

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

кафедра Компьютерных технологий

«Допущен к защите»
Заведующий кафедрой _____

(Ф.И.О., ученая степень, звание)

« _____ » 20__ г.
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Разработка сайта на технологии Ruby on Rails

Специальность БССТ-0403 - Информационные технологии и программное обеспечение

Выполнил (а) Шименов Д.К. BT-12-2
(Фамилия и инициалы) группа

Научный руководитель Байжанова Д.О., ст.преп.
(Фамилия и инициалы, ученая степень, звание)

Консультанты:

по экономической части:

Бекмурзаева А.Ч., к.э.н., доцент
(Фамилия и инициалы, ученая степень, звание)
А.Ч. « 18 » 05 2016 г.
(подпись)

по безопасности жизнедеятельности:

Тришарова Н.Г., Д.т.н., проф.
(Фамилия и инициалы, ученая степень, звание)
Н.Г. « 23 » 05 2016 г.
(подпись)

по применению вычислительной техники:

Байжанова Д.О., ст.преп.
(Фамилия и инициалы, ученая степень, звание)
Д.О. « 12 » 05 2016 г.
(подпись)

Нормоконтролер: Байжанова Д.О., ст.преп.

(Фамилия и инициалы, ученая степень, звание)

Д.О. « 26 » 05 2016 г.
(подпись)

Рецензент:

(Фамилия и инициалы, ученая степень, звание)

« _____ » 20__ г.
(подпись)

Алматы 2016 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Факультет Аэрокосмических и информационных технологий
Специальность 58 07 01 00 - Вычислительная техника и программирование
Кафедра Компьютерных технологий

ЗАДАНИЕ

на выполнение дипломного проекта

Студент Ахметов Дидар Каиратович
(фамилия, имя, отчество)

Тема проекта Разработка сайта на технологии Ruby on Rails

утверждена приказом ректора № 148 от «19» октября 2015 г.

Срок сдачи законченной работы «01» 06 2016 г.

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

- 1) Темы и история развития интернет трансляции
- 2) Разработать удобный сайт для файла видео формата Флеш - видео обзор и аналитических программ
- 3) Изучить сайты последние обзоры сайтов
- 4) Разработать удобный интерфейс для анализа видео

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

- 1) Анализ предметной области
- 2) Темы и история развития интернет трансляции
- 3) Средства и инструментальная разработка сайта
- 4) Выбор программирования Ruby
- 5) Реализация и разработка сайта
- 6) Структурное описание системы

Перечень графического материала (с точным указанием обязательных чертежей)

Диаграмма вариантов использования

Рекомендуемая основная литература

- 1) Замков Дм. Но Т. Код на языке Java - М.: ВМ, 2014, - 456 стр
- 2) Дэйв Макратонд. Базовое книга CSS3 - М.: Наука, 2009, - 986, стр
- 3) Дэйв Рэндалл. JavaScript. Подробное руководство, - М.: Наука 2014 - 640 стр

Консультанты по проекту с указанием относящихся к ним разделов

Раздел	Консультант	Сроки	Подпись
Экран. часть	Рехимова А.В.	17.03 - 27.05.16	А.В.
Б.М.Д	Трихачева И.Г.	18.03 - 23.05.16	И.Г.
Кодовая часть	Байтимова Д.О.	01.04 - 05.05.16	Д.О.
Примечательная часть	Байтимова Д.О.	04.04 - 06.05.16	Д.О.

Аннотация

Данная дипломная работа посвящена созданию сайта на фреймворке Ruby on Rails для транслирование видео-обзоров и прямых трансляций футбольных матчей. Фронтэнд сайта был написан на языках разметки HTML, JavaScript и CSS, также под управлением Apache сервера.

Дипломная работа сочетает в себе введение, пять разделов и заключение. В введение было описан весь смысл создание сайта, актуальность его в рынке. Пять разделов описывают инструменты разработки, средства разработки, их плюсы и минусы, полное описание внешнего вида самого сайта, его весь функционал. В заключение было описано сделанные выводы после проделанной работы. В разделах экономики была описана полная описание её стоимость на рынке и была высчитана сумма на разработку сайта от и до. В безопасности жизнедеятельности было выяснено то, что может пагубно отразиться на здоровье человека. Это также конденсирование воздуха и свежий воздух в рабочем помещении. Также были выявлены что на человека пагубно может сказаться освещение в рабочей комнате. Все эти недочеты были выявлены и устранены.

Андатпа

Бұл дипломдық жұмыс Ruby on Rails фреймворк технологиясымен футболдық матчтардың бейнешолуын және тікелей трансляциясын тамашалауға арналған сайт. Сайттың фронтэнді Apache веб-сервері басқарумен HTML, Javascript, CSS тілдерінде жазылған.

Дипломдық жұмыс кіріспеден, бес бөлімнен және қорытындыдан тұрады. Кіріспеде сайт құрудың толық сипаттамасы, оның еңбек нарығындағы өзектілігі сипатталған. Бес бөлімде сайт жасаудың құрал-саймандары мен жабдықтары, олардың артықшылықтары мен кемшіліктері, сайттың фронталды келбетінің толық сипаттамасы және функционалдығы жазылған. Қорытындыда бүкіл жасалынған жұмыстың нәтижесі келтірілген. Экономикалық бөлімде сайт құрудың басынан бастап соңына дейінгі сомасы есептелген және оның еңбек нарығындағы құны толық сипатталған. Өміртіршілік қауіпсіздігі бөлімінде адам денсаулығына кері әсер тигізетін параметрлер анықталған. Ол жұмыс бөлмесіндегі таза ауаның және ауа айналымының жеткіліксіздігі болып табылады. Сонымен қатар, адам денсаулығына жұмыс бөлмесіндегі жарықтандыру да зиянды әсерін тигізуі мүмкін. Бұл барлық кемшіліктер қарастырылып жойылды.

Annotation

This thesis is devoted to the creation of a site on Ruby on Rails. As the theme for the development I have chosen the site for broadcasting video reviews and live broadcasts of football matches. Frontend site was written in markup languages HTML, Javascript, and CSS, also under the control of the Apache server.

Graduation project combines an introduction, five chapters and a conclusion. The introduction described the whole point of creating the site, its relevance in the marketplace. Five sections describe the development tools, development tools, their pros and cons, a full description of the appearance of the site, its full functionality. In conclusion, the findings have been described after the work done. The sections of the economy has been described a complete description of its value on the market and has been calculated on the amount of development of the site and out. The life safety, it was found that that may adversely affect human health. It is also a condensation of air and fresh air in the work. It identified that the person may be detrimental to the lighting in the work room. All these shortcomings have been identified and eliminated.

Содержание

Введение	5
1 Анализ предметной области	7
1.1 Понятие и история развитие интернет трансляции	7
1.2 Ruby on Rails как веб-ориентированная среда разработки	11
1.3 Преимущество Ruby on Rails	12
1.4 Удобные инструменты разработки	14
1.5 Архитектура Model-View-Controller	12
1.6 Архитектура приложений Ruby on Rails	14
1.7 Соглашение по конфигурации (Convention over Configuration)	16
1.8 Схема миграций (Schema Migration)	18
2 Среда и инструменты разработки сайта	20
2.1 Текстовый редактор Notepad++	21
2.2 Web сервер Apache	21
2.3 Язык гипертекстовой разметки	23
2.4 Язык программирование JavaScript	25
2.5 Cascading style sheets	26
2.6 Язык программирование Ruby	27
3 Реализация и разработка сайта	30
3.1 Реализация и разработка сайта	35
3.2 Структурная схема системы	37
3.3 Описание программы	40
4 Экономическое обоснование проекта	50
4.1 Трудовые ресурсы, используемые в работе	50
4.2 Оборудование, используемое в работе	51
4.3 Программное обеспечение, используемое в работе	52
4.4 Сроки реализации проекта	55
4.5 Расчет стоимости работы по разработке	58
4.6 Расчет затрат на оплату труда	60
4.7 Цена интеллектуального труда	61
5 Безопасность жизнедеятельности	65
5.1 Анализ условий труда	68
5.2 Расчет искусственного освещения	70
5.3 Разработка вопросов пожарной безопасности	71
5.4 Расчет системы кондиционирования	73
5.5 Аналитический расчет искусственного освещения	74
Заключение	76
Список литературы	77
Диаграмма Приложение А	78

Введение

Интернет-видео - система, основанная на двусторонней передаче цифрового телевизионного сигнала через подключение к Интернету через широкополосное соединение. Интернет-видео достигла нового уровня развития. Если она казалась простой жителям трюком, сейчас интернет-видео становится колоссальным масштабом до недавнего времени. Члены приманить большое разнообразие возможностей применения телевизионного контента, которые до недавнего времени, пока не осуществимы на традиционном телевидении. [1]

Веб-трансляция выигрывает войну за внимание с традиционными средствами массовой информации (эфирное, кабельное и спутниковое телевидение, радио, газет и журналов). Согласно исследованию эМаркетер (www.emarketer.com), проведенного в 2015 году в США 70 процентов пользователей смотреть видео онлайн, а 20 процентов делают это ежедневно. наземной телевизионной аудитории в Казахстане значительно снижается из-за Интернета. По мнению аналитиков, в 2018 году темпы роста в онлайн-видео было 84 процентов. Интернет-трансляция открытой, постоянно развивающейся сети, в которой многие малые и средние производители предлагают контент автор видео. Это мир многостороннего и разнообразного видео, созданных независимых студий, групп и отдельных лиц во всех уголках земного шара. В ходе этой части дипломного проекта была достигнута основная цель работы - анализ условий труда. Мы также знаем, что. В комнате, в дополнение к оборудованию и рабочих мест могут быть различные обогреватели и кондиционеры [2].

Разработка веб-сайта представляет собой сложный и трудоемкий процесс, требующий больших затрат умственного труда. Стоимость разработки включает в себя следующие категории расходов: вычеты из заработной платы на социальные нужды, амортизация, затраты на электроэнергию.

Наибольший удельный вес в общей стоимости разработки программного обеспечения, затрат на рабочую силу в размере 124,370.4 тенге, что составляет 23% от разработки проекта. [18]

Благодаря интеграции Интернет - технологий и архитектуры реализации операции клиент-сервер и обслуживанию информационных систем значительно упрощается, сохраняя при этом относительно высокую эффективность и простоту использования общей информации. Актуальность данной темы является то, что видео-обзоры на сегодняшний день очень популярны среди любителей футбола. Футбол игра миллионов, и как факт, что они вложили много денег. Целью данного проекта является разработка это место для удобного просмотра программ обзора и анализа

видео футбол. Получите последние обзоры игр, разработать удобный интерфейс для пользователей.

Разработка и проектирование веб-сайта производится с использованием компьютерных технологий. В этой комнате работает веб-разработчик, Attendant и системного администратора.

При разработке программного обеспечения играет важную роль в правильной организации условий труда в рабочей зоне.

Так как эта работа связана с продолжительностью пребывания на компьютере, необходимо учитывать скорость искусственного света, анализ пожарной безопасности и кондиционирования воздуха для обеспечения благоприятных условий работы.

Этот сайт будет включать множество игр, таких как клубных соревнованиях и международных турнирах.

Сегодня, онлайн-трансляция быстро развивается. Существует интернет-бизнеса онлайн трансляция спортивных мероприятий и соревнований, особенно футбол, командные виды спорта, хоккей и другие известные и динамичных игр, любимых многими. Транслировать абсолютно любой вид спорта, главное, чтобы иметь возможность получить свою прибыль от него. Интернет бизнес имеет ряд преимуществ по сравнению с оффлайн бизнеса. Итак, что это заработок на основе. Например, пользователи некоторых стран не имеют возможность смотреть спортивные состязания онлайн-трансляции из-за блокирования видео в их регионе. Вещание на вашем матче сайте вы можете не только увеличить трафик на вашей жизни, но вы можете организовать трансляцию видео на других сайтах за отдельную плату.

1 Анализ предметной области

1.1 Понятие и история развитие интернет трансляции

Улучшение Интернет увеличило возможность работы с удаленными блоками связи помощь и поддержку для всех видов проблем через Интернет и т.д. Структура акцента информационных систем могут быть выполнены по отношению к 2-х независимых компонентов. Вычислительная структура, которая представляет собой группу телекоммуникационной сети программного обеспечения и информацию об организации. Связанные функциональные подсистемы, которые обеспечивают решение проблем и достижение. Возможности Интернета как инструмента и в качестве медиа-среды часто вводят в заблуждение, не только ученые, но и пользователи. Открытая среда в то же время Интернет стал доступен и одним из основных глобальных вещателей, творческие и обычные пользователи. И те, и другие были заинтересованы в создании своего собственного уникального телевизора, используя Интернет. Различные возможности привели к различным подходам к вещанию в Интернете. Интернет-вещание - это открытая, постоянно развивается сеть, в которой многие малые и средние производители предлагают контента автор видео. Это мир разностороннего и разнообразного видео, созданных независимых студий, групп и отдельных лиц во всех уголках земного шара. Интернет-ТВ, по определению, предоставляет широкий спектр медиа-продуктов, программ и передач, чем зрители привыкли видеть на обычном телевизоре. Интернет-телевидение использует открытые стандарты и форматы, которые позволяют каждому пользователю стать вещателем. Открытая платформа гарантирует, наряду с эфирными каналами, практически неограниченное количество независимых авторских прав и интернет-каналов. [3] Вещателем напрямую получает прямой канал связи с аудиторией, независимо от поставщика услуг Интернета или кабельного оператора. Любое техническое решение Интернет-ТВ является наиболее независимым от пользовательского оборудования. Трансляция интернет-телевидение трансграничный, предназначена для глобальной аудитории, если это совместимо с распределением контента. Интернет-телевидение модель ориентирована не на массовую аудиторию и бесконечным числом каналов, как можно ближе к интересам целевой аудитории. В результате сегментации интернет-вещания, что приводит к появлению узкоспециализированных каналов, таких как канал для адвокатов, занимающихся исключительно в арбитражных делах. Таким образом, Интернет открыл DanceJam сайт, созданный специально для танцоров, которые могут выложить свои танцевальные видео, сгруппированные по категориям. Футбольный клуб Chelsea предоставляет услуги подключения к

своей собственной системе широкополосного вещания. В дополнение к онлайн-трансляции, абоненты будут первыми получать новости клуба и будет иметь возможность вступать в переписку по электронной почте для игроков своей любимой команды. Разделяя взгляды одного из идеологов Web 2.0, Тим О'Рейли, развитие интернет-телевидения можно условно разделить на два периода - до того 2000-2001 годы и с 2001 года (принята вещателям инструменты Web 2.0). Если первая фаза характеризуется экспериментируя с видеотрансляции, а также попытки крупных корпораций использовать Интернет для вещания, то второй этап интернет-телевидения переживает второе рождение в результате социализации, появление видео общин, а также как все возрастающей доступности видео оборудования, таких как телефонные конвергенции и видеокамерах [4]. На первом этапе вещания участков процесса была серия экспериментов. Первое упоминание о тест веб-трансляции видео в Соединенных Штатах содержатся в хронологию Интернета и от ноября 1992 года Одним из первых опытов интернет-вещания была проведена студентами Университета штата Мэн факультета журналистики в ЭКСПО- 1998 выставка в Ганновере, США. Организованный студентами просматриваемой программы около 16 тысяч человек. Каждый день в эфире выходит через 5 программ. Двое из них были 10 минут новостные блоки, другие были полностью посвящены анализу воздействия компании. В сентябре 1999 года он был создан интернет-проект специально для вещания в сети глобальной сети мира несколько концертов звезд эстрады. В 1999 году канадский интернет-проект начал 17 США вещания телеканалов в сети Интернет. Руководство не занимается какой-либо производства или покупки видео контента. По большому счету, бизнес-модель была очень близка к бизнес-модели тела подземных сетей. В феврале 2000 года руководство было вынуждено закрыть сайт, чтобы не нарушать существующего канадского законодательства. В параллельных экспериментах с интернет-вещания стала представлять и коммерческий интерес. В 1998 году американский миллиардер Марк Кубан начал мультимедийный интернет-проект Broadcast.com, который разместил видео и аудио файлы, с особым акцентом на спортивных мероприятиях. Сайт предлагает посетителям Broadcast.com единый портал для доступа к последним новостям, спортивных матчей и судебных разбирательств. В то же время компания продает рекламу, привлекая рекламодателей, что сайт посещают умных, образованных и богатых людей. Стоит отметить, что в 1998 году, подключение к Интернету достаточно для просмотра видео в Интернете, могли позволить себе только состоятельные люди. Broadcast.com компания позволяет получить через интернет-программы 500 радио- и 65 телевизионных станций. В 1999 году владелец компании Марк продал. Канал и был типичным примером финансового канала, ведущего трансляции в режиме реального времени. Государственный канал работает профессиональных ведущих, операторов, журналистов и других экспертов. Редакционная группа компаний, которые участвовали в проекте Internet. В

июне 2002 года канал был закрыт. Канал при поддержке рекламодателей и крупных финансовых вложений. Тем не менее, в момент выхода из интернет-рынка крупные игроки (доткомов) финансовые потоки иссякли. Одним из основных принципов, которые по сути превратили идею распространения информации, это возможность социализации, создавая личный кабинет - блог или видео-блог, который может быть использован для распространения текстов, фото и видео. Здесь каждый может почувствовать свою уникальность, создать свой собственный видеоканал, чтобы получить обратную связь от читателей и зрителей. Основные принципы. Концепция длинного хвоста, авторство которого принадлежит главному редактору журнала Wired Крис Андерсон. Он заметил, что нишевые продукты могут принести такую же или большую прибыль, чем наиболее продаваемых продуктов. Новая технология распространения информации, в том числе видео стал веб-синдикации - одновременная публикация одного и того же материала на различных страницах или веб-сайтов. Для этого, RSS-лент, содержащих заголовки текстовых материалов, аудио или видео. Изначально эта технология использовалась на новостных ресурсах и блогах, но постепенно расширил сферу применения и в настоящее время широко используется для распространения звука. Большое количество контента, созданного самими пользователями, бесплатно, но с удовольствием (например, YouTube Интернет как платформа позволяет хранить неограниченное количество информации удаленно и использовать веб-приложения, заменив большую часть платного программного обеспечения (например, , docs.google.com) открытый веб--Resource руководствоваться коллективным разумом, а не на экспертной оценке (например, wikimedia.org); Развитие блогосферы позволило журналистам, чтобы избежать самоцензуры и гражданской журналистики помогли простым пользователям быстро сосредоточиться на текущих социально-политических вопросов (например, ohmynews.com). 1995 Японский разработчик свободного программного обеспечения Юкиhiro Мацумото (ака Matz) придумал свой собственный язык программирования Ruby, и написал интерпретатор для него. В одном из интервью, Мац сказал: "Мне нужен был язык сценариев более мощный, чем Perl, и более объектно-ориентированный, чем Python. Поэтому он решил создать свой собственный язык." Название Рубин указывает ссылку на Perl языке, из которого многие из особенностей синтаксиса и семантики были взяты (английский жемчуг -. Рубин - Перл, рубин). Реализация Межплатформенное интерпретатора языка распространяется с открытым исходным кодом, вы можете копировать и изменять. Последняя на момент написания этой статьи, версия 1.8.5 опубликована 28 августа 2006 г. Рубин - интерпретируемый язык высокого уровня для быстрого и удобного объектно-ориентированного программирования. Он имеет операционной системы независимой реализации многопоточности, сильный динамический ввод, "сборщик мусора" и многие другие функции. Рубин аналогичен по своим

характеристикам синтаксиса Perl и Eiffel, объектно-ориентированный подход - к Smalltalk. Некоторые особенности языка взяты из Python, LISP, Дилана и CLU. Рубин на Rails (RoR или просто Rails) - это «37signals» компания разработала сотрудник Наунмайером Дэвид Ханссон (David Heinemeier Ханссон) среду для создания веб-приложений. Rails написан на Ruby, и должна быть архитектура MVC (Model-View-Controller, Model-View-Controller). Одним из главных преимуществ является то, что эта среда не разработан с нуля, и был изолирован от успешно работает веб-приложения Basecamp (<http://basecamp.com>) - программное обеспечение для управления проектами. Первая версия Rails появилась в июле 2004 года, Среда уже опробован на текущем проекте, и не требует длительного и комплексного тестирования, как Рубин на рельсы изначально была частью известной операционной системы. Оставалось только адаптироваться к самостоятельной жизни. Rails - открытую среду, которая была ключом к его постоянному совершенствованию и быстрому распространению. Рубин на Rails (RoR) - рамки, написанный на языке программирования Ruby, который реализует архитектурный шаблон Model-View-Controller для веб-приложений, а также обеспечивает интеграцию с веб-сервером и сервером базы данных. Это программное обеспечение с открытым исходным кодом и распространяется под лицензией MIT. Созданный Дэвидом Ханссон Неунемеуером на основе его работы в компании 37signals инструмента управления проектами Basecamp, и выпущен в июле 2004 года 23 декабря 2008 года команда проекта Merb объединились с командой Rails для создания следующей версии Rails 3, которая будет сочетать в себе лучшее особенности обеих структур. [5]

На основе этих принципов разработки приложений:

- Для того, чтобы свести к минимуму дублирование кода в приложениях (принцип Не повторяй себя);
- По умолчанию, соглашение о конфигурации, типичной для большинства приложений (принцип Конвенции по конфигурации) - четкое описание конфигурации требуется только в исключительных случаях.

1.2 Ruby on Rails как веб-ориентированная среда разработки

Разработка веб-приложений требует тщательного изучения существующих концепций сегодня для решения проблем. Неотъемлемая часть успешного проекта является анализ существующих подходов, строительство новых и выбор оптимальных решений. Важную роль в создании веб-приложения является выбор рамок. Рубин на Rails является одной из тех сред разработки, которые стали выбирать для создания широкого спектра так называемых web2.0 приложений.

Рубин на Rails (RoR или просто Rails) - мощная программная среда для разработки и развертывания служб веб-приложений, которые способны к интеграции с Web-сервером и сервером базы данных. Эта структура представляет собой архитектурный шаблон Модель -View- контроллер (Model-View-Controller). Динамический AJAX-интерфейс, обработка запросов и выдача данных в контроллер, предметной области, отражены в базе данных - для всего из Rails предоставляет однородную среду разработки на языке Ruby.

Рубин - интерпретируемый язык высокого уровня, который поддерживает несколько парадигм программирования, в первую очередь объект. В 1995 году японский разработчик свободного программного обеспечения Юкихиро Мацумото пришел со своим собственным языком программирования Ruby, и написал интерпретатор для него. Этот язык программирования имеет независимо от реализации операционной системы многопоточности, сильной динамической типизации, "сборщика мусора", а также многие другие функции. Особенности синтаксиса Ruby, похож на Perl и Eiffel языке, некоторые особенности языка взяты из Python. В объектно-ориентированном подходе, она близка к Smalltalk. Рубин на Rails рамках был разработан Дэвидом Хаунтауэгом Ханссон. Rails появился в версии первого года в июле 2004 года. Rails представляет собой программное обеспечение с открытым исходным кодом, и это вызвало ее быстрое расширение и постоянное совершенствование.

Web-серверы.

Рубин на Rails может работать с любым сервером, который поддерживает технологию FastCGI, например, Apache, Lighttpd, SCGI. Во время разработки, чаще всего самый простой способ заключается в использовании WEBrick - Web-сервер, который поставляется с [6] Ruby.

База данных. Сервер баз данных поддерживает Ruby On Rails MySQL, Firebird, PostgreSQL, IBM DB2, Oracle, Microsoft SQL Server, встроенные, SQLite базы данных.

1.3 Преимущество Ruby on Rails

Основное преимущество языка программирования Ruby, и Ruby On Rails рамки считается скорость развития. Практика показывает, что темпы развития проектов RoR увеличилась на 30-40 процентов по сравнению с любым другим языком программирования или рамки. В первую очередь уровень развития роста определяется широким набором готовых к дому рабочие инструменты RoR, огромный набор готовых решений в сообществе, Ruby языка и простоты программирования на нем. Одним из наиболее важных частей культуры является социальной RoR. Я решил проблему - помочь решить другую. Я реализовал модуль - поделиться с сообществом. Таким образом, на данный момент в свободном доступе находятся тысячи готовых решений различных проблем. системы аутентификации,

авторизации, комментирующие, платежные системы, списки рассылки и многое другое (все, что обычно разрабатывается «с нуля»), реализованный реализован кем-то еще, апробированы и рекомендованы многочисленным сообществом. Рубин на рельсы - это основа. Зачастую структура не позволяет любителем. Конечно, Ruby на Rails, вы можете "изобретать колесо" и программирование во всех направлениях, не полагаясь на стандарты; но часто это не требуется. Стандарты размещения файлов в проекте, стандарты написания кода в проекте, общие правила программирования на Ruby On Rails структурирован много любого проекта. Благодаря этому проект становится читаемым. Ввод проекта для начинающих очень быстро. Опыт показывает, что любой новичок в проекте в первый день работы делает свои первые полезные изменения. В связи с этим, он не считается большой проблемой, если развитие проекта изначально возглавлял команду программистов, а также поддержку проекта или пересмотра - совершенно другой. Проект RoR на априорно понятной любому разработчику.

1.4 Удобные инструменты разработки

Тестирование.

При разработке любого крупного проекта возникает разумный вопрос. Как и кто будет тестировать проект? Это не всегда есть средства и желание создать целые отделы тестирования, к тому же я хочу, чтобы автоматизировать этот процесс. В отличие от других структур, в рамках RoR являются отличными инструментами для автоматизированного тестирования. В других языках программирования, базы, а также регулярные инструменты тестирования этого не делают. Конечно, есть развитие сторонних производителей, что позволяет организовать автоматическое тестирование проекта в РНР, но они не ставят "из коробки" и их использование программисты часто не понимают. Проект по Ruby On Rails, в идеале, код проекта не написан не до тех пор, пока под кодом не написано тесты. RoR идеология включает в себя оригинальное использование методов BDD (Поведение Driven Development) и TDD (Test Driven Development) [7].

Кэширование.

Кэширование проекты - один из наиболее важных этапов развития крупного интернет-проекта. В РНР существуют различные варианты для кэширования данных. Эти опции и инструменты привинчены, прилагаются, крепятся, прикрепите сторону. До сих пор, РНР сообщество нет единого мнения: лучше использовать лучший способ для кэширования данных, какие инструменты использовать.

Рубин на Rails в базовой конфигурации имеет регулярные средства кэширования данных. В начале доступных инструментов для реализации кэширования данных по проекту. Вы можете кэшировать целые страницы или блоки кода. Можно кэшировать результаты запросов и ActiveRecord-модели. Кэш может быть, как с помощью Memcached или Redis, а также другие средства. Для реализации кэширования на Ruby On Rails проекта вам в 95 процентах случаев ничего, кроме готовых и собственных решений не требуется.

Локализация.

Часто возникает ситуация, когда кто-то сделал проект, а потом вдруг понимает, что для того, чтобы продолжить развитие проекта требует английской версии. Разработчики на РНР, в то же время начать, чтобы начать говорить о том, что это не было предусмотрено заранее, это очень долгий и трудоемкий. Скажем, открыть параллельный проект, который будет полной копией его, и перевести его.

Рубин на рельсы в базе имеет средства локализации проекта. Вам может понадобиться, чтобы обеспечить поддержку различных языков на сайте на начальном этапе, так и в будущем. RoR способен распространять различные шаблоны для разных языков, он содержит конфигурационные файлы с переводами терминов, а также многих других установленных инструментов для локализации проекта.

Маршрутизация.

Часто во многих проектах РНР, мы можем увидеть картину, когда адрес конкретной страницы огромен и непонятен. Рубин на рельсы имеет гибкость полный рабочий день, чтобы настроить маршрутизацию, тип адреса, названия основных разделов. Можно быстро изменить адреса в одном месте без необходимости изменения адреса всего проекта. Сообщество-RoR разработчики ОСТАЛЬНЫЕ идеология активно используется. Адрес страницы в сети на Ruby On Rails проектов всегда ясно, красиво, хорошо понятно, поисковых систем, просто.

Проверка.

В Ruby On Rails прекрасно реализованы инструменты для проверки входящих данных. Ваши пользователи должны заполнять формы и проверить правильность ввода адреса электронной почты, пароль, или наличие необходимой минимальной длины имени пользователя - стандартные инструменты Rails поможет вам в этом.

Миграция и работа с базой данных. Многие бытовые проблемы на PHP проектов - невозможности очистить средства и инструменты управления ли структуру базы данных. Изменения в структуре часто сделаны вручную и непосредственно в базу данных. Из-за этого проекта существуют многочисленные странные поля и таблицы, о которых никто не может вспомнить ничего. В Ruby On Rails, есть полный рабочий день инструменты для работы с базами данных - «Миграция». Структура базы данных хранится в коде приложения и выполнен из проекта. Ваша структура будет всегда находиться в хранилище, то любые изменения в структуре будут задокументированы и привязаны к конкретной фиксации в хранилище.

Безопасность.

Рубин на Rails по умолчанию сильно обострены безопасности проекта. При использовании RoR инструментов исключены инъекции SQL и атаки XSS. Все входные параметры на экране по умолчанию. Выходные переменные в шаблонах также на экране только если вы не укажете опцию обратной связи. У деvelopepa нет никаких шансов, чтобы предотвратить ошибки безопасности (не без исключений, конечно).

1.5 Архитектура Model-View-Controller

Пример классической модели архитектуры MVC (Model-View-Controller или Модель-Представление-Контроллер) приведен на рисунке 1.1

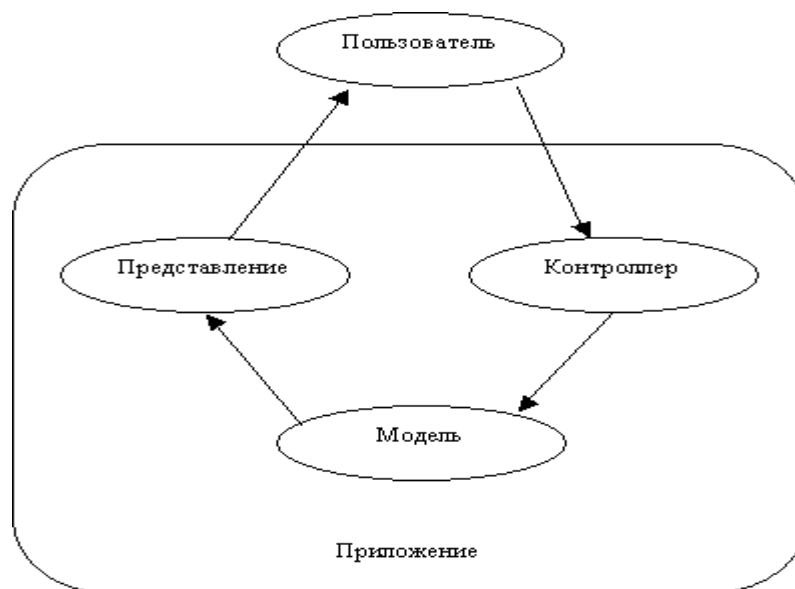


Рисунок 1.1 - Классическая модель трехзвенной архитектуры MVC (модель-представление-контроллер)

Эта архитектура характеризуется тремя основными элементами:

- Модель (Model) является ядром приложения. Модель отвечает за управление состоянием приложения. Логика приложения в модели выполняет две важные задачи: возвращает информацию о состоянии приложения, и изменяет состояние приложения. Представление (View) - средство для отображения информации, данные модели и ее состояние. Контроллер (Controller) -interpretiruet данные, введенные пользователем, и информирует модель и представление о необходимости соответствующей реакции.

1.6 Архитектура приложений Ruby on Rails

ActiveRecord и ActionPack сначала реализует презентационный слой, который является моделью и ActionPack сочетает в себе два других MVC-контроллера и просматривать на уровне архитектуры. Упрощенно MVC архитектуры в Ruby On Rails в целом можно описать следующим образом, на рисунке 2.

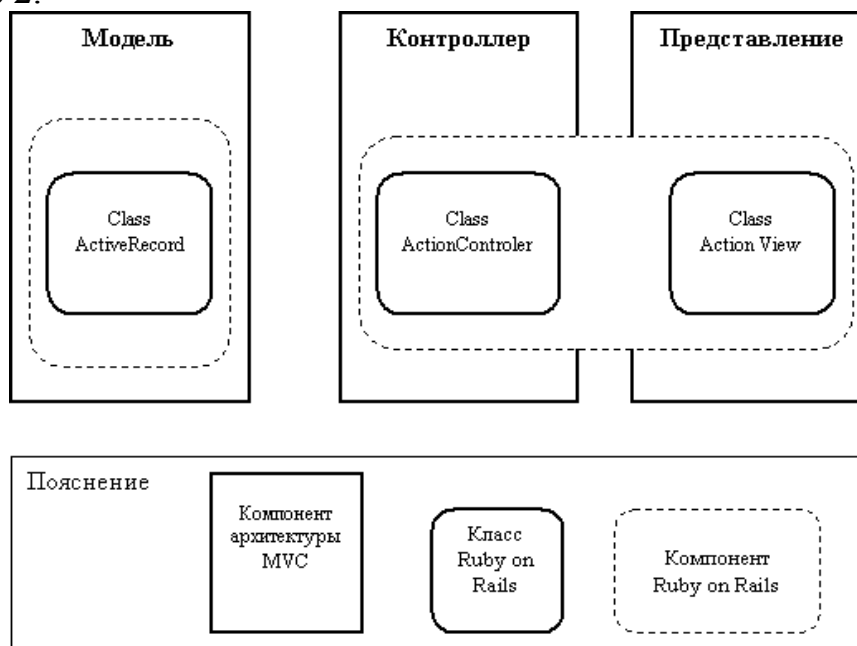


Рисунок 1.2 – Архитектура MVC в Ruby on Rails

ActiveRecord позволяет базу данных представление в виде объектов. Это создает модель предметной области, которая сочетает в себе самих данных, а также методов их контроля. Следуя стандартам ORM-модели (объектно-реляционного отображения и объектно-реляционные отображения), таблица появляется в классе строки таблицы - объекты, поля - атрибуты объекта. С силой динамической типизации в Ruby-разработчика языка, чтобы наследовать модель класса от базового класса ActiveRecord :: Base.

ActionPack содержит две части: ApplicationController, ответственные за реализацию бизнес-логики, и ActionView - механизм. ApplicationController-это контроллер в Rails-менеджер приложений. Он управляет логикой приложения программы, выступая в качестве связующего звена между данными и уровнем представления веб-браузера. Контроллеры в Ruby On Rails является класс, производный от ActionController :: Base. Открытые методы контроллера являются так называемые действия.

Действие контроллера отвечает за ряд задач:

- Решение о том, как обрабатывать конкретный запрос необходимо.
- Пример данных от модели к предложению передачи.
- Получение информации из запроса пользователя и использовать его для создания или изменения данных в модели.

ActionView (представительства).

В Rails обрабатывает представление обо всем, что отправляется в браузер. Представления - это шаблоны страниц с использованием RHTML (HTML со встроенным Ruby) или RXML (XML, генерируемой с использованием Ruby). Action View управляет отрисовкой вложенных и частичных шаблонов, и включает в себя поддержку AJAX.

Рубин на Rails определяет следующие принципы разработки приложений:

- Рубин на рельсы обеспечивает механизмы повторного использования, чтобы свести к минимуму дублирования кода в своих приложениях (принцип не повторяться).

- По умолчанию, соглашение о конфигурации, типичной для большинства областей применения (принцип Конвенции по конфигурации). Явная спецификация конфигурации требуется только в исключительных случаях.

DRY.

Если вам необходимо изменить поведение приложения, которое было построено на этом принципе, DRY, нет необходимости изменять код приложения в более чем одном месте. Например, вместо копирования и вставки кода аналогична или идентична функциональности, оно сконструировано таким образом, что функциональные возможности, хранится только один раз, в одном месте, а затем он делает ссылку на применение тех местах, где она необходима, Это также является примером

поддержки Ruby On Rails СУХОЙ принципе является тот факт, что, в отличие от Java, Rails 'не должны повторяться в определении схемы приложения схемы базы данных. Для Rails базы данных является надежным источником информации об источнике данных, и Rails собственные базы данных запрашивает информацию, которая может потребоваться для обеспечения правильной работы данных. DRY принцип сохраняется и, когда речь идет о реализации таких технологий, как Ajax (Asynchronous JavaScript и XML). Ajax-реализовать подход, который позволяет веб-приложение, чтобы заменить содержимое браузера пользователя динамически или обмениваться данными с формой сервера без перезагрузки страницы. При создании разработчиков Ajax-приложения часто приходится дублировать код, так как сайт должен функционировать браузеры, которые не поддерживают Ajax, а также те, которые поддерживают эту технологию. Таким образом, код, необходимый для отображения результатов в обоих типах браузеров, в основном идентичными. рамки Rails легче работать с различными поколениями браузеров без дублирования кода. [8].

1.7 Соглашение по конфигурации (Convention over Configuration)

Понятие "Соглашение Конфигурация" относится к тому, что Rails предоставляет набор настроек по умолчанию для процесса построения типичных веб-приложений. Некоторые соглашения относятся к архитектуре приложения, другие соглашения об именах. Используемые библиотеки или отдельные файлы Рубиновые загружаются автоматически, не требуя ни строки кода, если они находятся в нужном месте. Информация о конфигурации обычно хранится в XML-файлах (Extensible Markup Language), и эти файлы могут быть большими и трудно поддерживать. Во многих случаях мы должны пройти через длительный процесс конфигурации, прежде чем приступить к созданию по меньшей мере, простое приложение. Вместо того, чтобы в XML, приложение Rails использует несколько простых механизмов, которые позволяют узнать все через отражение и открытия. Например, Rails использует `refleksiya` для привязки таблиц базы данных к объектам Руби.

Рубин на Rails предоставляет ряд полезных функций, некоторые из которых перечислены ниже.

1.8 Схема миграций (Schema Migration)

В Rails, можно вносить изменения в структуру базы данных, используя так называемую миграцию. Кроме того, миграция позволяет структуру базы данных обратно к предыдущей версии.

Рейк инструмент сборки.

Грабли - инструмент для автоматизации сборки кода. Она имеет библиотеку основных задач, таких как функции для задач манипулирования файлами, а также библиотека для удаления скомпилированных файлов.

Поддержка для автоматизированного тестирования.

Рубин на Rails позволяет писать автоматизированные тесты для вашего приложения, при попытке реализовать его как можно проще. Rails включает в себя библиотеки для поддержки функциональной, блок, интеграцию и тестирование нагрузки.

Настраиваемый (нетто) URL.

Rails четко отделяет URL имен файлов, имен методов и других чувствительных внутренних частей вашего приложения. URL в Rails простые и понятные, а не длинные и загадочные.

Рамки Рубин на рельсы можно рассматривать как независимую платформу с полным набором необходимых модулей, документация, готовые решения и среды разработки высокого качества. С Ruby On Rails, вы можете создавать веб-приложения для различных направлений и сложности. В некоторых случаях его использование является приемлемым и оптимальным для выполнения этой задачи [9].

2 Среда и инструменты разработки сайта

2.1 Текстовый редактор Notepad++

Примечание ++ это бесплатный редактор исходного кода с поддержкой синтаксиса большого количества языков программирования. Программа предлагает широкий набор возможностей и имеет минимальное потребление ресурсов процессора.

Среди дополнительных функций Примечание ++ - дополнительное освещение текста и возможность сворачивания блоков, согласно синтаксису языка программирования. Пользователь может определить синтаксис языка программирования. Можно настроить режим подсветки. Доступные принципы выбора цвета и операторы языка программирования.

Примечание ++ обеспечивает возможность одновременного просмотра и редактирования нескольких документов. Вы также можете просматривать и редактировать `vdvuh` окна отображать один и тот же документ в разных местах. Изменение просмотра документов воды будет автоматически перемещен во второе окно просмотра (т.е.. Е. Вы редактируете документ, который имеет клона во втором окне просмотра).

Другие функции Примечание ++:

- Авто-завершение вводить слово.
- Возможность создать свой собственный список функций API (или скачать его со страницы загрузки).
- Поддержка регулярных выражений Поиска / Замены.
- Полная поддержка перетаскивания фрагментов текста.
- Динамическое изменение окон просмотра.
- Автоматическое определение состояния файла (уведомление об изменении или удалении файла другой программы - с возможностью перезагрузить файл или удалить его из программы).
- Увеличение и уменьшение (увеличение).
- Поддержка многих языков.
- Перечень программ можно печатать в цвете - как вы можете видеть на экране в окне редактирования.
- Замечания по этому документу.
- Выделение скобок при редактировании текста.
- Запись макроса и его выполнение [10].

2.2 Web сервер Apache

Apache является кросс-платформенный программное обеспечение, которое поддерживает операционную систему Linux, BSD, Mac OS, Microsoft Windows, Novell NetWare, BeOS.

Основные преимущества Apache считаются надежность и гибкость конфигурации. Это позволяет подключать внешние модули для предоставления данных, использовать базу данных для аутентификации пользователей, модифицировать сообщения об ошибках, и так далее. D.

Поддержка IPv6. Сервер был написан в начале 1995 года, и считается, что название восходит к комическим названием «пятнистый» (с англ. - "Zaplatochka"), так как он исключает ошибки, то популярный World Wide Web-сервера NCSA HTTPD 1.3. Позже, версия 2.x, сервер был переписан и теперь содержит NCSA код. На данный момент развитие осуществляется в филиале 2.2 и версии 1.3 и 2.0, только безопасность исправляет ошибки, допущенные. В настоящее время последняя версия 2.4 - 2.4.3 (21 августа 2012), для первой версии 1.3.42.

Веб-сервер Apache разрабатывается и поддерживается открытым сообществом разработчиков под эгидой Apache Software Foundation, и включается в состав многих программных продуктов, в том числе базы данных Oracle и IBM WebSphere.

Apache Ядро включает в себя основные функциональные возможности, такие как обработка конфигурационных файлов, протокол HTTP и модулей системной нагрузки. Ядро (в отличие от модулей) полностью разработан Apache Software Foundation, без участия сторонних программистов.

Теоретически, ядро Apache может функционировать в чистом виде, без использования модулей. Однако функциональные возможности этого решения является крайне ограниченным.

Apache Ядро полностью написан на C. Программирование

Система конфигурации Apache основана на текстовых конфигурационных файлах. Он имеет три уровня обычной конфигурации:

- Конфигурация сервера (httpd.conf).
- Конфигурация виртуального хоста (httpd.conf с версии 2.2, дополнительные / HTTPD-vhosts.conf).
- Уровень каталогов Configuration (.htaccess).

Она имеет свои собственные конфигурационные файлы языка, основанные на директивах блоков. Практически все параметры ядра не могут быть изменены с помощью файлов конфигурации, до управления MPM. Большинство модулей имеет свои собственные настройки.

Некоторые из модулей, используемых в работе файлов конфигурации операционной системы (например, / и т.д. / пароль и / и т.д. / хостов).

Кроме того, параметры могут быть установлены с помощью командной строки [11].

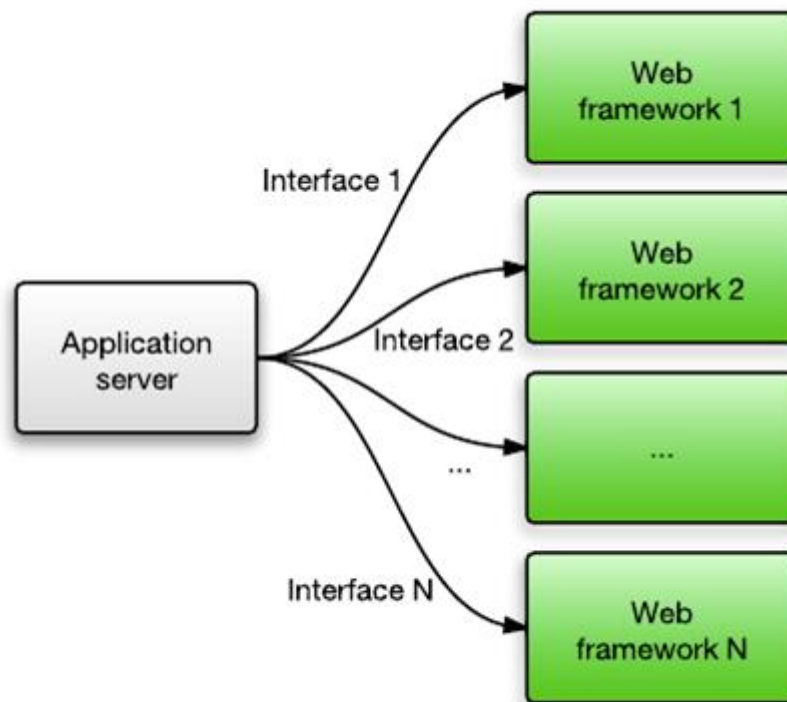


Рисунок 2.1 - Связь сервера и фреймворка

2.3 Язык гипертекстовой разметки

HTML (от английского HyperText Markup Language - «язык разметки гипертекста».) - Унифицированный язык разметки документов Всемирная паутина. Большинство веб-страниц содержат описание языка разметки HTML (или XHTML). HTML язык интерпретируется браузерами.. В результате чего от интерпретации форматированный текст отображается на экране компьютера или мобильного устройства.

Язык HTML является приложением SGML (стандартный обобщенный язык разметки) и соответствует международному стандарту ISO 8879.

Язык XHTML является строже версия HTML, то все ограничения XML и, на самом деле, XHTML можно рассматривать как применение XML в области гипертекстовой разметки.

Всемирная паутина HTML-страницы, как правило, передаются из браузера через HTTP или HTTPS сервер в виде простого текста или с помощью шифрования.

HTML был разработан британским ученым Тим Бернерс-Ли приблизительно в 1986-1991 годы в ЦЕРНе стен в Женеве, Швейцария. HTML создавался как язык для обмена научно-технических документов, пригодных для использования людьми, которые не являются экспертами в макете поля. HTML успешно справлялся с проблемой сложности SGML путем определения небольшого набора структурных и семантических элементов - дескрипторов. Дескрипторы также часто называют "тегов." С помощью HTML можно легко создать относительно простой, но красиво оформленный документ. Помимо упрощения структуры документа,

поддержка гипертекста HTML включен. Мультимедийные возможности были добавлены позже. [12]

Изначально язык HTML был задуман и создан как средство структурирования и форматирования документов без увязки их с воспроизведения медиа-файлов (дисплей). В идеале, текст содержания HTML был без стилистических и структурных искажений воспроизводиться на оборудовании с различной технической оснащенности (цветной экран современного компьютера, монохромный экран организатора, экран ограниченный размер мобильного телефона или устройства и воспроизведения голосового текста программного обеспечения), Тем не менее, текущее использование HTML очень далеко от своих первоначальных целей. Например, тег <таблица> используется для создания таблиц в документах, но часто используется для оформления и размещения элементов на странице. Со временем, основная идея HTML портативности была принесена в жертву потребностям современного мультимедиа и графического дизайна.

Плюсы сайты на HTML:

- HTML является первым языком веб-программирование. Именно на этой основе, многие команды строятся на PHP, JS и т.д. Таким образом, мы можем сказать, что она вечна. Если поиск по ряду признаков система может запретить любой движатель, HTML неприкасаемых в этом отношении.

- При написании сайта в чистом HTML на странице, нет ничего лишнего, она быстро загружается, хорошо доступны для поисковых систем, лучше оптимизированы.

- В HTML, вы можете думать через иерархию страниц, выберите основной и имеют добротню внутреннюю оптимизацию сайта.

2.4 Язык программирование JavaScript

JavaScript - Prototype-ориентированный язык программирования сценариев. Это реализация языка ECMAScript.

JavaScript обычно используется в качестве встроенного языка для программного доступа к объектам приложений. Наиболее широко используется браузеры, такие как язык сценариев для создания интерактивных веб-страниц.

Основные архитектурные особенности: динамическая типизация, слабая типизации, автоматическое управление памятью, основанный на прототипах программирования, функцию в качестве первого класса объектов.

На JS оказал влияние на многих языках, разработка была цель сделать язык, похожий на Java, но в то же время простой в использовании непрограммист. JS язык не принадлежит какой-либо компании или организации, которая отличает его от ряда языков программирования, используемых в веб-разработке.

В 1992 году Nombas (впоследствии приобрела Openwave) начала разработку встроенных сценариев СММ языка (С-минус-минус), который, в плане развития, должен был быть достаточно мощным, чтобы заменить макросы, сохраняя сходство с С для разработчиков не было легко для изучения. Основное отличие от моря была работа памяти. В новом языке все управление памятью осуществляется автоматически: не было никакой необходимости создавать буферы объявлять переменные, выполнять преобразование типов. Остальные языки сильно похожи друг на друга: в частности, Сmm поддерживает стандартные функции С, и операторы. Сmm был переименован ScriptEase, потому что первоначальное название звучало слишком негативным, и ссылки на него в С "пугали" людей. На основе этого собственный язык продукт СEnvі был создан. В конце ноября 1995 года Nombas разработал СEnvі версию введенной на веб-странице. Страницы, которые могут быть изменены с помощью языка сценариев, назывались Espresso Pages - они продемонстрировали использование языка сценариев для создания игр, проверки пользовательского ввода в форме и создавать анимацию. Эспрессо Страницы позиционируется как демо-версия, предназначенная, чтобы помочь себе представить, что произойдет, если браузер утвердит язык ШМ. Они работали только в 16-битный Netscape Navigator под управлением Windows [13].

Перед тем как Брендан Еуhom нанят компанией Netscape 4 апреля 1995 года, было поручено реализовать языковую programmirovaniyaScheme или что-то подобное в браузере Netscape. Как были размыты требования, Плач переданы группе, ответственной за серверные продукты, где он работал в течение месяца, занимаясь улучшением протокола HTTP. В мае девелопер был переведен обратно в команду, участие в стороне клиента (браузера), где он сразу же начал разрабатывать концепцию нового языка программирования. Управление развитием браузера, в том числе Том Пачино, Майкл Тоя, Рик Schell, был убежден, что Netscape должен поддерживать язык программирования, встроенный в страницы HTML-кода.

JS является объектно-ориентированный язык, но язык, используемый в прототипирования вызывает различия в работе с объектами по сравнению с традиционными классе-ориентированных языков программирования. Кроме того, JavaScript имеет ряд свойств, присущих функциональным языкам - функции, такие как объекты первого класса, объекты как списки, выделки, анонимные функции, затворы - что дает гибкость языка.

Несмотря на то, похож на синтаксис С, JavaScript имеет принципиальные отличия по сравнению с языком С:

- Объекты с возможностью интроспекции;
- Функция в качестве первого класса объектов;
- Автоматическое литье;
- Автоматический сбор мусора;
- Анонимные функции.

Язык отсутствуют такие полезные вещи, как:

- Модульная система: JavaScript не предоставляет возможности управлять зависимостями и сферы изоляции;
- Стандартная библиотека: в частности, нет никакого интерфейса прикладного программирования для работы с файловыми системами, контролировать входные и выходные потоки, основные типы двоичных данных;
- Стандартные интерфейсы для веб-серверов и баз данных;
- Система управления пакетами, которая будет контролироваться и автоматически устанавливается в соответствии с ними.

Выгоды:

- Максимально понятный для пользователя;
- Поддержка наиболее популярных браузеров "по умолчанию";
- Очень высокая скорость JavaScript;
- скрипты (программы, написанные на JavaScript) подключаются непосредственно к нагрузке и сразу запустить HTML-код веб-страниц;
- Программа может работать не только в браузере, но и на сервере.

Недостатки:

- Язык составляется в момент выполнения кода. Каждый раз, когда вы открываете сайт, JavaScript-код начинает собирать. По крайней мере, она увеличивается программы во время выполнения.
- Нет данных ввода. Проблема со всеми языками сценариев. В то время как выполнение кода достигает желаемых линий, работает ли он учиться. Но значительная часть, чтобы найти ошибки могут взять на себя компилятор, если бы я знал, что типы данных, с которыми он работает. А на скорости выполнения, код набирается быстрее.

2.5 Cascading style sheets

CSS формальный язык для описания внешнего вида документа, написанного с использованием языка разметки.

В основном используется как средство описания, появление веб-страниц, написанных на HTML и XHTML языков разметки, но также могут быть применены к любой из XML-документов, таких как SVG или XUL.

CSS используется создателями веб-страниц для определения цвета, шрифты, расположение отдельных блоков и других аспектов внешнего вида представления веб-страниц. Основной целью разработки CSS является логическим разделением описания структуры веб-страницы (которая сделана с помощью HTML или других языков разметки) от описания внешнего вида веб-страницы (которая в настоящее время производится с использованием CSS формальный язык). Такое разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его производительность, а также уменьшить сложность и повторяемость в структурном содержании. Кроме того, CSS позволяет представлять один и

тот же документ в различных стилях и методах вывода, таких как экранное представление, печатное представление, голос чтения (специальный голосовой браузер или чтения с экрана) или устройства вывода с помощью шрифта Брайля [14].

Перед тем как был CSS, дизайн веб-страницы могут быть реализованы непосредственно в содержимом документа, появление технологии CSS позволило разделить содержание и представления документа. Благодаря этой возможности было реально простое применение единого стиля дизайна для большинства этих документов, и это стало доступным быстрое изменение их конструкции. Использование CSS в веб-разработке имеет свои плюсы и минусы.

Преимущество:

- Сама по себе Easy язык CSS, наряду с принципом разделения дизайна от контента позволяет сократить время на разработку и поддержку.

- Есть несколько вариантов дизайна страниц, просмотренных на разных устройствах. Например, дизайн на экране компьютера предназначен для широкого, и будет полностью отображаться на экране, так и на мобильных устройствах будет адаптироваться к размеру экрана и некоторые элементы будут исключены из шоу, а также в печати, текст, который вы хотите напечатать, без ненужного (например, без меню колпачков).

- Сокращение времени загрузки веб-страниц на сайте в связи с передачей правил представления данных в отдельный CSS-файл. Этот браузер загружает только структуру документа, а также данные, хранящиеся на странице, а представление этих данных загружается браузером только один раз и может быть кэшируются Wat - Это уменьшает трафик времени загрузки и нагрузку на сервер.

- Простота конструктивных изменений. CSS-управляет отображением HTML-страниц. Когда возникает необходимость изменить дизайн сайта, нет необходимости редактировать каждую страницу. Для последующих изменений конструкции необходимо только изменить CSS-файл, и, как следствие, изменение конструкции быстрее.

- CSS предоставляет дополнительные параметры форматирования, которые не даже воображаемых, когда только сами атрибуты.

- Улучшена совместимость с различными платформами за счет использования веб-стандартов.

Недостатки:

- Различные виды отображения макета в разных браузерах. Если устареть браузеры, то возможно, что одни и те же данные CSS по-разному интерпретируются ими.

- Необходимость исправления не только один CSS-файл и HTML-теги. Это часто происходит на практике, необходимость исправить не только один CSS-файл и HTML-теги, которые связаны с CSS селекторов.

Иногда, это значительно увеличивает время редактирования, а также тестирование.

2.6 Язык программирование Ruby

Рубин-динамические, светоотражающие, высокоуровневый интерпретируемый язык программирования для быстрого и удобного объектно-ориентированного программирования. Язык имеет операционную систему независимой реализации многопоточности, сильный динамический ввод, сборщик мусора и многие другие функции. По особенностям он близок к синтаксису Perl и Eiffel, объектно-ориентированный подход - к Smalltalk. Кроме того, некоторые особенности языка взяты из Python, Лисп, Dylan и Clojure.

Реализация Межплатформенное интерпретатора языка является полностью бесплатным.

Язык следует принципу «наименьшего удивления»: Программа должна вести себя как программист ожидает. Тем не менее, в контексте Ruby, не менее удивлены, когда знакомство с языком, и в его тщательного изучения. Сам Мацумото утверждает, что целью развития является сведение к минимуму сюрпризов при программировании для него, но он был удивлен, узнав после распространения языка, что программисты, думая, что, кажется, и для многих из них принцип «наименьшего удивления» совпало с его принцип.

Рубин также унаследовал Perl язык программирования идеологию с точки зрения программиста, чтобы достичь того же результата несколькими различными способами. Люди разные, и это необходимо для свободы выбора. "Я предпочитаю, чтобы обеспечить много способов, если это возможно, но поощрять или направлять пользователей, чтобы выбрать лучший путь, если это возможно."

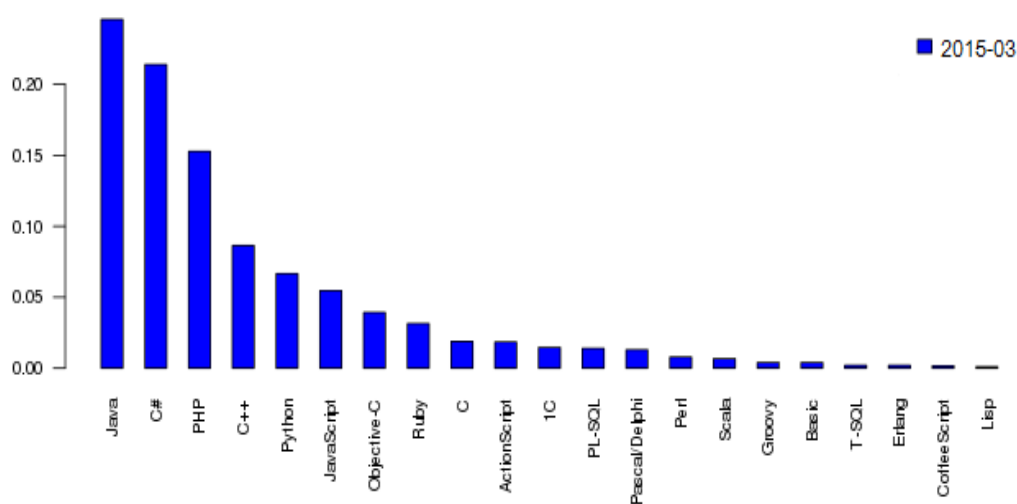
Одной из главных целей проекта было свободных программистов от рутинной работы, что калькулятор может работать быстрее и лучше. Особое внимание, в частности, было уделено рутинный упражнения (для обработки текста, управление), а также для их языка особенно хорошо настроены.

В противоположность этому, машинно-ориентированный язык, который работает быстрее, целью этого развития является язык, который ближе всего к человеку. Любая работа с компьютером выполняется и для людей, и вы должны заботиться прежде всего о людях, затраченные усилия. Язык

позволяет быстро и просто для человека, чтобы выполнить задачу, хотя, возможно, это потребует дополнительного времени работы компьютера.

Принципы языков программирования и устройства иногда отличает термин «Путь Руби» (англ. Руби Way). Хэл Фултон выделяет такие принципы, как "принцип наименьшего удивления простой, но не слишком просто", "" средней скорости работы программа, динамические, простые строгие правила, выполнение которых не достигает педантичность, необходимость создания полезной и красивой программы в качестве причины программирования. в общем, они не имеют точной формулировки и иногда этот термин используется для критики.

Какие языки часто пишутся. Данные из www.ethnologue.com



сайта.

Рисунок 2.2 – Рейтинг языков

Преимущество:

- Открытая разработка;
- Он работает на многих платформах;
- Может быть встроен в HTML-разметки;
- Относится к языкам программирования сверхвысокого уровня (VHLL), то есть, имеет высокий уровень абстракции и объективного подхода к реализации алгоритмов;
- Реализует концептуально чистый объектно-ориентированной парадигмы;
- Обеспечивает передовые методы для манипулирования строками и текста;
- Легко интегрируется в свои программы серверы баз данных высокой производительности (DB2, MySQL, Oracle и Sybase);
- Благодаря VHLL программа Рубин хорошо масштабируемой и простой в обслуживании;

- Простой и чистый синтаксис намного проще для программистов первые шаги в изучении этого языка;
 - Существует простой интерфейс программирования для создания многопоточных приложений;
 - Имеет расширенные средства для работы с массивами;
 - Может быть расширены возможности языка с использованием библиотек, написанных на С или Руби;
 - Зарезервированные слова могут быть идентификаторы, если он не создает неоднозначности для синтаксического анализа;
 - Дополнительные возможности для обеспечения безопасности;
 - Встроенный отладчик.
- Недостатки:
- Языковая подготовка выше начального уровня может быть затруднено;
 - Информационные ресурсы, посвященные Ruby, не хватает;
 - Рубин менее продуктивным по сравнению со многими другими языками, используемых в веб-разработке. [15]

3 Реализация и разработка сайта

3.1 Разработка на Ruby on Rails

Rails - это полная, многоуровневая платформа для создания веб-приложений, использующих базы данных, которая основана на архитектуре Model-View-Controller (Model-View-Controller, MVC).

Динамический AJAX-интерфейс, обработка запросов и выдача данных в контроллер, предметной области, отражены в базе данных - для всего из Rails предоставляет однородную среду разработки в Ruby. Все, что вам нужно для начала - база данных и веб-сервер.

Rails использовали все - от стартапов и некоммерческих организаций до крупных предприятий. Rails - это прежде всего инфраструктура, поэтому среда идеально подходит для любого типа веб-приложений, будь то программа для совместной работы, поддержки сообществ, электронного бизнеса, управление контентом, управление статистики.

Rails хорошо работает с несколькими веб-серверов и баз данных. В качестве веб-сервера рекомендуется использовать Apache или Nginx модуль Phusion пассажира. Rails также могут быть развернуты с помощью Unicorn, Thin, Mongrel или FastCGI. В качестве базы данных, можно использовать MySQL, PostgreSQL, SQLite, Oracle, SQL Server, DB2 или Firebird. Использование Rails может быть практически любой операционной системы, но мы рекомендуем развернуть систему семьи * Nix.

Преимущества Ruby On Rails для разработки веб-проектов:

- Большой, процветающее сообщество.
- Огромное количество библиотек и кода.
- Разработка скорости, используя Рубин на Rails веб-приложения в разы меньше.
- Рубин язык проще, чем PHP, и более интуитивным синтаксиса.
- Представление. С каждой новой версии оптимизирован Руби и Ruby 1.9 обогнал производительность PHP, Perl, python3.

3.2 Описание программы

Проект был реализован на языках HTML – язык разметки, Javascript - прототипно-ориентированный сценарный язык, CSS - формальный язык описания внешнего вида документа, динамический язык программирования с открытым исходным кодом.

В главной индексной странице представлены в основной контент сайта. Это последние обзоры матчей всех известных лиг мира. Как дополнительный контент представлены аналитические программы по разбору матчей.

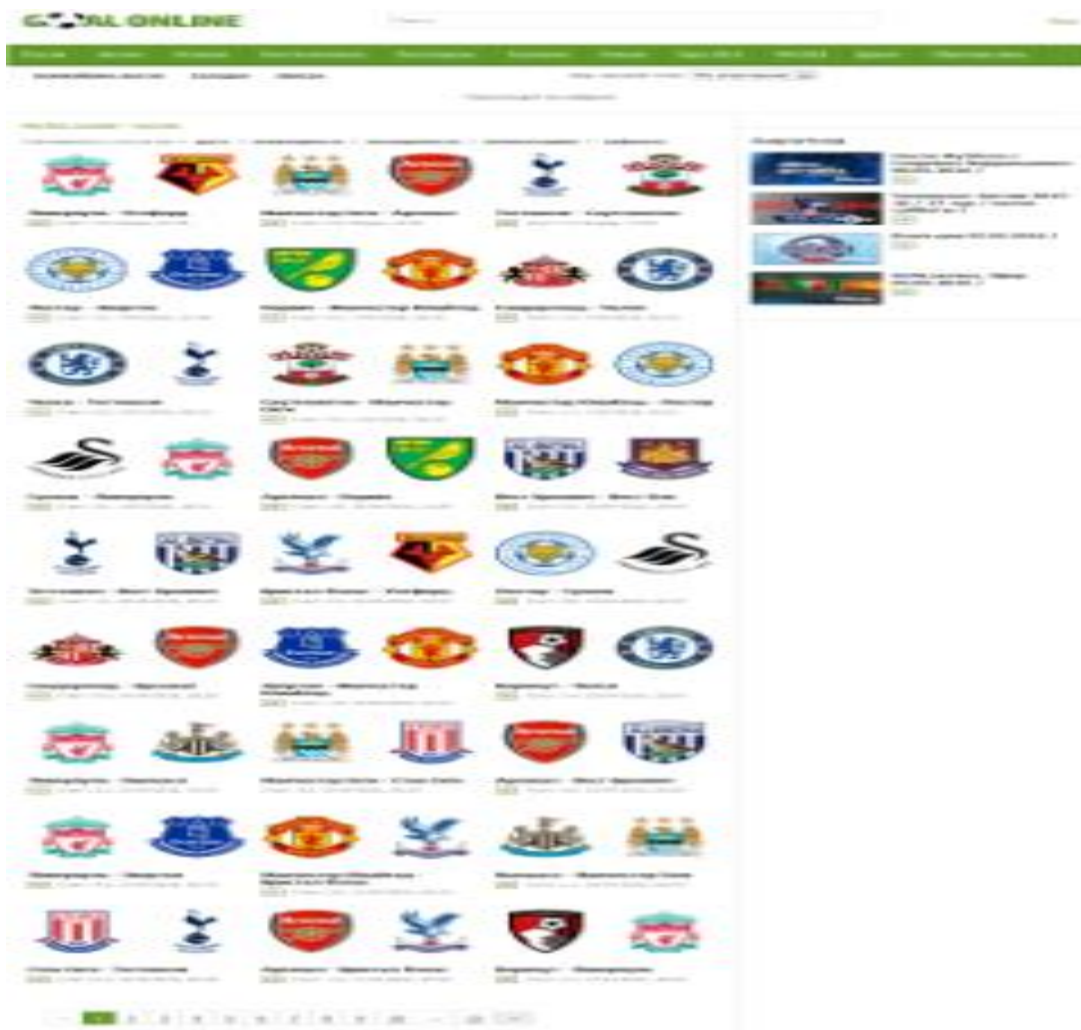


Рисунок 3.1 - Главная страница сайта golonline

В верхней части находится название сайта, поисковик, вход в регистрацию на сайт, меню, сортировка по датам, выбор часовых поясов, ссылки в виде эмблем клубов на информацию о клубе. В нижней части сайта находится основной контент и дополнительный. Основной находится по середине и занимает всю левую часть а дополнительный находится справа от основной части. Есть также краткое описание сайта и обратная связь.

Чтобы получать смотреть видео-обзоры не обязательно регистрироваться на сайте, сайт является бесплатным. Форма регистрации показаны на скрине и реализованы с помощью Javascript. Проверять правильность введенной информации в Textarea через регулярные выражения. Например: Существует ли логин или пароль, совпадают ли они. Можно было добавить капчу для предотвращения попыток спам-роботов. В дальнейшем надеюсь при дальнейшем развитие и прогрессе сайта добавим и эту функцию.

Регистрация

Ваш логин:

Ваш пароль:

Рисунок 3.2 - Форма регистрации сайта

Если пользователя уже регистрировался до этого можно просто в форме ввести логин и пароль и войти под своей учетной записью.

Главная страница

Ваш логин:

Ваш пароль:

[Зарегистрироваться](#)

Рисунок 3.3 - Форма авторизации сайта

При выборе “Вход” начинается проверка логина и пароля пользователя и он получает возможность оставлять комментарий к видео-обзорам.

Программный код представлен в приложения А.

Для каждого видео-обзора на основной странице выделена область width:175px и height:150px. Вот так выводится только эмблема клубов, название команд, счет, время событие и возможное максимальное качество видео (HD).



Рисунок 3.4 - Основной контент

На главной странице сайта представлены разное количество блоков. Это всем известные на мировой арене лиги а также во вкладке Другие представлены две лиги Украины, Франции, Других чемпионатов, Товарищеских матчей и Сборных. Ниже они также представлены.

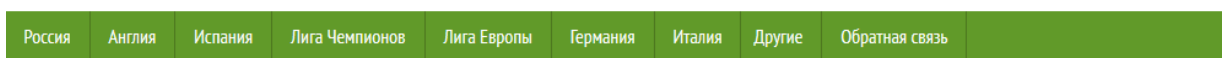


Рисунок 3.5 - Блок меню

В блоки меню во вкладке другие представлены разделы: Украина, Франция, другие чемпионаты, товарищеские матчи, сборные.

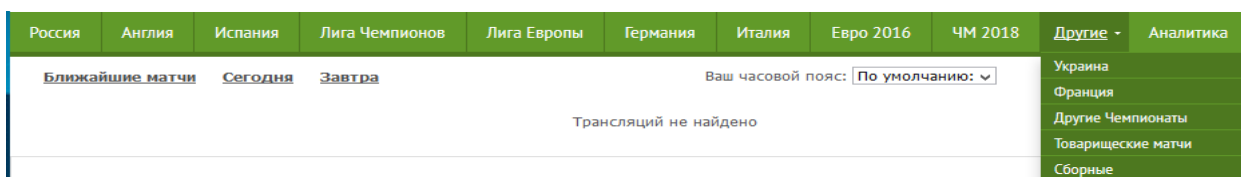


Рисунок 3.6 - Блок меню с вкладкой

Под этим блоком меню идет сортировка данных по датам. Это сортировка по Ближайшим матчам и сортировка на сегодня и завтра. Сортировка была сделана с помощью Tablesorter — это плагин для JavaScript-библиотеки jQuery, который позволяет сортировать HTML-таблицы, включающие теги thead и tbody, без перезагрузки страницы на стороне клиента. Tablesorter может анализировать и отсортировать различные типы данных, включая связанные данные в ячейке. Плагин имеет множество функций.

[Ближайшие матчи](#) [Сегодня](#) [Завтра](#)

Рисунок 3.7 - Сортировка по датам

Ваш часовой пояс:

Рисунок 3.8 - Выбор часового пояса

Следующий блок это блок с информацией о клубе и прошедших его матчах. Он сделан в виде ссылки на картинке.

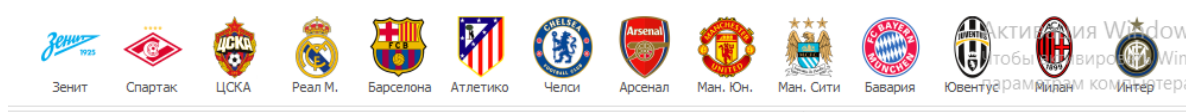


Рисунок 3.9 - Блок контент о клубах

Пройдя по ссылке изображения можно увидеть вот такую информацию.

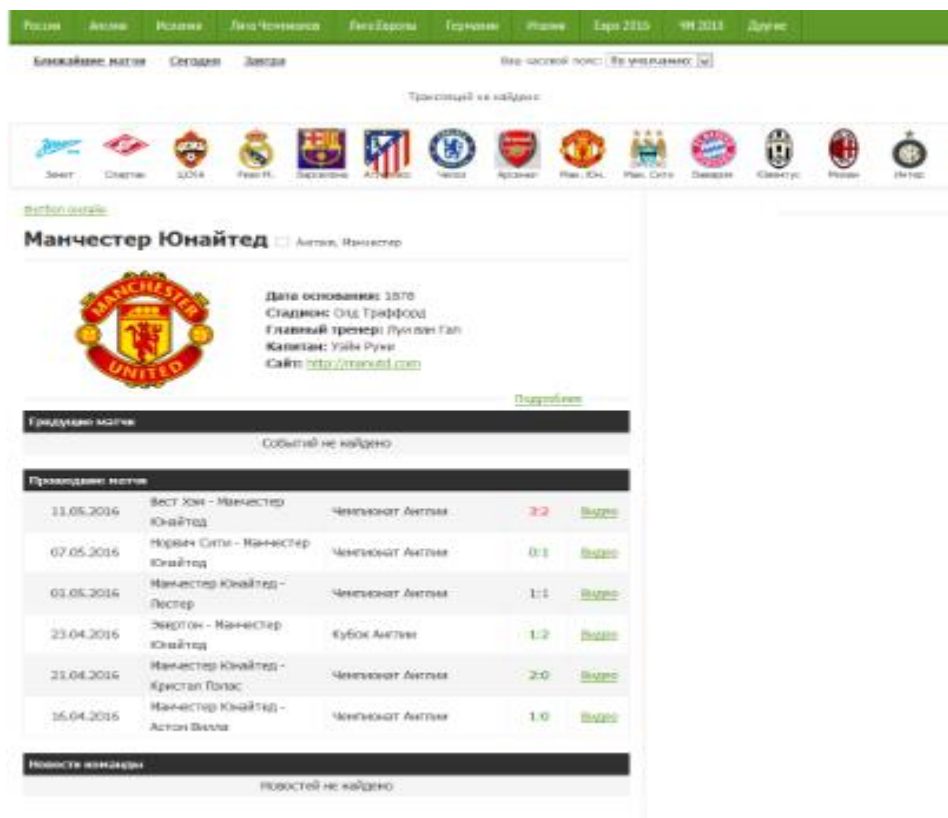


Рисунок 3.10 - Информация о клубе и о матчах клуба

Можно выбрать во вкладках часовой пояс и сайт сам подберет вам ближайшие матчи.



Рисунок 3.11 – Часовой пояс

Нажав на ссылку Подробнее можно увидеть всю информацию о клубе то есть её историю.

Манчестер Юнайтед + Англия, Манчестер



Дата основания: 1878

Стадион: Олд Траффорд

Главный тренер: Луи ван Гал

Капитан: Уэйн Руни

Сайт: <http://manutd.com>

[Подробнее](#)

Футбольный клуб «Манчестер Юнайтед» (англ. Manchester United Football Club; английское произношение: [ˈmæntʃɪstəː(juːˈneɪtəd)] — английский профессиональный футбольный клуб из Стретфорда, Большой Манчестер. Был основан в 1878 году под названием «Ньютон Хит (Ланкашир энд Йоркшир Рейлвей)», в 1902 году изменил название на «Манчестер Юнайтед». Один из самых популярных футбольных клубов в мире. Один из клубов-основателей английской Премьер-лиги в 1992 году. Домашний стадион «Юнайтед», «Олд Траффорд», был открыт в 1910 году. Он является крупнейшим после «Уэмбли» футбольным стадионам в Англии и вмещает около 76 тыс. болельщиков. «Манчестер Юнайтед» является одним из самых успешных английских клубов по количеству титулов за всю историю, а за последние 20 лет — самым успешным клубом в Англии, выиграв 27 крупных трофеев с ноября 1986 года, когда команду возглавил сэр Алекс Фергюсон. В сезоне 2012/13 «Юнайтед» выиграл чемпионат Англии в рекордный 20-й раз. В 1968 году «Манчестер Юнайтед» стал первым английским клубом, выигравшим Кубок европейских чемпионов, победив в финале португальскую «Бенфику» со счётом 4:1. В 1999 году клуб во второй раз выиграл главный европейский клубный трофей, одолев в финале немецкую «Баварию» со счётом 2:1. В 2008 году «Юнайтед» вновь выиграл Лигу чемпионов, обыграв в финале лондонский «Челси». Кроме того, «Юнайтед» является рекордсменом по количеству выигранных Кубков Англии (11 побед). На данный момент «Юнайтед» входит в тройку футбольных клубов с самыми высокими доходами. В апреле 2012 года журнал Forbes оценил клуб в \$2,235 млрд (£1,43 млрд). «Манчестер Юнайтед» является самым дорогостоящим спортивным клубом в мире. Главным тренером команды является нидерландский специалист Луи ван Гал. Он сменил исполняющего обязанности главного тренера Райана Гиггса, назначенного на временную должность после увольнения Дэвида Мойеса. Капитаном команды является Уэйн Руни, сменивший ушедшего из клуба летом 2014 года Нэнэню Видича.

Грядущие матчи

Событий не найдено

Прошедшие матчи

21.05.2016	Кристал Пэлес - Манчестер Юнайтед	Кубок Англии	1:2	Видео
18.05.2016	Манчестер Юнайтед - Борнмут	Чемпионат Англии	3:1	Видео
11.05.2016	Вест Хэм - Манчестер Юнайтед	Чемпионат Англии	3:2	Видео
07.05.2016	Норвич Сити - Манчестер Юнайтед	Чемпионат Англии	0:1	Видео

Рисунок 3.12 - Подробная информация о клубе

Так же есть две обновляемые таблицы. Это «Грядущие матчи» и «Прошедшие матчи». В это таблице также можно узнать дату, название турнира, счет и без особых проблем пройти по ссылке и посмотреть этот матч.

Грядущие матчи				
Событий не найдено				
Прошедшие матчи				
21.05.2016	Кристал Пэлэс - Манчестер Юнайтед	Кубок Англии	1:2	Видео
18.05.2016	Манчестер Юнайтед - Борнмут	Чемпионат Англии	3:1	Видео
11.05.2016	Вест Хэм - Манчестер Юнайтед	Чемпионат Англии	3:2	Видео
07.05.2016	Норвич Сити - Манчестер Юнайтед	Чемпионат Англии	0:1	Видео
01.05.2016	Манчестер Юнайтед - Лестер	Чемпионат Англии	1:1	Видео
23.04.2016	Эвертон - Манчестер Юнайтед	Кубок Англии	1:2	Видео

Рисунок 3.13 - Матчи определенного клуба

Есть также дополнительный блок в который можно заносить все перенесенные матчи или отмененные из за каких-то проблем.

Новости команды
Новостей не найдено

Рисунок 3.14 - Новости команды

С правой стороны основного контента находится дополнительный блок «Аналитика». В ней представлены аналитика по теме футбол.

Аналитика



Про Футбол. Эфир
22.05.2016 /
HD



8-16 / Эфир 22.05.2016 /
HD



Кубок Англии 2015-16 /
Финал /
HD



Чемпионат России
2015-16 / 30-й тур /
HD

Активация Window

Рисунок 3.15 – Аналитика

Чемпионат Англии 2015-16 / 37 тур / матчи субботы / видео обзор

Автор: first

Рисунок 3.16 – Обзор аналитики

Под обзор есть комментарий социальных сетей Вконтакте то есть один его из главных плюсов меньшая нагрузка на сервер за счет отправки меди(фото, видео). Он добавлен через `VK.init({apiId: мой айди, onlyWidgets: true});`



Рисунок 3.17 - Комментарий к видео

Также есть «Обратная связь» для связи с разработчиком, если возникнут проблемы, вопросы или будет деловое предложение можно будет написать мне первому Администратору сайта (first) и разработчику сайта. Тут есть майл(gmail, mail.ru). Также для проверки анти-бот добавлена капча(CAPTCHA) рисунок с кодом.

```
<script src="https://www.google.com/recaptcha/api.js" async
defer></script>
<div class="g-recaptcha" data-sitekey="your_site_key"></div>
```

Обратная связь

Написать сообщение администратору

Ваше имя: *

E-Mail: *

Заголовок: *

Получатель: *

Сообщение:

Код:

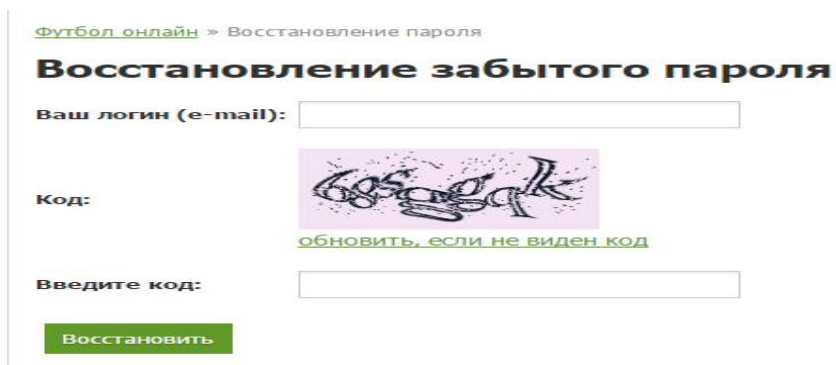
Включите эту картинку для отображения кода безопасности
[обновить, если не виден код](#)

Введите код:

Отправить

Рисунок 3.18 - Обратная связь


Восстановление пароля с капчей. Recoverable: позволяет восстанавливать забытый пароль. Отправляет инструкции по восстановлению на почту.



Футбол онлайн > Восстановление пароля

Восстановление забытого пароля

Ваш логин (e-mail):

Код: 

[обновить, если не виден код](#)

Введите код:

Рисунок 3.19 – Восстановление пароля с капчей

Можно также обратить внимание на поисковик по сайту. Ferret — движок, предназначенный для Ruby в целом, а не только для Rails приложений. Использование его в Rails проще чем можно себе представить. Для интеграции Rails и Ferret нужно установить плагин Acts As Ferret в свой сайт.



Поиск

Рисунок 3.20 – Поисковик

Один из самых нужных вещей для любителя футбола это прямая онлайн трансляция. Возможность смотреть футбол онлайн и находить максимально большое количество качественных источников, благодаря которым можно полностью насладиться просмотром матча с участием любимой команды.



Рисунок 3.21 – Прямая трансляция через другой источник

Тут представлена диаграмма архитектура Model-View-Controller

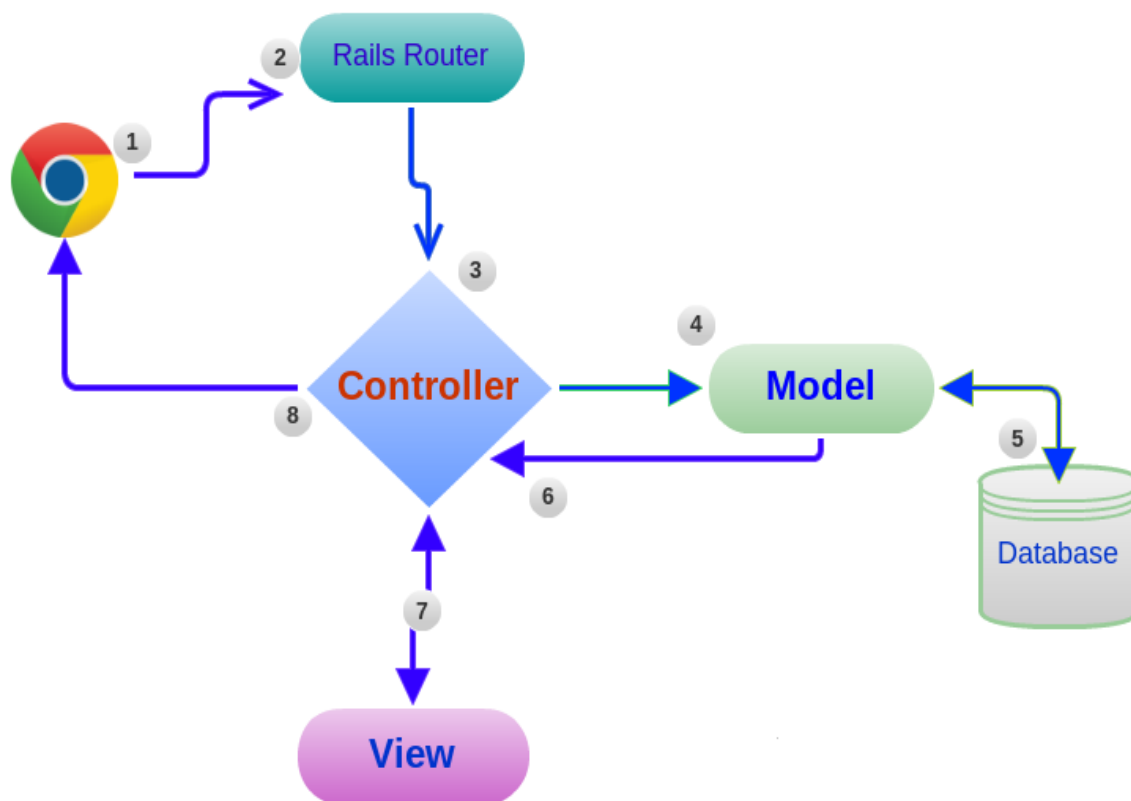


Рисунок 3.22 - Архитектура MVC

3.3 Тестирование

Тестирование является заключительным этапом разработки веб-сайта и играет важную роль в создании качественного программного обеспечения. Чем сложнее сайт, тем больше времени требуется для тестирования и отладки его. К сожалению, есть много примеров, когда разработчики и заказчики пропускают этап тестирования сайта, который почти всегда приводит к большой экономии средств и времени в будущем, недовольство пользователей ресурсом и, как следствие, необходимость улучшения (или даже повторного развития) ресурса. В зависимости от конкретного проекта, тестирование может быть выделено до 50% от бюджета и временных ресурсов.

После завершения основных работ по созданию программной части веб-ресурса, повышения качества специалиста управления (другими словами - тестер) получает все необходимые документы и материалы на сайт и начать тестирование. Для организации тестирования веб-сайта при условии, специально разработанный метод, в соответствии с которым контроль проводится.

Тестирование сайта по пунктам:

1) Оно начинается с подготовительной работы - тестер проверяет полученные документы (анализ функциональных возможностей для

работы, изучает финальные макеты плана участка и тест для дальнейшего тестирования) ..

2) Функциональное тестирование - самый длинный этап тестового ресурса. Суть этого процесса заключается в проверке всю функциональность, описанную:

- Проверьте работу всех обязательных функций сайта;
- Тест производительности пользовательских форм в режиме онлайн (например, обратная связь, добавьте комментарий в блоге);
- Проверьте, поиск работы (в том числе актуальность результатов);
- Проверки гиперссылок, поиск битых ссылок;
- Проверки подгрузки файлов на сервере;
- Эксплуатационные проверки счетчиков, установленных на сайте;
- Просмотр содержимого страниц, чтобы соответствовать оригинальному контенту сайта, предоставленным заказчиком.

3) макет тест - при проверке макета первой вещи тестер проверяет расположение элемента, соответствующего их позиций, предусмотренных макеты, а также проверки изображений и графики оптимизации. Далее проверяется правильность кода. Во время компоновки важно соблюдать правильную иерархию объектов, и важно, чтобы проверить его достоверность после завершения. Браузеры, несмотря на явно неверный код, в любом случае, попытайтесь отобразить веб-страницу. Но поскольку не существует единых правил о том, как будет показано "кривой" документ, каждый браузер пытается сделать это самостоятельно. А это, в свою очередь, приводит к тому, что тот же документ может отличаться в разных браузерах. Исправление очевидных ошибок и систематизации кода, как правило, приводит к стабильному результату. После завершения проверки на действительность, специалист выполняет проверку на совместимость с различными браузерами, т.е. проверяет, что сайт в различных браузерах, а также для различных настроек дисплея.

4) Тестирование безопасности - На данном этапе, проверка специалист тестирования - доступ ли пользователи в офисе / частных страниц, а также проводит проверку безопасности всех критических страниц (например, администрирование раздел сайта) от внешнего воздействия.

5) сайта Тестирование производительности - проводится с целью определения скорости веб-сайта или его частей под определенной нагрузкой. Тестирование производительности включает в себя такие типы *testestirovaniya*:

б) нагрузочное тестирование - самая простая форма тестирования производительности. нагрузочное тестирование обычно проводится для того, чтобы оценить поведение сайта (или приложения) для данной ожидаемой нагрузки. Эта нагрузка может быть, например, ожидаемое количество пользователей, одновременно работающих на площадке, совершающие заданное число транзакций в интервале времени. Этот тип

тестирования обычно обеспечивает время отклика всех наиболее важных бизнес-функций.

7) Скорость испытания - проверить скорость загрузки сайта, чтобы определить скорость отработки сценария, загружать изображения и содержание. Этот тест проводится для того, чтобы оптимизировать процесс загрузки веб-сайта, а также определение оптимальной конфигурации сервера.

4 Экономическое обоснование проекта

4.1 Трудовые ресурсы, используемые в работе

В данной работе мы используем умственный труд, стоимость которого выше, чем стоимость физического труда.

В процессе разработки были использованы:

- Дизайнер;
- Программист;
- Глава.

Количество сотрудников, которые принимали участие в проектировании приведены в таблице 4.1 [16].

Таблица 4.1 – Сотрудники и их заработная плата

Должность	Количество человек	Зарботная плата в месяц, тенге
Программист	1	70000
Дизайнер	1	60000
Руководитель проекта	1	100000
Итого	3	230000

4.2 Оборудование, используемое в работе

Оборудование, используемое при разработке сайта представлено в таблице 4.2.

Таблица 4.2 – Перечень оборудования, необходимого для разработки

Наименование изделий	Характеристика	Количество единиц	Цена за единицу, тенге	Общая сумма, тенге
Ноутбук	AMD A6-3400M APU GeForce GT 940M 4GB HDD 500Gb 15.6" LED LCD (16:9)	2	75000	150000
Многофункциональное устройство	Acer Core i3	1	30000	30000
Итого				180000

Цены на оборудование представлены без учета НДС.

4.3 Программное обеспечение, используемое в работе

При разработке интернет-магазина с помощью следующего программного обеспечения был использован:

- Windows 7 - операционная система;
- Обратите внимание ++ HTML - редактор;
- Denwer - программная оболочка для создания и отладки сайтов.

Программное обеспечение, используемое в разработке сайта представлена в таблице 4.3.

Таблица 4.3 – перечень программного обеспечения, необходимого для разработки интернет-магазина светотехники

Программное обеспечение	Стоимость, тенге
Windows 7 максимальная	15000
Note++	бесплатно
Denwer	бесплатно
Итого	15000

Цены на ПО приведены без учета НДС.

4.4 Сроки реализации проекта

Процесс обучения и дата участка исследования состоит из 6 этапов:

- Коллекция содержания футбола для сайта;
- Исследования и разработки базы данных;
- Инженерный дизайн;
- Бэкэнда и внешний интерфейс;
- Тестирование;
- Редактирование отчетов.

Этапы и сроки реализации проекта представлены в таблице 4.4.

Frontend - это привлечение новых клиентов. Общедоступная продукты должны быть дешевыми, но не бесплатно. В переднем конце главное - не получить контакт потенциальных клиентов и завоевать доверие клиентов платить. Чем дороже передний конец, тем меньше будет клиентская база как результат, но тем активнее он будет покупать бэкэнд продукты. В идеале, внешний интерфейс должен быть на 100%, чтобы покрыть свои расходы на рекламу, чтобы получить самопогашающихся ведет. На переднем конце вы не должны пытаться заработать деньги, это задача бэкэндом [16].

Бэкэнда приложение - то, что приносит много денег. Очень дорогой продукт или услугу. Непосредственно для рекламы не очень эффективна (высокая стоимость рекламы), но игнорировать эту возможность не должно быть (затраты на рекламу компенсировать стоимость продукта). Backend, как правило, рекламируется привлечь фронтального клиентскую базу.

Таблица 4.4 - Этапы и сроки реализации проекта

Перечень работ	Недели от начала работ
----------------	------------------------

		1	2	3	4	5	6	7	8	9	10
1 этап	Постановка задачи	■									
	Проектирование дизайна сайта	■	■								
	Планирование и парсинг литературы		■	■							
	Парсинг материала для заполнения контента сайта		■	■							
2 этап	Разработка и построение базы данных			■	■						
	Внедрение в базу данных				■	■					
3 этап	Проектирование интерфейса сайта					■	■	■			
4 этап	Фронтенд сайта					■	■	■			
	Бэкенд сайта						■	■	■		
	Разработка админ. Панели							■	■	■	
5 этап	Тестирование сайта									■	
	Отладка сайта									■	
6 этап	Оформление НИР										■
	Тест и отправка отчета										■

4.5 Расчет стоимости работы по разработке

Оплата затраты трудовго времени на разработку плана на каждом этапе, а также трудовые издержки.

Издержки на проработку данного плана вычисляется по формуле

$$C = \text{ФОТ} + O_c + A + \text{Э} + C_{\text{пр}} + H; \quad (4.1)$$

где
 ФОТ - фонд оплаты труда;
 O_c - социальный налог;
 A - амортизационные отчисления;
 Э - затраты на электроэнергию;
 $C_{\text{пр}}$ - прочие расходы;
 H - накладные расходы.

4.6 Расчет затрат на оплату труда

Издержки на плату работы для персонала, принимающего участие в проектировании проекта вычисляется по формуле:

$$\text{ФОТ} = \text{З}_{\text{осн}} + \text{З}_{\text{доп}} ; \quad (4.2)$$

где $\text{З}_{\text{осн}}$ - основная заработная плата;

$\text{З}_{\text{доп}}$ - дополнительная заработная плата.

программист, дизайнер и руководитель обуславливаются на договорной стоимости в объеме 70000, 60000 и 100000 тенге.

На ступенях программы участники разработают проект вовлечены неравномерно, для этого необходимо вычислить среднюю дневную заработную плату, а затем полный объем заработной платы, в зависимости от фактического участия. [17]

Средний дневной доход работника вычисляется по формуле:

$$D = \frac{\text{ЗП}_m}{D_r} ; \quad (4.3)$$

где ЗП_m – месячный объем платы;

D_r – объем рабочих дней в месяце (это 26 дней – шестидневная рабочая неделя).

1) Программист

$$D = \frac{70000}{26} = 2692,3 \text{ тенге/день};$$

2) дизайнер

$$D = \frac{60000}{26} = 2307,6 \text{ тенге/день};$$

3) руководитель

$$D = \frac{100000}{26} = 3846,15 \text{ тенге};$$

где ЗП_m – месячный объем платы;

D_r – объем трудовых дней в месяце;

$Ч_p$ – Объем времени трудового дня ($Ч_p = 8$).

Делаем расчет заработной платы за определенный час:

1) программист

$$H = \frac{70000}{26 * 8} = 336,5 \text{ тенге/час};$$

2) дизайнер

$$H = \frac{60000}{26 * 8} = 288,5 \text{ тенге/час};$$

3) руководитель

$$H = \frac{100000}{26 * 8} = 480,8 \text{ тенге/час.}$$

Продолжительность цикла в днях по каждому из работ вычисляется по формуле:

$$t_n = \frac{T}{q_n * z * K}; \quad (4.5)$$

где T – трудозатратность каждого шага, час;

q_n – объем работников по каждому шагу;

z – длительность каждого трудового дня, $z = 7$ часов;

K – коэффициент выполнения норм времени, $K = 1,2$.

Наше значение t_n доводим в наибольшую сторону до полных рабочих дней.

$$t_1 = \frac{24}{1 * 7 * 1,2} \approx 3 \text{ - программист, выделение нашего задачи.}$$

Для всех этапов работ вычислены похожие расчеты:

$$t_n = 1 + 6 + 7 + 3 + 8 + 3 + 3 + 8 + 14 + 5 + 3 + 1 + 1 + 1 = 60 \text{ дней.}$$

Таким образом, для проведения всех работ необходимо 60 дней.

Резюме персонала на оплату труда, участвующих в разработке проекта, приведены в таблице 4.5. Таблица 4.5 – Сводные данные по расчету заработной платы персонала

Наименование этапов	Исполнитель	Трудоемкость, норма-час	Длительность цикла, дни	Заработная плата за час работы, тенге	Сумма заработной платы, тенге
Постановка нашего задание	Программист	22	3	326,5	8076
Определение дизайна	Дизайнер	32	4	288,5	9232
Парсинг и изучение контента части	Программист	22	3	326,5	8076
Парсинг материала и наполнение контента сайта	Программист	30	4	326,5	10768
Расчет для базы данных	Программист	42	6	336,5	16152
Наполнение контента базы данных	Программист	22	3	336,5	8076
Фронтенд сайта	Программист	62	8	326,5	21536
Бэкенд сайта	Программист	101	13	326,5	34996
Проектирование головной. панели	Программист	35	6	326,5	16152
Тестирование программы	Программист	22	2	326,5	5384
Отладка программы	Программист	28	2	326,5	5384
Оформление отчета	Руководитель	5	1	480,8	3846,4
Тесты и подготовка отчета	Руководитель	4	1	480,8	3846,4
Итого		470	60	4903,6	113064

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

$$Z_{\text{доп}} = Z_{\text{осн}} * 0,1; \quad (4.6)$$

и составит:

$$З_{\text{доп}} = 113064 * 0,1 = 11306,4 \text{ тенге.}$$

Таким образом, издержки на расплату работы произведенным расчетам и по формуле 4.2 составит

$$\text{ФОТ} = 113064 + 11306,4 = 124370,4 \text{ тенге.}$$

4.7 Расчет затрат по социальному налогу

Социальный налог в общем будет 11% (статья 358 пункт первый НК РК) от общего дохода работника, и вычисляется вот так:

$$O_c = (\text{ФОТ} - \text{ПО}) * 11\%; \quad (4.7)$$

где ПО – пенсионные отчисления, в общем будет 10% из ФОТ а социальным налогом не убираются, определяется отчисления через формулу:

$$\text{ПО} = \text{ФОТ} * 10\%; \quad (4.8)$$

$$\text{ПО} = 124370,4 * 0,1 = 12437,04 \text{ тенге.}$$

Таким вот методом, в общем все с определенными вычислениями по согласной формуле 4.7 объем отчислений на социальные нужды будет:

$$O_c = (124370,4 - 12437,04) * 0,11 = 12312,7 \text{ тенге.}$$

4.8 Расчет амортизационных отчислений

Амортизационные отчисления определяются по формуле

$$A_i = \frac{H_A * C_{\text{ПЕР}} * N}{100 * n}; \quad (4.9)$$

где H_A – норма амортизации;

$C_{\text{ПЕР}}$ – первоначальная стоимость оборудования;

N – количество дней на выполнение работ;

n – количество рабочих дней в году.

Норма амортизации на вычислительную технику обойдется около 40% от общей стоимости всей нашей техники, на ПО - 15%.

Таким вот методом, амортизационный отчисления по определяемой техники, в соответствии с формулой 4.9 составят:

- для ноутбука

$$A_1 = \frac{45 * 75000 * 4 * 70}{100 * 365} = 9863,01 \text{ тенге};$$

- на многофункциональное устройство

$$A_2 = \frac{30 * 30000 * 3}{100 * 365} = 131,5 \text{ тенге};$$

- на программное обеспечение

$$A_3 = \frac{25 * 15000 * 70}{100 * 365} = 369,9 \text{ тенге};$$

$$A_i = 9863,01 + 131,5 + 369,9 = 10364,41 \text{ тенге.}$$

Сводные результаты расчета амортизационных отчислений представлены в таблице 4.6.

Таблица 4.6 – Сводные данные по расчету затрат на амортизацию

Название техники	Объем	Норма амортизации, %	Сумма амортизации, тенге
Ноутбук	2	45	9863,01
Функциональное оборудование	1	45	131,5
ПО	2	10	369,9
Итого			10364,41

4.9 Расчет затрат на электроэнергию

Поскольку расчеты используются для получения электроэнергии электротехнику, необходимо определить потребность в электроэнергии. Потребности в электроэнергии для нужд исследования может быть потребность электроэнергии для работы оборудования и дополнительных затрат. И должен вычисляется по формуле:

$$\mathcal{E} = \mathcal{Z}_{\text{эл.эн.об.}} + \mathcal{Z}_{\text{доп.}}; \quad (4.10)$$

где $\mathcal{Z}_{\text{эл.эн.об.}}$ – нужды на электричество на технику;

$\mathcal{Z}_{\text{доп.}}$ – нужды на электричество для дополнительных затрат.

Нужды электричество на технику вычисляется по общей формуле

$$\mathcal{Z}_{\text{эл.эн.об.}} = W * T * S * K_{\text{исп}}; \quad (4.11)$$

где W – используемая мощность, Вт;
 T – время общего труда, час;
 S – общий наш тариф (1кВт = 16,02 тенге);
 $K_{исп}$ – коэффициент потребления ($K_{исп} = 0,9$).

$$Z_{эл.эн.об.(\text{ноутбук})} = 0,8 * 150 * 10,02 * 0,5 * 2 = 3027,78 \text{ тенге};$$

$$Z_{эл.эн.об.(\text{МФУ})} = 0,5 * 32 * 10,02 * 0,5 = 415,24 \text{ тенге}.$$

Общая количество нужд на электричество нашего главной техники согласно формуле 4.11 будет

$$Z_{эл.эн.об.} = 3027,78 + 415,24 = 3443,02 \text{ тенге}.$$

Нужды на дополнительные показатели взяты из общего от затрат на техники в объеме 5% и вычисляется по формуле

$$Z_{доп} = Z_{эл.эн.об.} * 5\%; \quad (4.12)$$

и будет

$$Z_{доп} = 3443,02 * 0,05 = 172,2 \text{ тенге}.$$

Таким может быть, общие нужды на электричество, по формуле 4.11 составляют

$$\Sigma = 3443,02 + 172,2 = 3615,2 \text{ тенге}.$$

Таблица 4.7 – Расчеты на затраты и электричество

Название техники	Общий объем	Используемая мощность, Вт	Кол-во рабочих дней	Коэффициент выполнения	Часы работы техники	Общая сумма нужд, тенге
Ноутбук	2	0,7	60	0,9	480	3443,02
МФУ	1	0,9	4	0,9	32	172,2
Итого						3615,2

4.6 Расчет накладных и прочих расходов

также включены те самые другие расходы:

1) Потребность в глобальной сети 55 дней (примерно 2 месяца).
 Стоимость Интернет в месяц составляет 4660 тенге. Мы получаем:

$$4660 * 2 = 9320 \text{ тенге};$$

2) на канцелярские расходы:

- Упаковка из бумаги формата А4 - 2200 тенге;
- Упаковка ручки - 200 тенге;
- Карандаш 3 - 340 тенге;
- Резинка - 30 тенге;
- Линия - 150 тенге;
- Наклейки - 350 тенге.

Общая стоимость канцелярских принадлежностей:

$$200 + 2200 + 30 + 350 + 30 + 200 = 2800 \text{ тенге};$$

3) аренда помещения в течение 55 дней (около 2 месяцев).

Площадь номера составляет:

$$S = 7_{\text{м}} * 7_{\text{м}} = 49 \text{ м}^2.$$

Стоимость одного квадратного метра равна 2600 тенге. Мы получаем:

$$* 2600 \text{ м}^2 * 2 = 108000 \text{ тенге}.$$

Прочие расходы включают в себя

$$SBP = 1800 + 9000 + 108000 = 118800 \text{ тенге}.$$

составляют накладные расходы на 50% от всех затрат и рассчитываются по формуле:

$$H = (\text{ФОТ} + O_c + A + Э + C_{\text{ПР}}) * 50\%; \quad (4.13)$$

$H = (124370,4 + 12312,7 + 10364,41 + 3615,2 + 118800) * 0,5 = 134731,4$ тенге.

4.6 Расчет стоимости по всем статьям затрат и определение структуры затрат

Соотношение общих потребностей формула 4.1 для разработки моего сайта вверх

$$C = 124,370.4 + 12,312.7 + 10364,41 + 3615,2 + 118800 + = 269,462.8$$

404,194.1 тенге.

Предполагаемые затраты на трансляциях дизайн сайта, а также график затраты, полученные в таблице 4.8 и на рисунке 4.1.

Оценка делается на производстве экономических однородных элементов. Элементы должны отражать стоимость определенного вида жизни, и они имеют вид:

- Материальные затраты;- Амортизация основных средств.

Таблица 4.8 – Общие цифры на цены на проектирование сайта

Название затрат	Общие, тенге
Фонд оплаты труда	124370,4
Социальный налог	12312,7
Амортизация	10364,41
Затраты на электроэнергию	3615,2
Прочие расходы	118800
Накладные расходы	269462,8
Итого	538925,5

Диаграмма нужд для нашей разработки

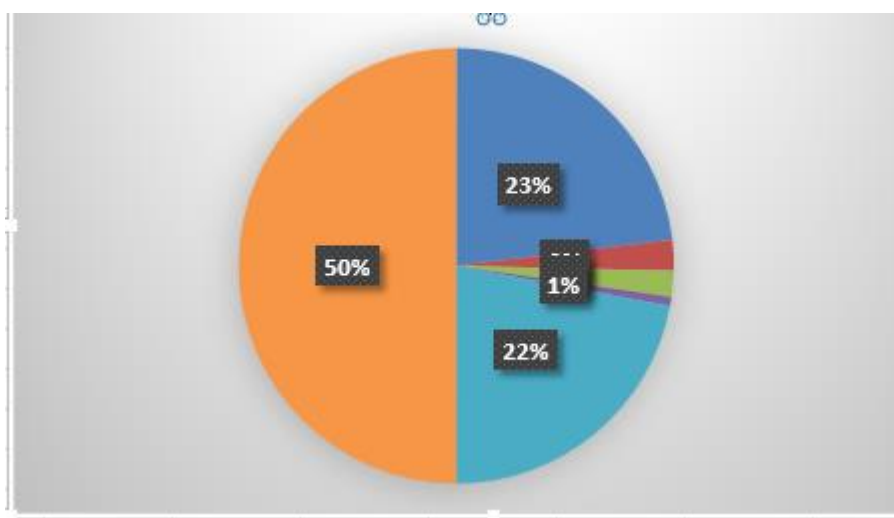


Рисунок 4.1 – Диаграмма нужд на разработку сайта

4.7 Цена интеллектуального труда

Цена проектирование сайта суммируется из общей стоимости и нашего чистого дохода

$$Ц = С + П \quad (4.14)$$

где С – стоимость нашего сайта;

П – чистая прибыль.

При вычисление начальной цены нужно определить для себя нужную уровень рентабельности (здесь 20%) реализации программных продуктов

$$Ц_{п} = С * \left(1 + \frac{Р}{100}\right) \quad (4.15)$$

где Р – рентабельность [17].

$$C_{\Pi} = 538925,5 * \left(1 + \frac{20}{100}\right) = 646710,6 \text{ тенге.}$$

Цена реализации проекта вычисляется из формулы

$$C_p = C_{\Pi} + \text{НДС} \quad (4.16)$$

где НДС – налог на суммарное значение по ставке 12%

$$\text{НДС} = C_{\Pi} * 12\% \quad (4.17)$$

$$\text{НДС} = 646710,6 * 0,12 = 77605,3 \text{ тенге.}$$

В соотношении с формулой 4.16 цена проектирование сайта составит

$$C_p = 646710,6 + 77605,3 = 724315,9 \text{ тенге.}$$

Потребность от спроса, окончательная цена на интеллектуальный труд в условиях наших рыночных отношений будет чаще всего варьироваться и изменяться.

5 Безопасность жизнедеятельности

5.1 Анализ условий труда

Исследование и анализ сайта должно происходить с использованием компьютерных технологий. В некоторых областях работает программист, дизайнер и менеджер проекта.

В нашем исследовании программного обеспечения, которое является основным местом надлежащее распределение ресурсов и условий труда в рабочей комнате.

Так как наш сайт связан с довольно длительным пребыванием для компьютеров, вам нужно выучить некоторые правила искусственного освещения, пожарной безопасности и правил кондиционирования воздуха для обеспечения условий труда человека.

Рассмотрим пример нашей комнаты. В комнате, в дополнение к в дополнение к данной области, могут быть различные вентиляторы и кондиционеры. Комната, где было установлено компьютерное оборудование. Общая часть компьютеров могут быть добавлены, чтобы сделать: лазерный принтер. вентиляторы и внешние шумы:

- Электромагнитное излучение, это наш главный источник монитора;
- Программное обеспечение как менеджер программ и ПК на рабочем месте может чувствовать себя следующие опасности:

- Вредные микробиологические условия, которые определяются температурой в помещении и на улице, его оборудование и сила, относительная влажность в воздухе, а также проветривать на;

- Nеpravlny поддержания на рабочем месте освещение, в связи с ухудшением зрения и sokrashenie эффективности koffitsenta.

Психологические и физические факторы, которые привели к ухудшению состояния здоровья работника [18].

План комнаты показан на рисунке 5.1

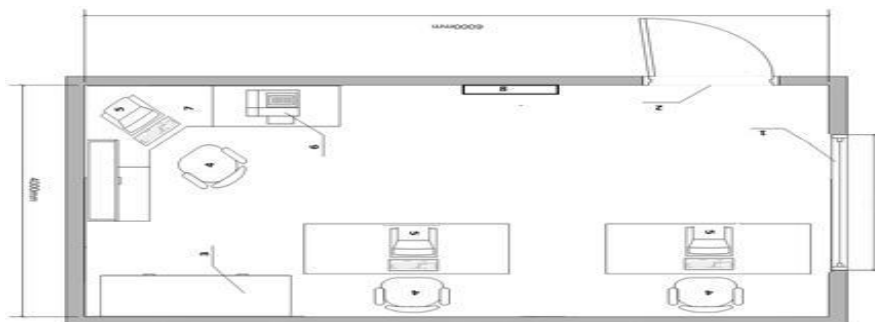


Рисунок 5.1 - Рабочая комната

где 1 - коробка, 2 - 3 двери - кондиционер, 4 - стулья, 5 - компьютеры, 6 - лазерный принтер 7, 8 стенд, 9 - сейф.

Размеры рабочей аудитории:

- Высота - 2,5 м;
- Ширина - 5 м;
- Длина - 7 м;
- Гостиная площадью 35 м²;
- Площадь мебель ≈ 14 м²;

- Комната на из-за не относятся к категории тяжелых (не тяжелая физическая, часть Ia, действия выполняются, сидя и не требуют тяжелого физического труда) (ГОСТ 12.1.022-68);

- Искусственное освещение - свет: 1 лампа, каждый с 1 люминесцентными лампами.

При работе с компьютером вы можете быть потенциально опасные факторы, действие которых по физике это может быть вредно и привести к травмам. Основные особенности с их расположением возникновения и частей сообщили в ГОСТ 22.1.203-44 / 30 и изложены в таблице 5.1.

Таблица 5.1 – Вредные факторы

Название	Место расположение	ППУ, ДДК	Следствия
Высокое Напряжения ЭЦ	Место работника	ГОСТ 11.2.138-92. Упр≤35В	Травмы от удара тока
ЭД	Щит распределение	ГОСТ 11.1.007.4-75 ГОСТ 21.2.104-75	Травмы от огня
Название факторов	Геопозиция	ППУ, ДДК	Что может случиться
Неправильная освещенность	Комната	СДиП 33-03 E=200 лк	Усталость, сонливость Ухудшение зрение

5.2 Расчет искусственного освещения

Расчет искусственного освещения конструкции потребления соотношения метод.

В течение определенной части действия, чтобы выбрать систему освещения, источник световых волн для данного типа лампы или части рабочего помещения.

Необходимо знать количество светильников, необходимых для завершения Эмина и мощности технологии освещения, необходимо

обеспечить, чтобы в нормальных условиях недостаточной освещенности рабочего помещения. [19]. Норма освещенности задана в таблице 5.2 - 400 лк.

Таблица 5.2 - Основная характеристика люминесцентной лампы

Нормальная мощность, Вт	Нормальный световой поток ламп типа T5	Размер ламп, м	
		2P	Длина полная
26	4355	35,58	22

Как мы берем лампу T5 Liline ABCPlus длина F35W / 888. Лампа: 1500 мм, ширина: 255 мм.

Расчет искусственного освещения фактора вычелывает методов потребления.

Коэффициенты потолка Обратное также стены и соотношение пола вам нужно взять с собой:

$$\rho_{\text{пот}} = 70\%; \rho_{\text{ст}} = 50\%; \rho_{\text{пол}} = 30.$$



Рисунок 5.2 – Светильник LG T5

Определим высоту подвеса светильника в рабочей комнате;

$$H = h - h_p - h_c; \quad (5.1)$$

h_c - где длина светильника к потолку, $h_c = 0,11$ м;
 h_p - длина рабочей поверхности на нашем полу;
л.с. = 0,8 м. h - высота помещения, $ч = 3,2$.

$$H = 3,2 - 0,8 - 0,11 = 2,29 \text{ м.}$$

Более успешным является длина от окна иллюминатора определяется как

Предложить исправление

$$L = \lambda \cdot H \quad (5.2)$$

где $\lambda = 1,2 \div 1,4$;

$$L = 1,2 \cdot 2,29 = 2,748 \text{ м.}$$

Длина от стенки до нашего ближайшего осветителя, когда трудимся у стенки не проходит, вычисляем через формулу:

$$l_1 = (0,4 \div 0,5) L; \quad (5.3)$$

коэффициент запаса равен $k_3 = 1,2$.

Вычислим объем люминесцентных лампочек через формулу

$$N = \frac{E_H \cdot S \cdot Z \cdot K_3}{n \cdot \Phi_{л} \cdot \eta}; \quad (5.4)$$

где S- площадь комнаты;

k_3 - коэффициент резерва;

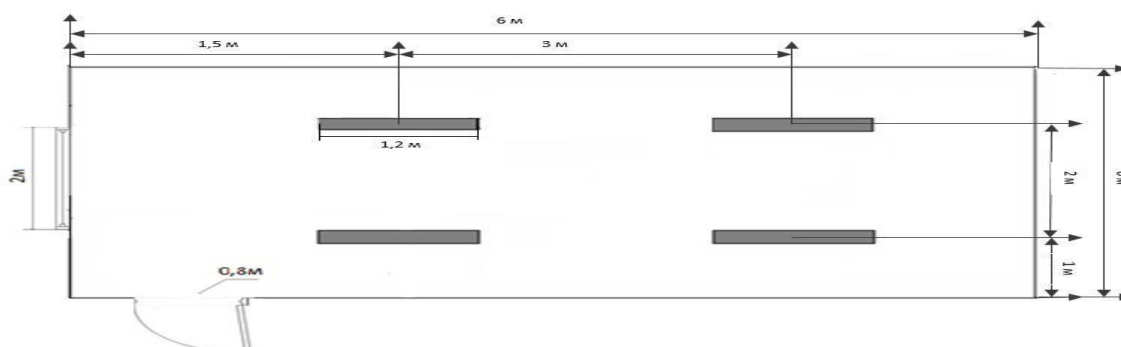
E- заданная самая малая света, $E=400$ лк;

Z- коэффициент неполного света, $Z=1,1$;

n- число лампочек в осветителе;

$\Phi_{л}$ - осветительная полоса нашей лампы, $\Phi_{л}=4550$ лм.

Это было необходимо для стабилизации приемы света 325 лк потребности 5 осветитель 3 люминесцентные лампы в каждой, мощность каждой лампочки составляет 47 Вт, так что нет необходимости увеличивать или уменьшать количество наших ламп, для наших отношений с санитарно правила. [20].



5.3 Разработка вопросов пожарной безопасности

По словам GPIP 3.05.04-74 и DNKS-03-74 конструкцию в зависимости от степени возгорания угроз, от многоцелевых и пожарной нагрузки горючих продуктов, относится ко второй группе норм С.

Результаты возникновения нашего огня могут быть разными:

- Части нашего пожарного оборудования;
- Воспламенение материалов из нерабочих кнопок, розеток;
- Неправильное использование режима соблюдения оборудования, не нормируется эффекта на его сотрудников.

В случае пожара может изменить не только помещение, но и дорогостоящее оборудование, приводит к большой потере жизни. Поэтому необходимо, чтобы принятые меры для продвижения по выявлению и ликвидации пожаров. Исследование подстрекательства могут быть электронные схемы, компьютерные устройства, используемые для технической эксплуатации, блок питания, кондиционеры, там в результате различных ненадо progarelye оборудования и т.д. [21].

В связи с нашими требованиями правил техники безопасности огнеоу номера оснащены Огнетушители углекислотные ОУ-4, принимая во внимание - одно из НУ-4 150 м2. Общее количество номеров составляет 81,1 м2, таким образом, установить 2 ОУ-4. В рамках комбинированного пожаротушения uglekislotno-хладона композиции. Расчетная масса комбинированного мД персонала кислотно-хладона углекислые, для полного подавления огня рассчитывается по формуле 5.6.

$$m_d = k * g_n * v; \quad (5.6)$$

где $k=2,3$ -коэффициент не учитываемых потерь углекислотн-охладоновой смеси;

$g_n=0,25$ -нормативная массова концентрация углекислотно-хладонового состава.

V - объем помещения определяется по формуле 5.7

$$V = A * B * H; \quad (5.7)$$

где $A = 7$ м - длина комнаты; $B = 4$ м - ширина комнаты; $H = 4,4$ м - высота комнаты.

Тогда:

$$V = 6 * 4 * 3,2 = 76,8 \text{ м}^3.$$

И так:

$$m_d = 1,2 * 0,04 * 76,8 = 3,68 \text{ кг.}$$

Расчитанные цилиндров значение x определяется на основании емкости 20-литровую бутылку 12 кг кислоты хладон состав углекислые.

Внутренние сети водоснабжения 2Р-ди (мм), рассчитывается по формуле 5.8.

Аналогичная длина и 2р проточной воды составляет 1,4 кг / с.

Расчетные часы питания uglekislotno-хладона смеси т, рассчитывается по формуле 5.10.

$$d_i = 12 * \sqrt{2} = 17 \text{ мм.}$$

Нужды углекислотно-хладоновой смеси Q, в зависимости от вычисляется по формуле 5.11

$$m = 1,1 * m_d * \left(1 + \frac{k_2}{k_1}\right); \quad (5.11)$$

где $k_2 = 0,2$ - коэффициент вычисляющий остаток углекислотно-хладоновой смеси в баллонах и трубопроводах.

$$m = 1,1 * 3,68 * \left(1 + \frac{0,2}{0,1}\right) = 4,72.$$

Это объясняется тем, что полученные нами решения можно сделать вывод, что для обеспечения нормального функционирования автоматической системы пожаротушения требуется 1 бутылка uglekislotno хладон состава емкостью 20 литров, с массой 7,2 кг смеси. Автоматические установки газового пожаротушения должны автоматически выполняться в соответствии с ГОСТ 12,4.009-83 [22].

5.4 Расчет системы кондиционирования

В нашей таблице 5.2 приведены наиболее необходимые правила с учетом периода года в соответствии с параметрами микроклимата GNiDP 3.02.03-72. Для облегчения физической работы. Техники, установленные в операционной комнате, не является основным источником тепла вызывает (очень небольшой дефляции теплого воздуха техники никоим образом не влияют на рабочий климат помещения). Климатические условия использования технологии в соответствии с нашими климатическими условиями, нормированная к рабочей команде.

В нашем офисе вентиляции номера имеют естественную вентиляцию полосы, проходимого в строительстве зданий и открытых окон в летнее время. Как правило, в теплое время года, чтобы поддерживать необходимый микроклимат имеет кондиционер. Соответствующий климат

в нашем офисе обеспечивает сотрудников здравоохранения, работающих в разное время года, и, следовательно, увеличивает эффективность. А также для того, чтобы поддерживать все стандарты и микроклимат в помещении, рационально обустроить нашу систему кондиционирования воздуха.

Просто установите систему кондиционирования воздуха при расчете рабочего помещения. Кондиционер может обеспечить правильный климат в нашем рабочем климате и нормативными актами.

Общая длина L_s приточного воздуха по формуле:

$$L_{\text{пр}} = \frac{Q}{c \cdot \rho_{\text{пр}} \cdot (t_{\text{выт}} - t_{\text{пр}})}; \quad (5.12)$$

где $Q_{\text{изб}}$ - излишки испускание явной теплоты;

c - удельная теплоемкость в рабочей комнате;

$\rho_{\text{пр}}$ - плотность идущего в комнату воздуха;

$t_{\text{выт}}$ - температура убираемого из комнаты воздуха;

$t_{\text{пр}}$ - температура приточного воздуха, $^{\circ}\text{C}$.

Температура удаляемого из комнаты воздуха $t_{\text{выт}}$, $^{\circ}\text{C}$, вычисляется по формуле:

$$t_{\text{выт}} = t_{\text{рз}} + t^* (h_{\text{вп}} - Z); \quad (5.13)$$

где: $t_{\text{рз}}$ - температура комнате, которая не должна превышать допустимую по нормам ($t_{\text{рз}} < t_{\text{доп}}$), $^{\circ}\text{C}$;

$h_{\text{вп}}$ - расстояние от пола до центра вытяжных проемов (кондиционера), м;

Z - длина рабочей зоны, м.

Так как вычисление производится для теплого периода года, то примем:

$$t_{\text{рз}} = 22 \text{ } ^{\circ}\text{C}.$$

Наша внутренняя часть оборудование находится на высоте $h_{\text{вп}} = 2,8$ м:

$$t_{\text{выт}} = 22 + 1,2 \cdot (2,8 - 3,2) \text{ } ^{\circ}\text{C}.$$

Температура приходящего воздуха $t_{\text{пр}}$ при нахождении избытка явной теплоты должна не быть на 5 - 7 $^{\circ}\text{C}$ ниже температуры воздуха в комнате:

$$t_{\text{пр}} = 22 - 7 = 15 \text{ } ^{\circ}\text{C}.$$

Величину избыточного выделения явной теплоты $Q_{\text{изб}}$ находят на основании баланса теплоты в помещении по формуле:

$$Q_{\text{изб}} = \sum Q - \sum Q_{\text{ух}}; \quad (5.14)$$

где: $\sum Q$ - общее количество поступающей в комнату явной теплоты;
 $\sum Q_{yx}$ - общее количество удаляющийся из комнаты теплоты (за счет тепло-потерь стенами, нагрева приходящего в помещение теплого воздуха и т.п.).

Основными источниками избыточного тепла являются осветители, люди, и другие. Он может также необходимо принимать во внимание тот факт, что усиление тепла от солнечной радиации. В этой комнате наша электронные приборы тепла можно пренебречь. Кроме того, принять во внимание тепло искусственной лампы, от человека, количество тепла, поступающего в помещение через окна от солнечной радиации.

Отвод тепла от искусственного иллюминатора Q_2 , вычисляется в предположении, что практически все уходящей энергии, в конечном счете превращается в тепло, в соответствии с формулой:

$$Q_2 = 1000 * n; \quad (5.15)$$

где N - используемая мощность светильников

$$Q_2 = 1000 * 0,28 * 2 = 480 \text{ Вт.}$$

Тепловыделения от людей Q_3 вычисляют по формуле:

$$Q_3 = q_{ч} * n; \quad (5.16)$$

где: n - число работающих;

$q_{ч}$ - объем тепла, испускаемая одним субъектом

Объем тепла, приходящего в комнату от солнце $Q_{\text{ост.рад}}$, определяют по формуле:

$$Q_{\text{острад}} = F_{\text{ост}} * q_{\text{ост}} * A_{\text{ост}}; \quad (5.17)$$

для оболочки:

$$Q_{\text{острад}} = F_n * q_n * k_n; \quad (5.18)$$

Где $F_{\text{ост}}$ и F_n - объем поверхности и покрытия, м^2 ;

$q_{\text{ост}}$ и q_n - теплоступления через 2м^2 оболочки остекления и оболочки покрытия.

k_n - коэффициент теплопередачи оболочки.

Определение $q_{\text{ост}}$ в зависимости от физической ориентации поверхности и определение окон или фонарей берется в пределах 80 - 120, а коэффициента $A_{\text{ост}}$ в зависимости от типа остекления и его нужных свойств - в пределах 0,36 - 2,35, средние смысл теплоступления от солнечной радиации через оболочку в нужности от географической долготы и типа оболочки принимают в пределах 7 - 35.

$$F_{\text{ост}} = 2,6 * 2,3 * 4 * 1,7 \text{ м}^2.$$

Окно трудовой комнаты направлено на юго-запад.

$$Q_{\text{ост.рад}}=2,7*221*0,66=184 \text{ Вт.}$$

Нормальное значение теплопоступления для оболочки с учитывая географической долготы примем равным $Q_{\text{п.рад}}=27 \text{ Вт.}$

Потери тепла из комнаты $Q_{\text{ух}}$ «кВт» через стенку и двери, окошки оценивают также можно по формуле

$$Q_{\text{ух}}=\frac{\lambda-s*(t_{\text{выт}}-t_{\text{пр}})}{\delta}; \quad (5.18)$$

где λ - теплопроводность стенки

S - площадь, м^2 ;

δ - толщина стен, м.

Стены рабочей комнаты сделаны из тяжелого бетона М 800

$$Q_{\text{ух}}=\frac{2,3*3*(11,1+14)}{0,6}=277,53 \text{ Вт.}$$

Также нужно обеспечивать приход воздуха $L=231.6 \text{ м}^3/\text{ч}$, нужно использовать 2 кондиционера фирмы LG FlexteDQ-19DWS с максимальным расходом воздуха $180 \text{ м}^3/\text{ч}$.

Определим общее количество поступающей в комнату явной теплоты

$$\sum Q = Q_2 + Q_3 + Q_{\text{ост.рад}} + Q_{\text{п.р.р}}; \quad (5.19)$$

$$\sum Q = 480 + 145 + 297 + 18 = 940 \text{ Вт.}$$

Так как вычисления делаются для летнего периода времени количество лишнего испускание явной теплоты равна:

$$Q_{\text{изб}} = 940 \text{ Вт.}$$

Определим суммарный объем воздуха

$$L_{\text{пр}}=\frac{740}{2*2,3*(13,1-23)}=155,4 \frac{\text{м}^3}{\text{ч}}.$$

Нужно дать расход воздуха $L=155.4 \text{ м}^3/\text{ч}$, нужно использовать 2 кондиционера фирмы LG FlexteDQ-19DWS с большим расходом воздуха $180 \text{ м}^3/\text{ч}$.

Заключение

Разработка веб-сайта представляет собой сложный и трудоемкий процесс, требующий больших затрат умственного труда. Стоимость разработки включает в себя следующие категории расходов: вычеты из заработной платы на социальные нужды, амортизация, затраты на электроэнергию.

Наибольший удельный вес в общей стоимости разработки программного обеспечения, затрат на рабочую силу в размере 124,370.4 тенге, что составляет 23% от разработки проекта.

Это цена, которую мы получили, анализируя фонд оплаты труда различных веб-студии дали одинаковое количество рабочих и дней для проекта, как мы делаем. Анализируя расчеты, а точнее, в результате чего стоимость проекта, равной 724,315.9 тенге, то можно сделать вывод о том, что проект является экономически целесообразным.

В ходе тезиса достиг главной цели работы - анализ условий труда. Мы также знаем, что. В комнате, в дополнение к оборудованию и рабочих мест могут быть различные обогреватели и кондиционеры. Помещение, в котором был установлен ноутбук. Комплекс компьютерного оборудования дополнительно включает в себя: лазерный принтер.

Для достижения этой цели мы провели исследование и анализ структуры нашей организации. Расчет показал, что лампа достаточно для рабочей зоны освещения. Было установлено, что общее для создания стандартизированного освещения 400 лк лампы должно быть от 4 до 2 люминесцентные лампы в каждой, мощность каждой лампы должна составлять 36 Вт, а затем, не требуется, чтобы увеличить или уменьшить количество доступного света, соответствует санитарным нормам.

Кроме того, мы решили проблему со свежим воздухом в комнате, что плохо влияет на физиологическое и физическое состояние человека, который работает в этой комнате. Для того, чтобы поток воздуха $L = 122,4$ м³ / ч, вы можете использовать 1 кондиционер компании MonteAQ-Samsung-09AWA с максимальным расходом воздуха 195 м³ / ч.

Там также были приняты во внимание вопросы пожарной безопасности в здании. Для того, чтобы создать нормальную рабочую атмосферу в комнате.

Список литературы

- 1 Пайн К. Учись программировать. - М.:Высшая школа 2006. - 290 с.
- 2 Ashton F. Beginning Ruby: From Novice to Professional. - USA.:Apress 2007. – 645 с.
- 3 Фултон Х. Программирование на языке Ruby. – СПб: Питер, 2015. – 543 с.
- 4 Эдельсон Дж., Лю Г. Ruby на платформе Java. – М.: ВШ, 2014. – 456 с.
- 5 David A. Black : The Well-Grounded Rubyist. - USA.:Apress 2009. – 503 с.
- 6 Дэвид Макфарланд. Большая книга CSS3.-М.:Наука 2009. – 986 с.
- 7 E.Mair: CSS. Cascading Table Sheets.- USA.:Apress 2013. - 547 с.
- 8 Дэвид Флэнаган JavaScript. Подробное руководство.- М.:Наука 2014. - 678 с.
- 9 Бер Бибо. jQuery. Подробное руководство JavaScript.- М.:Наука 2012. – 335 с.
- 10 Лоусон, Шарп. Изучаем HTML 5.- М.:Наука 2000. – 432 с.
- 11 Rob Crowther, Ash and Greg. Wanish HTML5 in Action.- USA.:Manning изд 2002.–2003 с.
- 12 Флэнаган Д. Язык программирования Ruby.-М.:Высшая школа 1999.- 156 с.
- 13 Руби С., Томас Д., Хэнссон Д. Гибкая разработка веб-приложений в среде Rails. 4-е изд. (Rails 3.1).-М.:Наука 2012.-577 с.
- 14 <http://www.railstutorial.ru>
- 15 <http://rusrails.ru/a-guide-to-testing-rails-applications>
- 16 Куликов Л.М. Экономическая теория: Учебник. – М.: Издательство Проспект, 2015. – 432 с.
- 17 В.Д.Камаев. Экономическая теория. – М.: Юрайт издат 2010. - 557 с.
- 18 Ботузка Д.Ч. Охрана труда. – М.: Высшая школа, 1998. – 447 с.
- 19 Дымгин К.С. Освещение рабочего места. М.: Высшая школа, 1999. – 132 с.
- 20 Безопасность жизни. Под ред. Н.А. Белов. – М.: Знание, 2014 – 234 с.
- 21 Справочная книга для проектирования электрического освещения. Под ред. Г.Б. Кнорга. – Л.: Энергия, 1986 – 231с.
- 22 Б.С. Мастрюков. Опасные ситуации техногенного характера. Учебник для вузов / Б.С.Мастрюков.-М.: Академия, 2009. - 320 с.

Приложение А

Диаграмма вариантов использования

