиака, фенола и формальдегида. Часто работники

офисных помещений жалуются на следующие симптомы: жжение в глазах, покраснение век и конъюнктивы, чувство инородного тела или песка под веками, боли в области глазниц и лба, затуманивание зрения, замедленная перефокусировка с ближних объектов на дальние. Причиной тому является переутомление зрительного аппарата при длительной работе с экраном без перерывов. Данные симптомы имеют наименование данное Всемирной Организацией Здравоохранения, и именуется как «компьютерный зрительный синдром».

Так как данная работа связана с длительным нахождением за ноутбуком, необходимо проанализировать системы пожаротушения для обеспечения благоприятных условий работы.

Рассмотрим пример типового производственного помещения в бизнес центре. В помещении, кроме оборудования и рабочих мест, могут находиться кондиционеры и обогреватели. Помещение, в котором были поставлены 12 ноутбуков. Работают 12 человек: 2 программиста, 2 менеджера, 2 контент менеджера, 1 видео-оператор, 1 аудитор, 1 HR-менеджер и 3 руководителя.

Долговременное нахождение человека в зоне комбинированного влияния всевозможных не очень благоприятных окружающих факторов, имеет возможность привести к профессиональному заболеванию.

Помещение имеет следующие параметры:

а) размеры рабочего помещения:

- 1) высота, h – 3 м;
- 2) ширина, s – 9 м;
- 3) длина, l – 17 м;

б) общая площадь помещения составляет 153м^2

в) искусственное освещение – светильники: 6 светильников в каждом по 4 люминесцентные лампы.

В частности при организации рабочего места пользователя ПЭВМ должны быть соблюдены следующие основные условия:

- А) Правильное размещения рабочего места;
- Б) Необходимое рабочее пространство для производственного процесса;
- В) Искусственное и естественное освещение;
- Г) Допустимый уровень акустического шума;
- Д) Системы вентиляции и кондиционирования воздуха;

Характеристики используемого оборудования

При разработке и дальнейшей эксплуатации программного продукта используется следующее оборудование:

- А) ноутбуки – 12 шт.;
- Б) Люминесцентные лампы Т8/18W, 15 светильников по 4 лампы $E=400\text{лк}$;
- В) Кондиционер;
- Г) Система пожаротушения.

В таблице 5.1 приведены оптимальные нормы параметров микроклимата с учётом периода года. Оборудование, установленное в рабочем помещении, не является источником выделения тепла.

Таблица 5.1 – Оптимальные нормы микроклимата для помещений с ноутбуков

Период года	Категория работ	Температура воздуха °С не более	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	Легкая – 1а	22-24	40-60	0,1
Теплый	Легкая – 1а	23-25	40-60	0,1

Говоря о пожарах и пожарной безопасности можно сказать, что это один из важнейших факторов организации труда. При неправильной организации труда можно привести условия труда к очень невыгодным в чрезвычайном случае. Потому что если не установить пожарную систему или пожарную сигнализацию можно сказать, что в случае пожара, к примеру в соседней комнате вы можете попросту поздно узнать и вы не сможете предпринять меры по эвакуации или же устранения самого пожара. В нашем офисе пожар может начаться только по следующим условиям:

- 1)короткое замыкание;
- 2)оголенные провода;
- 3)возгорание от плиты;

Рассмотрим несколько видов пожарных систем и в чем их особенности.

Так как в нашем офисе уже проведена пожарная система с мы рассмотрим какие они бывают. Они бывают нескольких видов: сигнализации, автоматические системы пожаротушения, системы пожаротушения по кнопке. К сигнализациям можно отнести датчики, которые реагируют, либо на температуру (что бывает не очень выгодно), либо на дым (что является более эффективным среди сигнализаций). Также хочу отметить системы пожаротушения. Они также делятся на несколько типов: водяные, порошковые, газовые, аэрозольные и пенные. Они отличаются по площади покрытия, продолжительности работы, расходу воды.

Если сравнивать их между собой, то одно из самых распространенных является классическая система пожаротушения(водяная). Так как если сравнивать их по ценовой политике она является одна из самых дешевых и в то же время может справиться с пожаром в офисном или жилом помещении.

Пенная система пожаротушения является тоже не самой дорогостоящей, но в сравнении с водяной она более эффективна. Так как она справляется с

пожаром любой сложности за более короткое время при этом потребляя меньше воды.

На основе вышеизложенного материала мы сможем рассчитать свою пожарную систему и обосновать является ли она эффективной или нет.

5.2 Методика расчета установок водяного и пенного пожаротушения

Системы автоматического пожаротушения имеют разные типы огнетушащего вещества. За основное огнетушащее вещество мы взяли пену - коллоидная система из заполненных углекислым или инертным газом пузырьков. Данный вид пожаротушения минимизирует объем ущерба при пожаре, который может быть нанесен электронным устройствам и мебели.

Пенная пожарная система состоит из аппаратов для генерации пены, пеносмесителя, приборов дозирующей подачи, бака, водоводов и оросителей. Оросители делятся на спринклерные и дренчерные. При срабатывании датчика сигнала автоматически смешивается пена и вода, далее данная смесь распространяется через распылители. Употребление и воды, и пены для ликвидации пламени остается важным и эффективным способом борьбы с пожарами.

В нижеуказанной таблице 5.2 приводятся характеристики спринклерных установок внутри офиса.

Таблица 5.2. Параметры спринклерных установок

Расположение оборудования в офисе	Интенсивность орошения водой, л/(см ²)	Максимальное расстояние между спринклерными оросителями, м	Время работы установки, мин
Ноутбуки, видеорекамеры	0,65	3	20

Гидравлический расчет выполняется при условии водоснабжения от основного источника водоснабжения, в нашем случае центральное водоснабжение. Максимально давление на узле управления должно быть 1,0 Мпа.

Расчет расходы воды/раствора пенообразователя через распылитель/ороситель/генератор Q_d , л/с, производится по формуле 5.1:

$$Q_d = k \cdot \sqrt{H}, \quad (5.1)$$

где k – коэффициент производительности распылителя по таблице 5.3;

H – свободный напор перед распылителем/генератором/оросителем, м.

Таблица 5.3. Коэффициенты производительности оросителя

Ороситель (генератор)	Значение коэффициента k	Минимальный свободный напор, м	Максимальный допустимый напор, м
Водяной спринклерный и дренчерный с диаметром выходного отверстия, мм:			
8	0,20	5	100
10	0,31	5	100
12	0,45	5	100
15	0,71	10	100
20	1,25	10	100
Эвольвентный:			
ОЭ-16	0,27	15	80
ОЭ-25	0,66	15	80
ОЭ-50	2,73	15	80
Генератор пенный:			
ГЧС, ГЧСм	1,48	15	45
Пенный розеточный (ОПСР, ОПДР) с диаметром выходного отверстия, мм:			
10	0,31	15	100
15	0,71	15	100
Пенный (ОПС, ОПД)	0,55	15	80

Возьмем диаметр выходного отверстия распылителя 15 мм. Тогда по формуле 5.1 рассчитаем минимальный и максимальный расходы воды/раствора пены:

$$Q_{d \min} = 0,71 \cdot \sqrt{10} = 2,25 \text{ (л/с)}$$

$$Q_{d \max} = 0,71 \cdot \sqrt{100} = 7,1 \text{ (л/с)}$$

Для спринклерной установки расход воды/раствора пены определяется по следующей формуле:

$$Q = a \cdot b \cdot n \cdot q_n, \quad (5.2)$$

где a – расчетная длина одновременно орошаемой части, 20 м;

b — наибольшая ширина совмещенных столов, м;

n — количество техники;

q_n —интенсивность орошения, принимается по таблице 5.2.

В офисе длина книжных стеллажей равна 3 м, количество техники 12. Тогда по формуле 5.2:

$$Q = 20 \cdot 3 \cdot 12 \cdot 0,65 = 468 \text{ (л/с)}$$

Для расчета потерь напора на расчетном участке трубопроводов H_1 , м, воспользуемся следующей формулой:

$$H_1 = \frac{Q^2}{B}, \quad (5.3)$$

где Q – расходы воды/раствора пены на расчетном участке трубопровода, л/с;

B – характеристика трубопровода, рассчитываемая о формуле 5.4:

$$B = \frac{k_l}{l}, \quad (5.4)$$

где k_l – коэффициент, параметр труб, принимается по таблице 5.4;

l – длина расчетного участка трубопровода, м.

Таблица 5.4. Параметры труб

Трубы	Диаметр условного прохода, мм	Диаметр наружный, мм	Толщина стенки, мм	Значение k_l
Стальные электро-сварные (ГОСТ 10704-76)	15	18	2,0	0.0755
	20	25	2,0	0.75
	25	32	2,2	3.44

В рассматриваемом офисе используются стальные водогазопроводные трубы с диаметром условного прохода 15 мм. Для такого типа труб коэффициент k_l принимается 0,18. Длина расчетного участка 2м. Тогда по формуле 5.4 имеем:

$$B = \frac{0,18}{2} = 0,09$$

Далее по формуле 5.3 рассчитываем потери:

$$H_1 = \frac{468^2}{0,09} = 2\,433\,600 \text{ м}$$

Потери напора в узлах управления установок H_2 , м, определяются по формуле:

$$H_2 = \varepsilon \cdot Q^2, \quad (5.5)$$

где ε - коэффициент потерь напора в узле управления по таблице 5.5.

Таблица 5.5. Коэффициент потерь напора в узле управления

Узлы управления	Тип клапана	Диаметр клапана, мм	Коэффициент потерь напора ε
Спринклерной и дренчерной установок	БКМ, КМ	100	$2,35 \cdot 10^{-3}$
		150	$7,7 \cdot 10^{-4}$
		200	$1,98 \cdot 10^{-4}$
Дренчерной установки	ГД	65	$4,8 \cdot 10^{-2}$
		100	$6,34 \cdot 10^{-3}$
		150	$1,4 \cdot 10^{-3}$

По формуле 5.5 рассчитываем:

$$H_2 = 7,7 \cdot 10^{-4} \cdot 468 = 0,3603 \text{ (м)}$$

Для объемного пожаротушения объем пенообразующего раствора V_1 , м³, рассчитывается следующим образом:

$$V_1 = \frac{k_2 \cdot V}{k_3}, \quad (5.6)$$

где k_2 – коэффициент разрушения пены по таблице 5.6;

k_3 – кратность пены;

V – объем помещения, м³.

Таблица 5.6. Коэффициент разрушения пены

Тип горючего материала	Коэффициент k_2	Продолжительность работы установки
Твердые	3	25
Жидкие	4	15

Кратность пены это отношение объема пены к объему раствора пенообразователя. У нас пена средней кратности с кратностью 15. Объем защищаемого склада 459 м³. Подставим в формулу 2.6 все значения:

$$V_1 = \frac{3 \cdot 459}{15} = 91,8 \text{ (м}^3\text{)}$$

Далее рассчитаем количество одновременно работающих генераторов пенного раствора:

$$n = \frac{V_1}{Q_d \cdot t}, \quad (5.7)$$

где Q_d – примерная производительность одного генератора пенообразующего раствора, для генератора средней кратности 20 м³/мин;

t – продолжительность работы установки с пеной средней кратности, мин. Для пены средней кратности и помещения с твердыми горючими материалами продолжительность работы установки будет равна 25 мин.

$$n = \frac{91,8}{20 \cdot 25} = 0,184$$

Одновременно будет работать 0,184 генераторов пенного раствора.

Рассмотрим схему офиса, чтобы наглядно увидеть площадь, которую пожарная система будет покрывать в случае пожара. (Рисунок 5.1)

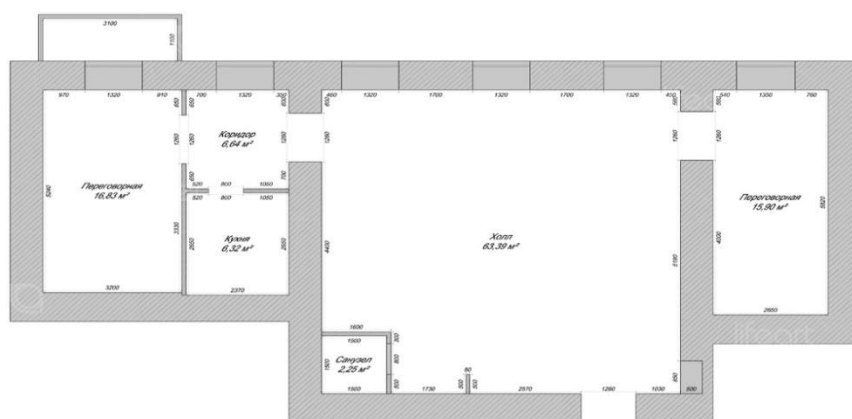


Рисунок 5.1 – схема офиса

5.3 Расчет искусственного освещения

Далее осуществляем проверку площади боковых световых проемов здания, предназначенных для обеспечения необходимыми показателями освещенности рабочего пространства. Разряд зрительной работы – III. Нормируемая освещённость – 400 лк. В качестве светильника возьмем люминесцентные лампы PLE-TPRO23/230. Длина светильника 80 см, ширина 20 см. Схема освещенности представлена на рисунке 5.3.1.

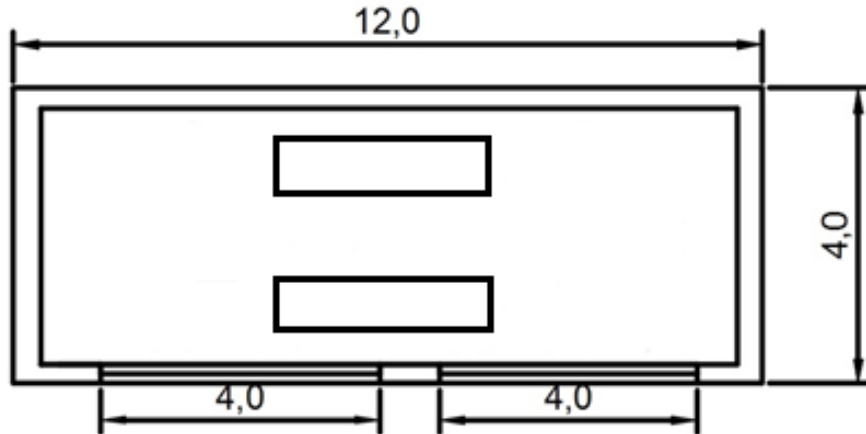


Рисунок 5.3.1 – Схема расчета освещенности.

Таблица 5.3 – Технические характеристики ламп PLE-TPRO23/230

Тип	Мощность, Вт	Напряжение, В	Световой поток, лм	Длина, см	Ширина, см
PLE-TPRO	23	230-240	1500	80	20

Для начала идет проверка соответствия минимальной заданной освещенности при имеющихся 6 светильниках.

$$E = \frac{N \cdot n \cdot \Phi_{л} \cdot \eta}{k_z \cdot S_{oc} \cdot Z}, \quad (5.7)$$

где, S_{oc} – площадь помещения = 153;
 k_z – коэффициент запаса = 1,2;

N – количество светильников=2;
 Z –коэффициент неравномерности освещения, $Z = 1,1$;
 n – количество ламп в светильнике = 4;
 $\Phi_{\text{л}}$ – световой поток выбранной лампы, $\Phi_{\text{л}} = 1500$ лм
 η – коэффициент использования, $\eta = 60\%$

$$E = \frac{6 \cdot 2 \cdot 1500 \cdot 0,6}{1,2 \cdot 153 \cdot 1,1} \approx 187$$

При 2 светильниках освещенность примерно равна 131 лк, что не соответствует условиям труда. В следствие чего возникает необходимость, увеличить количество светильников для обеспечения приемлемой освещенности. Для этого необходимо произвести расчеты для реконструкции:

Коэффициенты отражения от потолка стен и пола соответственно равны:

$$\rho_{\text{пот}} = 70 \%$$

$$\rho_{\text{ст}} = 50 \%$$

$$\rho_{\text{пол}} = 30 \%$$

Вычислим высоту подвеса светильника над рабочей поверхностью по формуле 5.6:

$$H = h - h_p - h_c, \quad (5.7)$$

где: h_c – расстояние от светильника до перекрытия, $h_c = 0,1$ м;

h_p – высота рабочей поверхности над полом, $h_p = 0,6$ м;

h – высота помещения, $h=3$ м.

$$H = 3 - 0,1 - 0,6 = 2,3 \text{ м}$$

Лучшее расстояние от окна до светильника определяется по формуле 5.8:

$$L = \lambda \cdot H, \quad (5.8)$$

где, $\lambda = 0,6 \div 1,5$

$$L = 1,5 \cdot 2,3 = 3,8 \approx 4 \text{ м}$$

Расстояние между светильниками:

$$L_b = L_a - 1,5 = 3,2 - 1,2 = 2 \text{ м}$$

Расстояние от стены до ближайшего светильника, когда работа у стены не проводится, определяем по формуле 5.9

$$l_a = l_b = 3/3 = 1, \quad (5.9)$$

$$l_1 = 0,4 \cdot 3,2 = 1,28 \text{ м}$$

Определяем индекс помещения по формуле 5.10

$$i = \frac{L \cdot B}{h_p \cdot (L + B)}, \quad (5.10)$$

$$i = \frac{4 \cdot 3}{2.175 \cdot (4 + 3)} = 0,79$$

Коэффициент использования в данном случае равен $\eta = 60\%$, коэффициент запаса равен $k_z = 1,2$

Определим количество люминесцентных ламп по формуле 5.10

$$N = \frac{E \cdot k_z \cdot S_{OC} \cdot Z}{n \cdot \Phi_{л} \cdot \eta} \quad (5.10)$$

где, S_{OC} – площадь помещения;

k_z – коэффициент запаса;

E – заданная минимальная освещенность, $E = 400$ лк;

Z – коэффициент неравномерности освещения, $Z = 1,4$;

n – количество ламп в светильнике;

$\Phi_{\text{л}}$ – световой поток выбранной лампы, $\Phi_{\text{л}} = 3570$ лм
 η – коэффициент использования, $\eta = 60\%$

$$N = \frac{400 \times 1,2 \times 153 \times 1,4}{4 \times 1500 \times 0,6} = 28,56 \approx 29$$

Всего для создания нормируемой освещенности 400 лк необходимо 29 люминесцентных лампы серии PLE-TPRO23/230, мощность каждой лампы должна быть не меньше 25 Вт, что соответствует действительности, а значит имеющегося в наличии освещения достаточно для соответствия санитарным нормам.

Заключение

В ходе выполнения дипломного проекта была разработана веб-сайт с формулами для вычисления поведения и анализа посетителя на сайтах. В рамках проекта было решено разработать веб-сайт для выявления формул поведения человека в веб-среде.

При реализации дипломного проекта были выполнены следующие задачи:

- был разработан веб сайт с алгоритмом анализа поведения посетителя веб сайта;
- сайт был подвезан к мировым рекламным сервисам;
- выведены формулы прогнозирования посетителей;
- установлена аналитическая система от WIX;
- улучшена рекламная компания сайта;
- создана маркетинговая страничка, а также повышена работоспособность и эффективность предприятий.

Было произведено экономическое обоснование целесообразности разрабатываемого программного продукта, где было сделано следующее заключение: цена реализации окупает все затраты, потрачены на разработку. Прибыль от реализации проекта равна 3 815 780 тенге;

Также были исследованы условия труда в помещении, где возможно будет применяться программа, проведен анализ потенциально опасных и вредных факторов в офисе, воздействующих на персонал и были предложены мероприятия по улучшению качества освещения в помещении, а также был произведен расчет систем кондиционирования.

Список литературы

- 1 Основы безопасности жизнедеятельности: Учебное пособие / сост.: М.К.Дюсебаев, Ж.С.Абдимуратов. -Алматы: АУЭС, 2013.-79 с
- 2 Калюжный Е.А. Безопасность жизнедеятельности: Учебное пособие / Е.А. Калюжный, С.В. Михайлова, С.Г. Напреев, Д.Г. Сидоров. - Арзамас:
- 3 Абрамов В.В. Безопасность жизнедеятельности: Учебное пособие для вузов. – Спб.: Питер, 2013. – 365 с.
- 4 Вершина, Г.А. Охрана труда: учебник / Г. А. Вершина и др. – Минск: ИВЦ Минфина, 2017. – 412 с
- 5 ГОСТ 12.1.004-91 «Система стандартов безопасности труда.»
- 6 Г. Боканова Методические указания по выполнению экономической части дипломных работ Алматы, АУЭС, 2020 – 35с.
- 7 Техничко-экономическое обоснование дипломных проектов Брест, БГТу, 2014 – 15с.
- 8 Методические указания к выполнению расчетно-графической работы для студентов специальности 080801 «Прикладная информатика (по областям)» Уфа, 2010 – 12с.
- 9 Методические указания по выполнению экономической части дипломных работ Москва, Московский университет им. С.Ю. Витте, 2016 – 21с.
- 10 Симионов Ю.Ф., Боромотов В.В. Информационный менеджмент. — Ростов н.Д: Феникс, 2013, 250с.
- 11 Бардзелл, Джеффри Macromedia Dreamweaver MX 2004 с ASP, ColdFusion и PHP. Из первых рук (+ CD-ROM) / Джеффри Бардзелл. - М.: Эком, 2017. - 560 с.
- 12 Бенкен, Е. PHP, MySQL, XML. Программирование для Интернета / Е. Бенкен. - М.: БХВ-Петербург, 2016. - 352 с. Wikipedia: Парковочные часы / URL: https://en.wikipedia.org/wiki/Parking_meter
- 13 Прохоренок, Николай HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера (+ CD-ROM) / Николай Прохоренок. - М.: БХВ-Петербург, 2012. - 912 с.
- 14 Andy, Harris HTML, XHTML and CSS All–In–One For Dummies® / Andy Harris. - Москва: Наука, 2014. - 173 с.
- 15 Ben, Henick HTML & CSS – The Good Parts / Ben Henick. - Москва: СИНТЕГ, 2013. - 350 с.
- 16 Ed, Tittel HTML, XHTML & CSS For Dummies® / Ed Tittel. - Москва: Гостехиздат, 2012. - 416 с.
- 17 Информационные технологии управления: Учебное пособие/ М.В. Бастриков, О.П. Пономарев; Институт «КВШУ».– Калининград: Изд-во Ин-та «КВШУ», 2005.– 140 с.

18 Информационные технологии управления: Учеб. пособие для вузов/ под ред. проф. Г.А. Титоренко. — 2-е изд., доп. — М.: ЮНИТИ-ДАНА, 2006. - 439 с.

19 Чудновский А.Д. Жукова М.А. Информационные технологии управления в туризме: учебное пособие.-М.:Кнорус,2009 -101 с.

20 Плотникова, Н.И. Комплексная автоматизация туристского бизнеса. Ч.II:Информационные технологии в сфере гостеприимства:Учебно-методическое пособие.-М.: Советский спорт, 2001 -208 с

21 Харрис Энди PHP/MySQL для начинающих; КУДИЦ-Образ - М., 2016. - 384 с.

22 Дронов, В. HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов / В. Дронов. - М.: БХВ-Петербург, 2014. - 138 с.

23 Дронов, Владимир HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов / Владимир Дронов. - М.: БХВ-Петербург, 2013. - 416 с.

24 Бенкен, Е. С. PHP, MySQL, XML. Программирование для Интернета (+ CD-ROM) / Е.С. Бенкен. - М.: БХВ-Петербург, 2011. - 304 с.

25 Зандстра Мэтт PHP. Объекты, шаблоны и методики программирования; Вильямс - М., 2016. - 560 с.

26 Колисниченко Денис PHP и MySQL. Разработка Web-приложений; БХВ-Петербург - М., 2017. - 560 с.

27 Квинт, Игорь Создаем сайты с помощью HTML, XHTML и CSS / Игорь Квинт. - М.: Питер, 2014. - 448 с.

Приложение А

Техническое задание

Техническое задание для разработки системы анализа поведения посетителей веб сайтов (человек в веб среде)

Общие требования:

- наименование разрабатываемой системы:
- разработка формул и алгоритмов для прогнозирования поведения людей в веб среде.
- цель разработки:
- реализовать веб сайт с аналитикой посещения пользователей.
- прогнозировать по данным анализа поведение человека в веб среде.
- предлагаемые технологии для разработки программы (на выбор разработчика):
- Notepad ++;
- CSS script;
- HTML;
- Java script;
- Angular;
- Plesk.
- выбор архитектуры построения:
- клиент-Сервер.
- предлагаемые языки и технологии программирования:
- Android;
- PHP;
- Java;
- HTML;
- CSS;
- Javascript;
- C#.
- общий объем программной части системы, Мб
- не более 1000 Мб.

Технические требования:

- требования к программному обеспечению:
- быстрая скорость обработки входящей информации;
- Веб-приложение;
- код программы должен быть читабельным и понятным.
- тестирование и отладка программного обеспечения:
- проверка работоспособности программы;

Продолжение приложения А

- тестирование системы продаж на корректность;
 - тестирование платформы на появление ошибок и перегруженности;
- Экономические требования:
- расчет стоимости системы и стоимости разработки программного обеспечения (подлежит обсуждению):
 - стоимость готового продукта 5 904 404тг;
 - стоимость разработки 1 513 040тг.
 - потенциальные клиенты и области применения:
 - веб-сайты и приложения по всему миру;

Приложение Б (ЛИСТИНГ ПРОГРАММЫ)

```
<script>
var wixBiSession = {
initialTimestamp: Date.now()
, ssrRequestTimestamp: 1587984649529
, requestId: publicModel.requestId
, viewerSessionId: 'xxxxxxxx-xxxx-4xxx-yxxx-
xxxxxxxxxxxx'.replace(/[xy]/g,function(c){ var
r=Math.random()*16|0,v=c=='x'?r:(r&0x3|0x8);return v.toString(16);})
, sessionId: '6ff9eadc-4bf3-4f61-8c36-8a865778dbf8'
, initialRequestTimestamp: performance.timeOrigin ? performance.timeOrigin :
Date.now() - performance.now()

, is_rollout: 0
, is_platform_loaded: 1
, suppressbi: false
, dc: '84'
, renderType: 'bolt'
, siteRevision: '867'
, siteCacheRevision: '1585893427971'
, wixBoltExclusionReason: ""
, wixBoltExclusionReasonMoreInfo: ""
, checkVisibility: (function () {
var alwaysVisible = document.hidden !== true;
function checkVisibility() {
alwaysVisible = alwaysVisible && document.hidden !== true;
return alwaysVisible;
}
document.addEventListener('visibilitychange', checkVisibility, false);
return checkVisibility;
})();
, cacheCookie: document.cookie.match(/ssr-
caching="?cache[,#\s]*desc=(\w+)(?:[,#\s]*varnish=(\w+))?(?:[,#\s]*dc[,#\s]*desc=
(\w+))?(?:[";|$)/
, setCaching: function (bodyCacheable) {
if (!bodyCacheable) {
```

Продолжение приложения Б

```
wixBiSession.caching = undefined;
}

var parts = wixBiSession.cacheCookie;
if (parts) {
  set(parts);
}

if (window.PerformanceServerTiming) {
  var serverTiming = performance.getEntriesByType('navigation')[0].serverTiming;
  if (serverTiming && serverTiming.length) {
    var names = [, 'cache', 'varnish', 'dc'];
    parts = [];
    serverTiming.forEach(function (entry) {
      var i = names.indexOf(entry.name);
      if (i > 0) {
        parts[i] = entry.description;
      }
    });
    set(parts);
  }
}

if (!wixBiSession.caching) {
  wixBiSession.caching = 'none';
  wixBiSession.isCached = false;
}

function set(parts) {
  if (bodyCacheable && parts[1]) {
    wixBiSession.caching = parts[1] + ',' + (parts[2] || 'none');
    wixBiSession.isCached = isCached(parts[1]) || isCached(parts[2]);
  }
  if (parts[3]) {
    wixBiSession.microPop = parts[3];
  }
}

function isCached(part) {
```


Продолжение приложения Б

```
return !!part && part.indexOf('hit') === 0;
}
}
, sendBeacon: function (url) {
if (!wixBiSession.suppressbi) {

var sent = false;
try {
sent = navigator.sendBeacon(url);
} catch (e) {}
if (!sent) {
(new Image()).src = url;
}

}
}
, sendBeat: (function () {
var beatUrl = 'https://frog.wix.com/bt?src=29&evid=3'
+ '&v=1.5735.0'
+ '&msid=618e077d-c7dd-4c32-b032-963f172f5981'
+ '&isp=1'
+ '&st=2'
+ '&dc=84'
+ '&iss=1'
var referrer = document.referrer;
if (referrer) {
beatUrl += '&ref=' + encodeURIComponent(referrer);
}
var match = document.cookie.match(/_wixCIDX=(^[^;]*)/)
if (match) {
beatUrl += '&client_id=' + match[1];
}
var prevMark = 'fetchStart';
return function(et, name, extra, pageNumber) {
var tts = Math.round(performance.now());
var ts = et === 1 ? 0 : Date.now() - wixBiSession.initialTimestamp;
if (name && performance.mark) {
var mark = name + ' (beat ' + et + ')';
performance.mark(mark);

if (performance.measure) {
```

Продолжение приложения Б

```
performance.measure('\u2B50' + mark, prevMark, mark);
}
prevMark = mark;
}
extra = extra || "";
if (extra.indexOf('pn=') === -1) {
extra += '&pn=' + (pageNumber || '1');
}
if (extra.indexOf('sr=') === -1 && screen.width) {
extra += '&sr=' + screen.width + 'x' + screen.height;
}
if (screen.availWidth) {
extra += '&sar=' + screen.availWidth + 'x' + screen.availHeight;
}
if (extra.indexOf('wr=') === -1 && window.innerWidth) {
extra += '&wr=' + window.innerWidth + 'x' + window.innerHeight;
}
if (window.outerWidth) {
extra += '&wor=' + window.outerWidth + 'x' + window.outerHeight;
}
if (extra.indexOf('ita=') === -1) {
extra += '&ita=' + toBool(wixBiSession.checkVisibility());
}
if (wixBiSession.siteRevision || wixBiSession.siteCacheRevision) {
extra += '&siterev=' + wixBiSession.siteRevision + "-" +
wixBiSession.siteCacheRevision;
}
if (wixBiSession.hasOwnProperty('isUsingMesh')) {
extra += '&ism=' + toBool(wixBiSession.isUsingMesh);
}
if (wixBiSession.hasOwnProperty('caching')) {
extra += genField('caching') + '&is_cached=' + toBool(wixBiSession.isCached);
}
wixBiSession.sendBeacon(beatUrl
+ '&et=' + et
+ (name ? '&event_name=' + encodeURIComponent(name) : "")
+ '&ts=' + ts
+ '&tts=' + tts
+ '&vsi=' + wixBiSession.viewerSessionId + '&rid=' + wixBiSession.requestId
```

Продолжение приложения Б

```
+ '&viewer_name=' + encodeURIComponent(wixBiSession.renderType)
+ '&is_rollout=' + wixBiSession.is_rollout
+ '&is_platform_loaded=' + wixBiSession.is_platform_loaded
+ genField('sessionId')
+ genField('visitorId', 'vid')
+ genField('siteMemberId', 'mid')
+ '&url=' + encodeURIComponent(location.href.replace(/^http(s)?:\:\/\/(www\.)?/, ""))
+ extra
);
};
```

```
function genField(name, label) {
return wixBiSession[name] ? '&' + (label || name) + '=' + wixBiSession[name] : "";
}
function toBool(v) {
return v ? '1' : '0';
}
})();
};
wixBiSession.setCaching(true);
wixBiSession.sendBeat(1, 'Init');
</script>
<script type="text/javascript">
```

```
var serviceTopology = {"cacheKillerVersion":"1","staticServerUrl"
var isStreaming = true;
```

```
var isSAVRollout = "false";
var googleAnalytics = "";
var ipAnonymization = false;
```

```
var googleRemarketing = "";
var googleTagManager = "";
var facebookRemarketing = "";
var yandexMetrika = "";
</script>
```

```
<script type="text/javascript">
var santaBase = 'https://static.parastorage.com/services/santa/1.9812.0';
```

Продолжение приложения Б

```
var boltBase = 'https://static.parastorage.com/services/wix-bolt/1.5735.0';
var boltVersion = '1.5735.0';
</script>
<script>
var requirejs = {
onNodeCreated: function(node) {
var src = node.getAttribute('src');
var shouldIgnore = ['googletagmanager.com', 'google-analytics.com',
'googleadservices.com', 'doubleclick.net', 'connect.facebook.net'].some(function
(domain) {
return src.indexOf(domain) !== -1;
});
if (!shouldIgnore) {
node.setAttribute('crossorigin', 'anonymous')
}
}
}
</script>

<script>
window.messageBuffer = [];
window.messageHandler = function(event) { messageBuffer.push(event); };
window.addEventListener('message', window.messageHandler, false);
</script>
<title>YTO AGRO Казахстан | Купить трактор</title>

<meta name="description" content="Продажа тракторов, комбайнов, навесного
оборудования и семян в Казахстане! www.kupiltraktor.kz">

<link rel="canonical" href="https://www.kupiltraktor.kz/">

<meta property="og:title" content="YTO AGRO Казахстан | Купить трактор">

<meta property="og:description" content="Продажа тракторов, комбайнов,
навесного оборудования и семян в Казахстане! www.kupiltraktor.kz">

<meta property="og:image"
content="https://static.wixstatic.com/media/a5e984_705efa8b31d947529971536e88

Продолжение приложения Б
<meta property="og:image:width" content="3543">
```

```
<meta property="og:image:height" content="3543">
<meta property="og:url" content="https://www.kupiltraktor.kz">
<meta property="og:site_name" content="УТО AGRO Казахстан">
<meta property="og:type" content="website">
<meta name="keywords" content="купить трактор, купить трактора, трактор в
алматы">
<meta name="google-site-verification" content="-
xMbwWbqIf2RaKNxXyyAI25a2HDg8Geli0MJ0rYRkXA">
```

```
<script type="text/javascript" charset="utf-8" async="" data-requirecontext="_"
data-requiremodule="controller" src="/УТО AGRO Казахстан _ Купить
трактор_files/controller.min.js.Без названия"
crossorigin="anonymous"></script><script type="text/javascript" charset="utf-8"
async="" data-requirecontext="_" data-requiremodule="siteRegionContainer"
src="/УТО AGRO Казахстан _ Купить
трактор_files/siteRegionContainer.min.js.Без названия"
crossorigin="anonymous"></script><script type="text/javascript" charset="utf-8"
async="" data-requirecontext="_" data-requiremodule="linkBar" src="/УТО
AGRO Казахстан _ Купить трактор_files/linkBar.min.js.Без названия"
crossorigin="anonymous"></script><script type="text/javascript" charset="utf-8"
async="" data-requirecontext="_" data-requiremodule="verticalMenu" src="/УТО
AGRO Казахстан _ Купить трактор_files/verticalMenu.min.js.Без названия"
crossorigin="anonymous"></script><script type="text/javascript" charset="utf-8"
async="" data-requirecontext="_" data-requiremodule="stripColumnsContainer"
src="/УТО AGRO Казахстан _ Купить
```

Продолжение приложения Б

```
трактор_files/stripColumnsContainer.min.js.Без названия"
crossorigin="anonymous"></script><script type="text/javascript" charset="utf-8"
async="" data-requirecontext="_" data-requiremodule="mediaContainer"
src="/УТО AGRO Казахстан _ Купить трактор_files/mediaContainer.min.js.Без
названия" crossorigin="anonymous"></script><script type="text/javascript"
charset="utf-8" async="" data-requirecontext="_" data-requiremodule="promote-
analytics-adapter" src="/УТО AGRO Казахстан _ Купить трактор_files/analytics-
event-adapter.bundle.min.js.Без названия"
crossorigin="anonymous"></script></head>
```

```
<body class="" data-js-loaded="true" style="">
```

```
<script type="text/javascript">
var htmlClassList = document.documentElement.classList;
```

Продолжение приложения Б

```
var bodyCacheable = true;
wixBiSession.setCaching(bodyCacheable);

var clientSideRender = false;

</script>

<!--body start html embeds start-->

<!--body start html embeds end-->

<li>
<p class="font_8"><span style="font-family:play,sans-serif;">Трактора
класса&nbsp;0.6-0.9&nbsp;активно используются:</span></p>
</li>
<li>
<p class="font_8"><span style="font-family:play,sans-serif;">в
растениеводстве&nbsp;на&nbsp;площадях до 50 га.</span></p>
</li>
<li>
<p class="font_8"><span style="font-family:play,sans-serif;">в
животноводстве</span></p>
</li>
<li>
<p class="font_8"><span style="font-family:play,sans-serif;">на
коммунальных&nbsp;предприятиях</span></p>
</li>
<p class="font_8"><span style="font-family:play,sans-serif;">Трактора класса
1.4&nbsp;активно используются:</span></p>
</li>
<li>
<p class="font_8"><span style="font-family:play,sans-serif;">в
растениеводстве&nbsp;на&nbsp;площадях до 200&nbsp;га.</span></p>
</li>
<li>
<p class="font_8"><span style="font-family:play,sans-serif;">в
животноводстве</span></p>
</li>
<li>
<p class="font_8"><span style="font-family:play,sans-serif;">на
коммунальных&nbsp;предприятиях</span></p>
```

Продолжение приложения Б

`<p class="font_8">Трактора класса 2 активно используются:</p>`

``

``

`<p class="font_8">в растениеводстве на площадях до 350 га.</p>`

``

``

`<p class="font_8">в животноводстве</p>`

``

``

`<p class="font_8">на коммунальных предприятиях</p>`

``

`<p class="font_8">Трактора класса 4 активно используются:</p>`

``

``

`<p class="font_8">в растениеводстве на площадях до 500 га.</p>`

``

``

Продолжение приложения Б

`<p class="font_8">в животноводстве</p>`

``

``

```
if (window.wixBiSession) {  
wixBiSession.isUsingMesh = true;  
if (wixBiSession.sendBeat) {  
wixBiSession.sendBeat(12, 'Partially visible', '&pid=th4j0');  
}
```

```
if (window.requestCloseWelcomeScreen) {  
window.requestCloseWelcomeScreen();  
}
```

```
}
```

</script></div></div>

```
private Result processFunction(String func, Result r)
{
if (func.equals("sin")) {
return new Result(Math.sin(Math.toRadians(r.acc)), r.rest);
} else if (func.equals("cos")) {
return new Result(Math.cos(Math.toRadians(r.acc)), r.rest);
} else if (func.equals("tan")) {
return new Result(Math.tan(Math.toRadians(r.acc)), r.rest);
} else {
System.err.println("function '" + func + "' is not defined");
}
return r;
}
} String[] formulas = new String[] { "L(i)", "r&&r*2", "W(u)(90)+4-cos(0)", "2--4",
"2**3*5-----7", "3.5.6-2" };
MatchParser p = new MatchParser();
p.setVariable("X", 2.0 );
for( int i = 0; i < formulas.length; i++){
try{
System.out.println( formulas[i] + "=" + p.Parse( formulas[i] ) );
}catch(Exception e){
System.err.println( "Error while parsing '"+formulas[i]+'"' with message: " +
e.getMessage() );
}
}
```


Приложение В Акт внедрения

ТМ «Leopold», «ИП Макеева Ж.М.»
ИНН 940817400816
Республика Казахстан, г.Алматы,
Ул. Бухар жырау 35
leopoldkz@vandex.kz
+7 707 131 75 69



Утверждаю
Директор ИП «Макеева Ж.М.»
Макеева Жазира Мухтарқызы
«20» мая 2020г.

АКТ ВНЕДРЕНИЯ

Настоящий акт составлен о том что, дипломная работа студента 4 курса университета НАО «АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ ИМЕНИ ГУМАРБЕКА ДАУКЕЕВА» группы ВТ-16-2 очной формы обучения Макеев Д.М. на тему «Разработка системы анализа поведения посетителей веб сайтов (человек в веб среде)» была внедрена в ИП «Макеева Ж.М.» с торговой маркой «LEOPOLD» и используется для рекламной компании в Digital Marketing и SMM. Данная работа обеспечивает прогнозирование поведения людей в веб-среде и выявление более эффективных рекламных методов размещения рекламы в интернете.

Директор
ИП «Макеева Ж.М.» ТМ «LEOPOLD»

Макеева Ж.М.

Подпись _____

