

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ
КАЗАХСТАН

Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ
им. ГУМАРБЕКА ДАУКЕЕВА»
Кафедра IT – инжиниринг

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

ДИПЛОМНЫЙ ПРОЕКТ

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

Специальность 5В070400 – Вычислительная техника и программное
обеспечение

Выполнила: Макеев Д.М. Группа ВТ-16-2

Научный руководитель: PhD, профессор Картбаев Т.С.

Консультанты:

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

(учёная степень, звание, Ф.И.О.)

_____ «___» _____ 2020 г.

по безопасности жизнедеятельности: доцент Приходько Н.Г

(учёная степень, звание, Ф.И.О.)

_____ «___» _____ 2020 г.

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

(учёная степень, звание, Ф.И.О.)

_____ «___» _____ 2020 г.

Нормоконтролер: ст.преп. Абсатарова Б.Р

(учёная степень, звание, Ф.И.О.)

_____ «___» _____ 2020 г.

Рецензент: _____

(учёная степень, звание, Ф.И.О.)

_____ «___» _____ 2020 г.

Алматы 2020

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

Институт систем управления и информационных технологий

Кафедра IT-инжиниринг

Специальность 5В070400 – «Вычислительная техника и программное обеспечение»

ЗАДАНИЕ

на выполнение дипломного проекта

Студенту Макееву Данияру Мухтарұлы

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

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

Срок сдачи законченного проекта «___» _____ 2020 г.

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

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

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

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

Основная рекомендуемая литература:

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

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

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

Консультация по проекту с указанием относящихся к ним разделов проекта

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

ГРАФИК
подготовки дипломной работы (проекта)

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

Дата выдачи задания « ____ » _____ 2020 г.

Заведующий кафедрой _____ А.А.Досжанова
(подпись)

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

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

АҢДАТПА

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

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

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

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

АННОТАЦИЯ

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

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

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

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

ANNOTATION

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

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

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

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

Содержание

Введение	11
1 Аналитика	12
1.1 Представление о web-приложении	12
1.2 Классификация web-приложений и их уникальности	14
2 Описание проекта	18
2.1 Язык гипертекстовой разметки, как основной язык Web-приложения	23
2.2 Язык PHP и его роль в разработке проекта	24
2.3 JavaScript	28
2.4 Notepad++	28
3 Проектирование и разработка программного продукта	32
3.1 Проектирование и разработка формул алгоритмов	32
3.2 Этапы разработки веб-приложения	34
3.3 Разработка дизайна Web-сайта	40
3.5 Проектирование и разработка кода сайта	43
4 Экономический расчет	40
4.1 Цели и задачи, решаемые в экономической части	51
4.2 Расчет трудоемкости разработки ПП	51
4.3 Расчет затрат на разработку ПП	51
4.7 Расчет эксплуатационных затрат	59
4.8 Расчет результатов от создания и использования ИС	61
4.9 Расчет основных показателей экономической эффективности	63
5 Безопасность жизнедеятельности	65
5.1 Анализ потенциально опасных и вредных факторов в офисе, воздействующих на персонал.	65
5.2 Методика расчета установок водяного и пенного пожаротушения	68
Заключение	77
Список литературы	78
Приложение А	80
Приложение Б	82
Приложение В	92

Введение

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

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

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

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

1 Аналитика

1.1 Представление о web-приложении

Web-приложение представляет собой приложение которое применяется в интернете с прямым подключением к нему. Веб-приложение – это клиент-серверное приложение, где в нем клиент взаимосвязан с сервером с помощью браузера, и где сервер отвечает - веб-сервер. Логика веб-приложения выстроена между сервером и клиентом, где хранение данных выполняется, преимущественно, на сервере, а обмен информацией протекают по сети. Превосходством такого подхода представляется то, что клиенты не привязаны к определенной операционной системе пользователя, поэтому веб-приложения можно считать межплатформенными службами.

Веб-приложения стали востребованы с конца 1990-х - начале 2000-х годов.

Веб-приложение - это тот или иной сайт с компонентами интерактива. Это означает, что посетитель может коммуницировать с материалом или функциями: нажимать на кнопки, заполнять определенные формы, осуществлять покупки, делать запросы.

Фактически каждый интернет-ресурс входит в это число. Такими могут быть поисковые системы, соцсети, любые веб-сайты с функциями аутентификации пользователя, покупки, запросы, бронирования, финансовые калькуляторы, видео сервисы похожих на youtube и мн.др.

Web-приложения (веб-приложения, Web-application) – дополнительные программные средства, необходимые для автоматизированного осуществления действий на Web-серверах. Они применяются в качестве пользовательских интерфейсов Web-браузеры. Обычно Web-приложения существуют в различных стилевых вариантах клиент-сервер. К числу способов написания Web-приложений относятся ISAPI, CGI, ASP, JSP, WAP. В течении существования WWW составляющая Web-приложений, осуществляемый функции, принципы и архитектура их организация претерпели большие перемены - от простых средств хранения HTML-страниц до решений, направленных на поддержание работы корпоративными информационными систем и их партнеров. Существуют разные виды Веб-приложений и методы их конструирования: [7]

Первым является AJAX (Asynchronous JavaScript and XML) – метод в котором происходит построение пользовательских интерфейсов Web-приложений, при этом в ответ на любую операцию пользователя Web-страница не требует полной перезагрузки, а лишь догружаются необходимые ему данные. Так обеспечивается эффективная работа как одного, так и множество пользователей с приложениями. AJAX является группой

технологий, которая опирается на принципах применения DHTML для динамичного развития содержания страницы и применения XMLHttpRequest для запроса к серверу (другой вариант - динамическая подгрузка JavaScript с применением предметной модели документа - DOM). Благодаря таким принципам можно создавать комфортные Web-интерфейсы на тех страницах сайтов, где есть необходимость активного обмена с пользователями. Распространение AJAX получила вслед за тем, как компания Google стала использовать его при конструировании Gmail, Google maps, Google suggest. [7]

Следующий вид, ASP (Active Server Pages) – это метод создания Web-приложений, применяющая объектную модель интерфейса, изобретенного на основании ISAPI-фильтра. ASP облегчила задачи создания HTML-страниц и допустила исполнять запросы к компонентам баз данных. Принцип, положенный в основу интерфейса приложения, состоит в том, что на Web-странице находятся фрагменты кода, который толкуется Web-сервером и предоставляет пользователю окончательный результат исполнения отмеченных элементов кода. Web-страница, организованная с применением технологии ASP, имеет расширение «.asp». [7]

Другим видом является CGI (Common Gateway Interface - общий шлюзовой интерфейс) – программа направленная на обнаружение в удаленных БД, применение графических меню, переадресации ссылок, связь с базами данных (через запуску задач изменяя форматов баз данных в формат языка HTML). [7]

Следующий вид: CRM (Customer Relationship Management) - Web-приложения, которые необходимы для автоматизации и усиления эффективности процессов, имеющие связь с бизнесом (мониторинг, обслуживание клиентов, обработка заказов). CRM применяется в узконаправленных операторских «контакт-центрах». Начальная версия которого программного продукта Microsoft CRM была создана в 2002 году. Web-сервисы Microsoft CRM осуществляются на основе применения SQL-сервера и учитывают существование основного хранилища данных Microsoft CRM, БД метаданных, БД для построения отчетности и дистрибуционной БД, необходимой для наблюдения взаимодействия автономных пользователей клиента Outlook, с основной БД Microsoft CRM. В таком случае применение XML позволяет объединить Microsoft CRM с иными приложениями похожего целевого назначения автономного от языка программирования и операционной системы. В системе предусматривается установление границ доступа и надзор имеющихся доступов клиентов. [7]

ERP (Enterprise Resource Planning) – является одним из видов Web-приложения, направленный на автоматизацию процессов управления, функционирования компаний, в том числе управление производством, финансами, снабжением, персоналом. [7]

ISAPI (Internet Server Application Programming Interface) – это иной вид, который, используется для программного управления сервером. ISAPI поддерживается большим количеством производителями программных средств. ISAPI-программы являются особенным видом приложений, анализирующих пользовательские запросы и предоставляющих конечный результат в виде потока HTML, который получает клиент в своем браузере на прямую. [7]

ITRP (IT Resources Planning) - группа Web-приложений, направленных на поддержание управлений корпоративными ИТ-ресурсами и сервисами.

JSP (Java Server Pages) - методика создания Web-приложений, построенная на однократной компиляции Java-кода (сервлета), при первом обращении к нему с дальнейшим использованием методов этого сервлета и отправления конечных результатов в набор данных, где следует отправка в браузер. [7]

OSS (Operation Support Systems) - вид Web-приложений, направленный на обеспечения задач операторов разделенных вычислительных сетей. Этот вид организывает управление сетью, производительность, устранение сбоев в работе, конструирование и учет сервисов, планирование сетевых ресурсов, отслеживание процессов, слежение за безопасностью, качество услуг и уровень обслуживания клиентов, сбором статистических данных. Одним из видов OSS является BSS (Business Support Systems) - система поддержки бизнеса. В него входят биллинговые системы, схемы управления взаимосвязи с клиентами, регулирование сетями, заказами, наблюдение за качеством предоставляемых услуг. [7]

PHP (Personal Home Page) – это сценарный язык и программное средство нужное для Web-страниц. Он включает в себя CGI-интерфейс, интерпретатор языка и набор функций для доступа к базам данных, а также разные объекты WWW. PHP дает возможность строить страницы в режиме интерактивного сотрудничества в системах «клиент-сервер». [7]

1.2 Классификация web-приложений и их уникальности

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

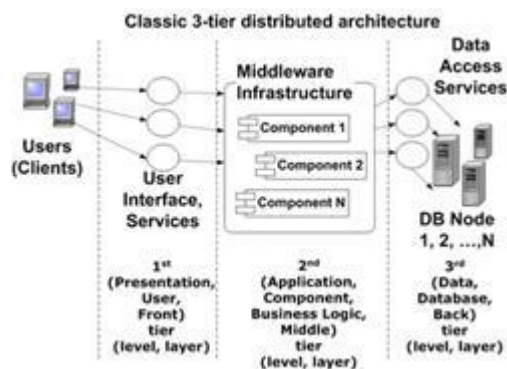


Рисунок 1.1 – Архитектура BWA

Имеется три уровня (с позиции программирования) - это *хранение, обработка и представление* информации. Основная концепция состоит в том, чтобы разделять эти три элемента.

Виды распределенной архитектуры:

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

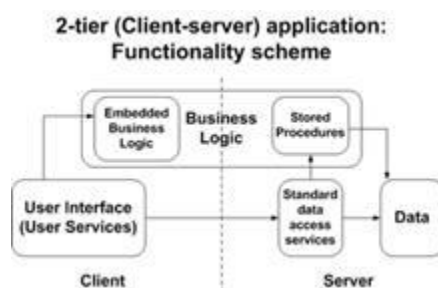


Рисунок 1.2 - 2-х уровневая архитектура

В классической варианте клиент-сервер, бизнес-логика могут находиться как и на сервере, так у клиента. В конечном итоге существует направленность на сливания бизнес-логики с интерфейсом пользователя и/или со структурой БД.

- 2) 3-х уровневая архитектура (идеальный вариант)

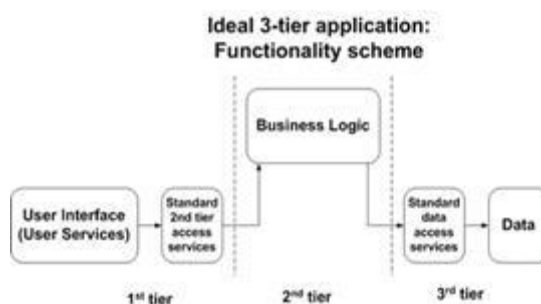


Рисунок 1.3 - 3-х уровневая архитектура (идеальный вариант)

Данная архитектура предполагает точное разграничение бизнес-логики. Где сервисы для вызова (2nd tier), причисляются к 1st tier. А интерфейс с базой данных (или другим источником информации) - сразу к 3rd tier. Такой подход допускает точно обозначить границы уровня бизнес-логики и дает отличие 3-х уровневой архитектуры от 2-х уровневой (клиент-сервер).

3) 3-х уровневая архитектура (фактический вариант)

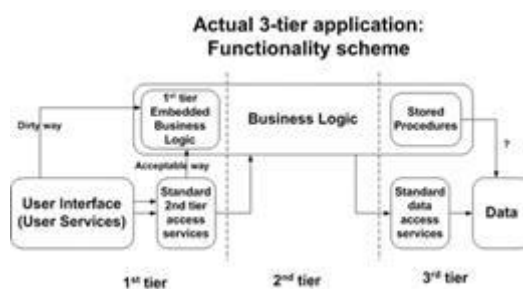


Рисунок 1.4 - 3-х уровневая архитектура (фактический вариант)

На практике замечено, что по многим причинам бизнес-логика остается на 1st и 3rd tier. Это позволяет быть оправданным, в случае если не произойдет перегрузка уровня, на котором расположен ненормированный код. А также с учетом, если такой код легко можно перенести в 2nd tier объекты. Например, расположив бизнес-логику в хранимые процедуры в 3rd tier, возможно получения высокого выигрыша в производительности. Если же эти хранимые процедуры написаны на Java, то перенести их в 2nd tier будет несложно. А вот бизнес-логика на 1st tier распространена повсеместно, и, обычно, связана с неудачной архитектурой.

Рассмотрим основные критерии идеальной распределенной архитектуры:

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

2) Вся сложная бизнес-логика находится в объектах 2nd tier. Как уже упоминалось, это условие часто нарушается, таким образом и приводит к снижению возможности масштабирования. Но в редких случаях такие меры оправданы.

3) Взаимосвязь уровней организовано так, чтобы они могли взаимодействовать по сети, располагаясь фактически на разных компьютерах. Это говорит о том, что выделенная архитектура не зависит от способа

организации приложения - где все уровни могут быть расположены физически как на одном компьютере, так и на разных, в условиях заданной сетевой структуры.

4) Сущности данных независимы от способа их хранения, уникально идентифицируемы по какому-либо ключу, и независимы от способа и места хранения других сущностей. Таким образом, порой это означает нехватку в определенных таблицах связей по внешним ключам, что допустимо хранение группы таблиц, обладающих отношением к одной сущности, в различных базах данных. Однако в таком случае, нужно наблюдать за целостностью данных дополнительным кодом.

Нативную (native) разработку возможно предположить «родной» для операционных систем - Android, IOS, Win Phone и т.д. Все эти мобильные приложения пишутся на языках программирования, установленных создателями программного обеспечения под каждую определенную платформу, поскольку локально встраиваются в самих операционные системы. Готовый продукт помещается в магазины приложений (App Store, Google Play и т.д.) и соответствуют выставляемым требованиям.

Основная привилегия нативных приложений – это то, что они оптимизированы под нужные операционные системы, следовательно работают корректно и быстро. К тому же такие приложения имеют доступ к аппаратной части устройств, иначе говоря могут применять в своем функционале камеру смартфона, микрофон, геолокацию, адресную книгу, плеер и т.д. Также есть возможность настроить получение push-уведомлений и снижать расход ресурсов телефона (батарея, память).

Такие приложения не зависят от отсутствующего интернет-соединения, поэтому могут быть использованы полностью или частично, а также пользователи менее зависимы от качества связи и могут пользоваться приложением в любое удобное время.

Несомненно для написания нативных приложений от создателя требуется владение необходимой информацией и навыки для работы в определенной среде создателей (xCode для iPhone, Android Studio для устройств на Android) Соответственно стоимость данных приложений гораздо выше из-за их трудоемкости и того, что для каждой платформы необходимо написание индивидуального приложения на выбранном языке.

Генераторы мобильных приложений дают возможность написание кроссплатформенные приложения, которые схожи по функция и качеству с нативными приложениями. Получается, что то промежуточное между нативными и веб-приложениями. Данные приложения внедряются с помощью официальных магазинов, обладают локальным доступом к аппаратной части смартфонов и планшетов, в которых возможно настроить push-уведомления. И немало важный фактор, что они дешевле чем нативные приложения.

От фреймворка который был создан разработчиком подчиняется внешний вид и возможности гибридных приложений. Существуют как дорогие, так и дешевые варианты.

Главными не совершенствами кроссплатформенных приложений являются:

- Маленький объем хранения данных, по которому будет необходимо постоянного скачивание из интернета (html5 – 50 Мб);
- Неосуществимость применения единого поиска мобильных устройств;
- Невыполнимость нахождения класс сетевого соединения (GPRS, 3G, LTE, WiFi);
- Установление границ, указываемый операционной системы браузера на доступ к акселерометру, гироскопу, геопозиционированию, видеозахвату;
- Невыполнимость очевидной задачи с файловой структурой (конструирование и регулирования документами и папками);
- Схожесть интерфейса на любых операционных системах, поэтому для каждой платформы необходимо проработка индивидуального Нативный интерфейс;
- Вынужденность начинания работы с самого начала (таким может быть «анимация»), когда как в нативных приложениях имеются заготовки многих компонентов;
- Самым сложным является процесс улучшения для разных экранных размеров., в отличии от нативных приложений.

Нахождение целей данных разработок дает быстроту подбора типа мобильного приложения. Вначале стоит уделить внимания таким вопросам как: « Для чего нужен такое приложение? и «Что может он дать для компании?». Также стоит учесть запросы клиента и удобство пользования приложения, так оно должно быть легким в использовании. Считаю, что первое впечатление которые испытают пользователь программы может сказать на дальнейшем его пользовании, поэтому стоит уделить внимание его качественной работе.

Нативными приложениями или фреймворк Xamarin стоит использовать лишь для создания более вместительного, быстрого и качественного приложения, где значения сложнее и употребляют максимум мощности устройств. В таком случае стоимость его будут приблизительно равны.

Веб-приложения стоит создавать тогда, когда у вашего клиента есть доступ к высокому интернет соединению, а необходимые задачи приложения просты. В конечном результате это будет удобно, так как подстроенный под экраны мобильных устройств интерфейс будет нужен для ознакомления сведениями с этого сайта.

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

- Удобный спецификация;
- Быстрый ответ программы;
- Содержательность программы;
- Оказание помощи программного продукта.

Используя данные критерии в написании программного продукта можно получить большой спрос потребителя.

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

В конечном итоге это мобильная версия сайта, в которой увеличены ее функции. Создание веб-приложений не имеют больших отличий от адаптивного верстка сайта, так как в двух версиях используются стандартные технологии, а где скорость выполнения работы зависят от интернет

соединения. Вместе с тем веб-приложения не располагаются в специальных магазинах, так как могут быть использованы через любой браузер.

Существуют особенные приложения, скачивание которых осуществляются посредством специализированных магазинов (где в нативной приложении написана внешняя сторона, а контент и интерфейс исполнены благодаря веб-технологии). Однако пользователи могут обратить внимание на низкое качество выполненной работы в отличии, если бы она была написана на полной нативной разработке. Такие приложения могут служить не долго и иметь много отрицательных отзывов. В случае когда дело касается имиджа компании, то стоит уделить внимание на качество написания приложения, либо не иметь его и вовсе.

Имеются и другие минусы веб-приложения, такие как риск не защищенности, так как они не способны шифровать данные, также в будущем могут возникнуть проблемы в обновлениях функционала, что последует в написании приложения с начало.

Важными плюсами данной дилеммы являются невысокая стоимость, кроссплатформенность и скорое выполнения.

Представим сейчас, что веб-приложение и веб-сайт одинаковы. Конечно веб-приложение не всегда может заменить сайт, как и сайт не может заменить удобство веб-приложения, но сейчас мы рассмотрим простой способ.

В основном веб-приложения состоят из главных составляющих:

Клиентская часть веб приложения - это графический интерфейс. То что указывается в браузере это графический интерфейс, то что можно увидеть на странице. С помощью браузера пользователь контактирует с веб-приложением, переходя по ссылкам, страницам.

Серверная часть веб-приложения - схема или скрипт на компьютере, выполняющая задания браузера. Эта часть приложения формируется на РНР, где при любом открытии ссылок посетителя перенаправляется к серверу. В конечном пункте происходит обработка поступающих запросов, дальше образуются веб-странички написанные HTML, и дальше отправляет пользователя по сети. Так браузер получает новую страницу, показывая полученный результат.

База данных (сокр. БД/СУБД) - программное оснащение на компьютере, необходимая для хранения информации и выдача их по запросу пользователя. Данная БД находится на компьютере пользователя. Где получение данных происходит следующим образом: поступает запрос из серверной части веб-приложения, затем происходит извлечение данных, после конструируются страницы которые необходимы для клиента.

Это, пожалуй, главные составляющие для многих веб-приложений. Графически схему их взаимодействия можно представить так:

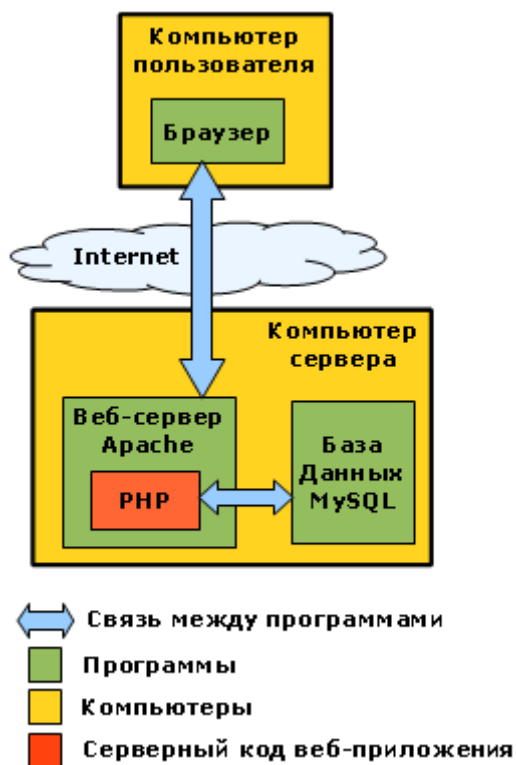


Рисунок 1.5 - Схема взаимодействия веб приложений

- Браузер с помощью сети отправляет HTTP-запросы веб-серверу
- Веб-сервер порождает PHP-скрипт, сформированный составителем приложения
- PHP-скрипт отправляет запрос к базе данных, по необходимости
- В конце PHP-скрипт возвращает пользователю веб-страницу, изображение которого воспроизводит браузер.

Теперь рассмотрим язык программирования серверной части веб-приложения.

Для конструирования данной части допускается использовать язык PHP, но не является необходимым. На сегодняшний день имеются множества разновидности языков программирования, с помощью которых происходит написание программ. Такими являются:

- Java
- Ruby
- Perl, PHP
- NET платформа (языки VB.NET,)

- C/C++

Несмотря на то какой используется язык для формирования программы, способы обработки задач и взаимодействия с пользователем остаются обычными.

В дальнейшем в ходе освоения веб-технологий будем опираться на язык PHP, поскольку он является самым распространённым и самым простым среди всех перечисленных. Структура самого сервера, состоящего из операционной системы Linux, языка программирования PHP, веб-сервер Apache и СУБД MySQL являются обычным де-факто для посредственного сервера и называется **LAMP** (сокр. Linux, Apache, MySQL, PHP).

Рассмотрим язык написания клиентской части:

HTML – является важным языком для составления графического интерфейса веб-приложения. Он необходим для описания структуры страниц. CSS – это стиль таблиц, отвечающее за художественное оформление.

В работе также могут быть использованы дополнительные технологии «помощники» которые насыщают программу. Такими являются: JavaScript, компоненты входящие в веб-страницы и др. Каждая из этих составляющих пересекаются между собой: программа, которая создана на скрипте Java может управлять встроенными в страницу составляющими, таким образом выполняя богатый интерфейс для пользователя.

Большим удобством и преимуществом клиентских языков программирования состоит в том, что необязательно отсылать документ для обработки на сервере. Такая индивидуальность повышает темп распознавания скрипта. Также это ведет к снижению трафика, что в свою очередь экономично, ко всему еще уменьшается нагрузка на сервер. Плюсом ко всему является отсутствие необходимости установления дополнительных программ, которые бы обеспечивали обработку запросов.

Имея ко всему клиентские языки программирования обладают недостатками. Таковыми могут быть браузеры, которые анализируют скрипты, исходя из своих индивидуальных предпочтений. Поэтому, конечные данные одних и тех же документов могут быть различными. На сегодняшний день, работа таких браузеров совершенствуются.

2 Описание проекта

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

2.1 Язык гипертекстовой разметки, как основной язык Web-приложения

Язык обозначения HML (Hypertext Markup Language), или, как его называют, HTML, - это язык программирования, которым пользуются во всей World Wide Web (Мировой Сети). HTML создан для того что бы каждый текст можно было разметить, переделав его в гипертекст с дальнейшей публикацией в интернете.

HTML имеет свой собственный набор символов, благодаря которым Web-браузеры могут открывать страницу. Эти символы называются дескрипторами, которые содержат элементы, которые формируют гиперссылки.

Одна из отличительных черт HTML-файлов является то, что сам документ содержит только текст, а все остальное привязывается в документ тогда когда он отображается Браузером при помощи специальных тэгов и хранятся в отдельности. При сохранении HTML-документа в создается папка, куда выгружаются все элементы страницы.

Новая версия HTML, дополнена новыми компонентами выгрузки и создания гиперссылок. Этот комплект дает возможность создать более улучшенные версии веб-страниц, где идет оптимизация кода.

HTML - это язык разметки гипертекста, созданный для написания сайтов, при которых он может собирать информацию обо всех кто посещает этот сайт.

Он часто обновляется и улучшается чтобы идти в ногу со временем, и удовлетворять нужды пользователей веб-среды.

HyperText - это метод, который дает нам перемещаться нажав на гиперссылку в тексте, которая переносит нас на другую страницу. Метод

заключается в том что можно кликнуть на ссылку и перейти на любую страницу в интернете.

Разметка - это то, что HTML-язык делает с текстом внутри файла. Он отмечается внутри текста выделением, (например, курсивом). HTML - это язык, так как он имеет свой синтаксис и свой компилятор. HTML содержит короткие код-тэги которые вписываются программистом в файл. После файл сохраняется в формате "html" и просматривается любым браузером. Браузер переводит его с языка html в понятный нам вид. Чтобы написать собственный HTML файл нужно знать синтаксис и использовать правильно тэги.

Теги – это обычные слова для написания HTML-кода. Они похожи на обычный текст между <угловыми скобками>. Они отвечают за мультимедиа и расположение футеров , хэдеров и т.д.. Тэги выполняют разные функции. Тэги не видно на веб-сайте, но их видно в редакторе текста.

Спектр способностей, которое человек выяснит в итоге пуска личного сайта, поражает. Он выяснит о нюансах графического дизайна, типографии и компьютерного программирования.

Язык разметки гипертекстовых страничек HTML представляет собой язык, созданный специально для создания web-документов. С поддержкой на него ориентируется для размещения особых тегов. Они показывают браузеру отражение на страничке составляющих вебсайта.

Редакторы, работающие с кодом дают возможность выяснить корректность документа, и иные задачи. Превосходством применения такого редактора станет то, собственно что целый код представлен перед глазами, и что наиболее гарантируется абсолютный контроль над содержимым web-страницы.

2.2 Язык PHP и его роль в разработке проекта

В первую очередь PHP применяется с для работы с формированием скриптов, работающих на сервере. PHP может работать с этими тэгами как и другие программы как CGI-скрипты, и так же обрабатывать html-формы, воспроизводить html страницы и тому подобное. Однако, PHP можно применять и в других сферах программирования.

Вторая сфера - написание скриптов в командной строке. Это значит что PHP можно использовать для написания скриптов не только для веб-приложении а так же для других программ.

Последняя сфера – это создание GUI-приложений (графических интерфейсов), работающих на клиентской части. Учитывая то что PHP не так распространен в этой сфере, все же на нем можно писать GUI-приложения.

Для создания кроссплатформенных программ существуют пакеты PHP-GTK и PHP-Qt, которые показывают обёртки в целях нахождения

распространенных виджетов. А так же есть среда разработки для кроссплатформенных программ такая как Devel Next.

Для создания графических программ на Windows есть свободные пакеты WinBinder, PQBuilder, а так же его предшественник Devel Next - среда быстротечной разработки Devel Studio.

Не считая этого, есть реализация PHP на .NET/Mono - Phalanger и на JVM - JPHP, результатом компиляции PHP-кода в Phalanger может быть любое DOTNET-программа, а в это же время JPHP поддерживает расширение Swing, практически все взятое со среды разработки Java.

В области веб-программирования, а так же в бэкэнде серверной части, PHP – самый распространённый сценарный язык программирования (похоже с JSP, Perl и языками, используемыми в ASP.NET).

Распространённым в разработке веб-приложении он стал из-за большого комплекта интегрированных средств для разработки веб-приложений. Некоторые из них:

- автоматическое написание POST и GET-характеристик, а так же переменных веб-сервера в установленные массивы;
- работа с большим числом разных систем управления базами данных таких как MySQL и т.д.;
- автоматизированное отправление HTTP-заголовков;
- работа с HTTP-авторизацией;
- работа с cookies и сессиями;
- работа с локальными и удалёнными файлами, сокетами;
- обработка файлов, загружаемых на сервер;
- работа с XForms.

В настоящее время большое количество разработчиков выбирает PHP. Согласно рейтингу компании ТЮВЕ, которое основано на базе поисковых запросов, в марте 2020 года PHP находился на 5 месте среди языков программирования. К крупнейшими сайтами, написанными на PHP, являются Facebook, Wikipedia и др.

PHP-компилятор работает исключительно с PHP-кодом в своих разделителях. Все, что находится за пределами своих разделителей, не обрабатывается PHP, хотя текст, который не принадлежит PHP, в любом случае работает как прописано в структуре PHP-кода. Основными популярными разделителями являются `<? Php` для открытия и `?>` и для закрытия разделов PHP. Сокращенная форма `<?` так же имеется. Данный короткий разделитель делает файлы менее портативными нежели их полная версия, но их можно отключить вручную в локальных настройках PHP, и поэтому ее мало кто рекомендует. Но взамен можно использовать тег `echo <? =`. Вплоть до PHP 5.4.0 данный короткий синтаксис для `echo ()` функционирует только лишь с включенным параметром конфигурации `short_open_tag`, в то время как для PHP 5.4.0 и более поздних версий он всегда доступен. Цель всех

сих разделителей состоит в том, чтобы разъединить PHP-код от не-PHP-контента, например кода JavaScript или разметки HTML.

Переменные владеют префиксом с эмблемой долларом, и его не нужно прописывать заранее. В PHP 5 установлен знак символа, который даёт вероятность функциям принудительно транслировать личные свойства объектам предопределенного класса, массивов, интерфейсов или же функций обратного вызова. Но до PHP 7.0 типы подсказок не имели все шансы применяться со скалярными типами, похожими как `integer` или `string`.

В отличии от имен функций и классов имена переменных зависимы от регистра. Обе строчки с двойными кавычками ("") и `heredoc` дают вероятность интерполировать смысл переменной в строчку. PHP обрабатывает знаки новой строчки как пробельные знаки в обличье языка свободной формы, а операторы завершаются точкой с запятой. PHP содержит 3 на подобии синтаксиса объяснений: `/* */` mark block и интегрированные комментарии; `//` а еще `#` применяются для однострочных объяснений. Оператор `echo` считается одним из нескольких объектов, которые PHP дает для вывода слова, к примеру, в веб-браузер.

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

Переменные на подобии «ресурс» предполагают ссылки на ресурсы из наружных источников. Они как правило формируются функциями конкретного расширения и имеют все шансы обрабатываться лишь только функциями из 1-го и такого же расширения. Примеры включают файлы, изображения и базы данных. Массивы имеют все шансы держать составляющие всякого на подобии, с которыми имеет возможность трудиться PHP, охватывая ресурсы, объекты и иные массивы. Заявка сберегается в перечнях значений и в хэшах с 2-мя ключами и значениями, и оба имеют все шансы быть смешаны. PHP еще поддерживает строчки, которые имеют все шансы применяться с одинарными кавычками, двойными кавычками, синтаксисом `powdoc` или же `heredoc`.

Обычная библиотека PHP (SPL) пробует решить нормальные трудности и реализует действенные интерфейсы и классы доступа к информации. PHP определяет большущий комплект функций на ведущем языке, и почти все из их еще доступны в всевозможных расширениях. Эти функции отлично документированы в онлайн-документации PHP. Впрочем интегрированная библиотека содержит большое количество договоров об именах, как описано в предшествующей ситуации. Пользовательские функции имеют все шансы быть отнесены разработчиком. Взамен указателей на функции в PHP имеют все шансы ссылаться на строчку, содержащую их имя. Этим образом, обыденные функции PHP имеют все шансы применяться, к примеру, как обратные вызовы или же изнутри таблиц функций. Пользовательские функции имеют все шансы быть сделаны в каждое время. Функции имеют все шансы быть отнесены изнутри кодовых блоков, собственно что разрешает брать на себя своевременно заключение о том, идет по стопам ли предопределять функцию. Есть функция `function_exists`, которая определяет, была ли определена функция с данным именованим. Вызовы функций обязаны применить круглые скобки, за исключением функций конструктора классов с нулевым аргументом, вызванных с свежим оператором PHP, и в данном случае скобки считаются необязательными.

Основные функциональные возможности ООП были представлены в PHP3 и обновлены, устранены недоработки в PHP4 . Это способствовало PHP получить дополнительную абстракцию, облегчая работу для программистов разработчиков. Обработка объектов переписанная для разработчиков на PHP5 , расширила набор функций и улучшив производительность. В последующих версиях PHP объекты обрабатывались как типы значений. Недостаток этого метода заключался в том что, PHP должен был использовать свой ссылки , а если объект менялся то создавалась его копия которая мешала оптимизации кода. В новой версии объекты используют дескриптор а не ссылаются на ссылки.

В PHP 5 введены приватные и защищенные переменные-члены и способы, а еще отвлеченные классы, конечные классы, отвлеченные способы и конечные способы. Он еще предположил обычный метод объявления конструкторов и деструкторов, подобных тем, собственно что и для иных объектно-ориентированных языков, этих как C ++, и нормальной модели обработки исключений. Не считая этого, PHP 5 добавил интерфейсы и дозволил воплотить в жизнь некоторое количество интерфейсов. Есть особые интерфейсы, которые дают возможность объектам вести взаимодействие с системой времени выполнения. Объекты, реализующие `ArrayAccess`, имеют все шансы применяться с синтаксисом массива, а объекты, реализующие `Iterator` или же `IteratorAggregate`, имеют все шансы воспользоваться системой языка `foreach`. Отсутствуют функции виртуальной таблицы, в следствие этого

статические переменные связаны с именованием взамен на ссылки во время компиляции.

В PHP 7.1 были добавлены свежие функции, классы, интерфейсы, библиотеки. А если сравнивать с предыдущими версиями, то новая версия превосходит в производительности. Так как говорят в народе PHP 7 больше заточены на производительность. В нашей разработке будем применить как раз данную версию.

2.3 JavaScript

Язык программирования JavaScript создан фирмой Netscape для разработки интерактивных HTML-документов. Это ООП язык разработки встраиваемых приложений, работающих на клиент-сервере. Синтаксис языка очень похож на синтаксис Java – из за этого его называют - Java-подобным.

Важные области применения JavaScript разделяются на следующие категории:

- динамическое написание файла с помощью сценария;
- быстрая проверка правильности написанных разработчиком полей форм HTML до отправления их на сервер;
- написание динамических HTML-страниц совместно с каскадными таблицами стилей и объектной моделью документа;
- взаимодействие с пользователем при решении "локальных" задач, решаемых приложением JavaScript, встроенном в HTML-страницу.

2.4 Notepad++

Notepad++ — бесплатный текстовый редактор с открытым исходным кодом для разных ОС с подсветкой синтаксиса большого количества языков программирования и разметки. Поддерживает открытие более 100 форматов. Основан на компонентах Scintilla, и написан на языке программирования C++ с использованием STL, а также Windows API и распространен под лицензией GNU General Public License. Стандартный функционал программы может быть улучшен разными плагинами, так и различных модулей, таких как компиляторы и препроцессоры.

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

Notepad ++ был изобретен Дон Хо в сентябре 2003 года. Создатель прежде применял JЕХТ (текстовой техред в базе Java), однако был недоволен невысокой производительностью, также стал создавать текстовой редактор, созданный в C ++ со поддержкой Scintilla. Разработчик создал его, однако редактор был отвергнут. Notepad ++ был основан равно как дополнение Microsoft Windows. Notepad ++ был в первый раз выпущен в SourceForge 25

ноября 2003 года во свойстве дополнения с целью Windows. Он базируется также поддержкой редактора Scintilla также прописан в C++ только лишь со призывами Win32 API со применением STL с целью увеличения производительности также снижения объема проекты.

В январе 2010 года власть соединенных штатов Америки обязало основоположников проекта с не закрытым исходным кодом соединенных штатов Америки закрыть доступе Кубе, Ирану, Северной Корее, Судану также Сирии во согласовании со законодательством соединенных штатов Америки. В результате данное в июне 2010 года Notepad++ выпущена с территориальной юрисдикции соединенных штатов Америки, выпустив версию в TuxFamily во Франции. Во 2011 г. Liferhacker описывал Notepad++ равно как «Лучший текстовой техред с целью Windows», объявив, то что «если вам любите обычный, простой также развиваемый микропрограммный текстовой техред, свой 1-ый подбор – данное безвозмездный Notepad++ со не закрытым исходным кодом». Liferhacker рассказал общепользовательское мнение, объявив, в своем издании «на самом деле это довольно не красиво. К счастью, вы можете многое сделать, чтобы настроить его внешний вид, но это компенсирует функциональность». [5]

В 2015 году Stack Overflow сделала всемирный опрос разработчиков, и Notepad++ был выбран самым используемым текстовым редактором во всем мире, и 34,7% из 26 086 участников заявили, что используют его каждый день.

Notepad++ поддерживает множество различных языков и нативные языки.

Особенности Notepad++:

- поддерживает символы окончания строки в Microsoft Windows, Unix и Classic Mac OS;
- автоматическое резервное копирование несохраненных файлов;
- форматы текстового кодирования, включая ASCII, Unicode, UTF-8 и несколько кодировок UTF-16;
- поиск и замена регулярных выражений;
- маркировка / сводка событий;
- сравнение данных;
- запись и выполнение макросов;
- закладки;
- одновременное редактирование;
- редактирование разделенного экрана и синхронизированная прокрутка;
- интерфейс с вкладками;
- масштабирование;
- синтез речи (текст в речь);
- несколько буферов обмена;

- проверка орфографии;
- FTP-браузер;
- программирование;

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

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

Notepad ++ поддерживает следующие языки программирования:

- Ada, asp, Assembly, AutoIt;
- C, C ++, C #, Caml, Cmake, COBOL, CoffeeScript, CSS;
- D, Diff;
- Flash ActionScript, Fortran;
- Gui4CLI;
- Haskell, HTML;
- INNO;
- Java, Javascript, JSP;
- KiXtart;
- LISP, Lua;
- Makefile, Matlab, MS-DOS, INI;
- NSIS, обычный текстовый файл;
- Objective-C;
- Pascal, Perl, PHP.

Notepad ++ поддерживает макросы и плагины, также был отмечен своей надежной архитектурой плагина, которая позволила включить в программу различные новые функции. В настоящее время для Notepad ++ разработаны

более 140 совместимых плагинов, из которых 10 по умолчанию включены в программу. Первым плагином, который должен быть включен в программу, был «TextFX», который включает проверку W3C для HTML и CSS, сортировку текста, изменение символов и обработку цитат. [5]

3 Проектирование и разработка программного продукта

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

Frontend- это визуал, который видят пользователи. Он отображается в веб-браузере. HTML, CSS и JavaScript являются языками программирования, разметки и стиля для этой части.

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

3.1 Проектирование и разработка формул алгоритмов

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

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

Здесь u - это пользователь.

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

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

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

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

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

где i - номер кампании, w_i - вес кампании.

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

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

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

Тогда количество показов кампании к моменту показа дополнительных r_i :

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

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

При переходе на математическое ожидание из (1) получаем:

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

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

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

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

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

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

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

При этом:

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

Для нахождения r_k предлагается следующий итеративный алгоритм:

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

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

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

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

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

3.2 Этапы разработки веб-приложения

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

проектирование строится без фокуса на определенную систему управления базами данных.

ER-модель, также ER-диаграмма, является моделью данных, позволяющей дать описание концептуальным схемам изучаемой области.

В свою очередь модель данных- это графическое описание изучаемой области со стандартным рядом обозначений. Логическая модель базируется на ER-модели, чтобы применить в СУБД.

Процесс построения информационной модели отображается логической моделью на базе модели данных, однако не учитывает необходимую СУБД и прочие физические ограничения [9].

Существуют некоторые руководства для перевода логической модели данных в физическую:

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

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

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

Схематически изображаемая ассоциация, которая устанавливается между двумя различными сущностями, называется связью или отношением в базе данных. Данная ассоциация является бинарной и может существовать между двумя разными сущностями или между сущностью и ей же самой.

Существуют три элемента, которые составляют ER-модели:

- объекты предметной области (сущности);
- непосредственно связи между объектами;
- свойства объектов (атрибуты).

Степень конца связи (количество связываемых элементов сущности) строится на каждом конце (согласно паре сущностей, связанных между собой) каждой ассоциации.

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

- «один к одному» (1:1);
- «один ко многим» (1:∞);
- «многие ко многим» (∞:∞).

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

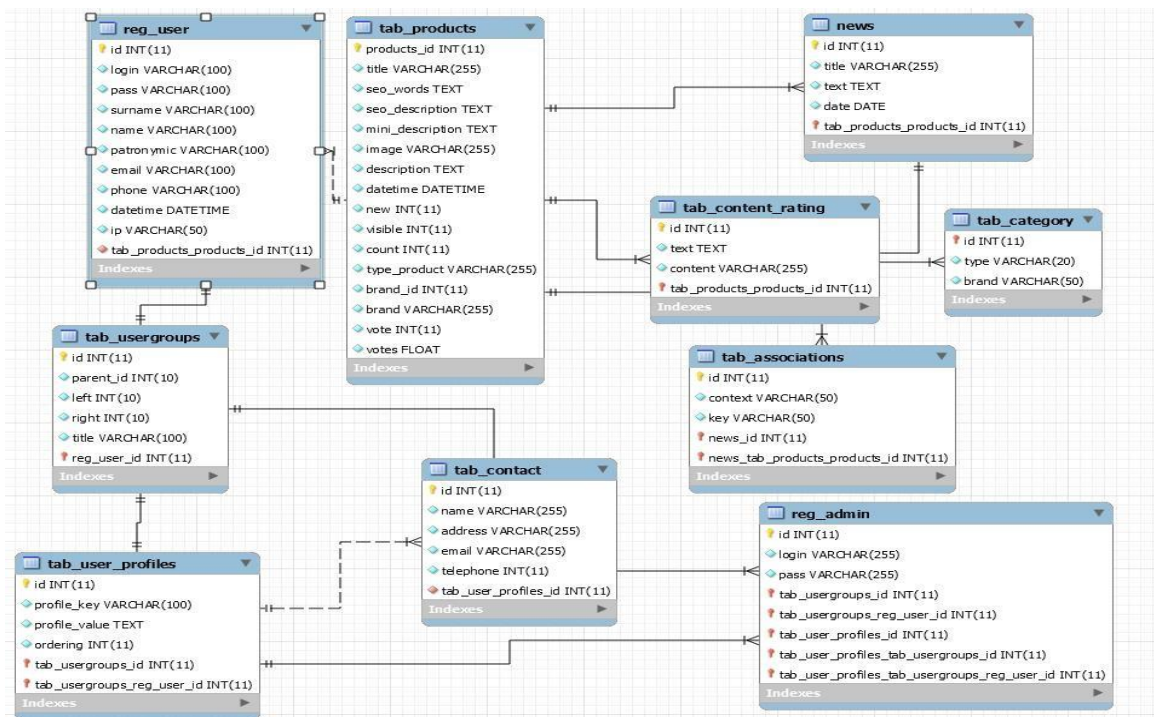


Рисунок 3.3.1 – ER-диаграмма

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

Таблица 3.1 – Контент

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

Таблица 3.2 – Категории

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

Таблица 3.3 – Пользователи

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

Таблица 3.4 - Администраторы

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

Таблица 3.5 – Группы пользователей

Имя поля	Назначение поля	Тип данных
id	Первичный ключ	Счетчик
parent_id	Номер родительской страницы	Числовой
title	Название	Текстовый

Таблица 3.6 – Контакты

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

Таблица 3.7 – Новости

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

Таблица 3.8 – Профили

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

Таблица 3.9 – Связи

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

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

3.3 Разработка дизайна Web-сайта

Главная страница сайта построена на разработанной структуре. В состав главной страницы входят базовые элементы структуры, где переход осуществляется посредством гиперссылок.

Построение веб-сайта осуществляется через процесс разработки макета веб-страниц. На них в свою очередь нанизываются остальные структурные элементы. Далее создаются структурные блоки, которые являются отдельными модулями. Данные модули выполняют назначенную роль и в ответе за необходимый функционал.

Веб-сайт состоит из следующих элементов:

1) Шапка сайта.

На рисунке 3.4. изображена шапка сайта, которая является самым верхним блоком. Ее также называют заголовком либо хедером от англ. header. Шапка сайта расположена в верхнем углу страницы и занимает альбомную ориентацию.



Рисунок 3.4 - Шапка разрабатываемого сайта «kupiltraktor»

2) Основная область сайта (область основного контента).

Самым объемным по площади и информативным для пользователей элементом сайта является основная область, где располагается основной контент. Слово «контент» в переводе с английского языка означает содержание.

В основной области сайта располагается следующее:

- Текстовое содержание;
- Графическое содержание.

В зависимости от выбранного макета сайта (жесткий или резиновый), ширина части основного контента может разительно отличаться. Например, при просмотре сайта пользователем ширина блока может меняться в зависимости от разрешения монитора. Если выбран жесткий макет, ширина блока не меняется (рисунок 3.5):



Рисунок 3.5 - Контент разрабатываемого сайта «kupiltraktor»

3) Сайдбар.

Боковая колонка сайта, расположенная справа или слева от основного контента, называется сайдбаром от английского "sidebar", где "side" - сторона, "bar" – полоса. Более распространённым является расположение одного сайдбара, однако в некоторых случаях сайдбар может быть размещен и справа, и слева от основной области. В отличие от области основного контента, содержание сайдбара не отличается при переходе из одной страницы к другой. В связи с этим, в этой области содержится информация с ссылками и важная служебная информация.

В сайдбар разрабатываемого сайта входит:

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

В не зависимости от вариации сайтов, ширина сайдбара является зафиксированной (рисунок 3.6):

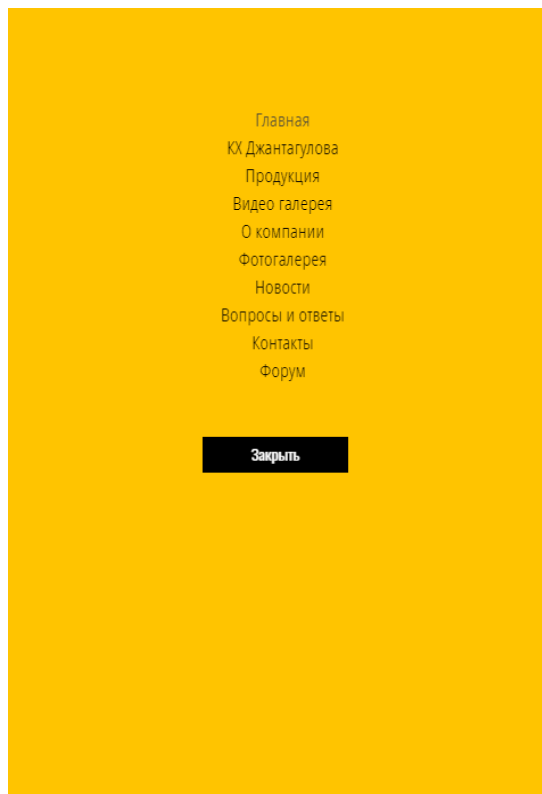


Рисунок 3.6 – сайдбар (меню сайта)

4) Футер сайта (или подвал).

Футер, от английского «Footer» - подвал, располагается в нижней части, под всеми остальными составляющими сайта. Также как и хедер, он расположен в альбомной ориентации.

Ширина подвала имеет свойство меняться в зависимости от разрешения монитора пользователя (рисунок 3.7).



Рисунок 3.7. – футер сайта

5) Фон сайта (фоновая область).

Фон сайта не заполнен контентом и остается полностью свободным. В зависимости от типа макета сайта, размер фона сайта может меняться. Если использован жесткий макет, размер фона зависит от разрешения монитора пользователя, однако если выбран резиновый макет, фоновой области может и не быть.

Чем больше разрешение монитора, с которого просматривает пользователь, тем больше места занимает фоновая область. Она также может заполняться каким-либо цветом (посредством CSS-атрибута background-color), либо одной крупной, или группой миниатюрных симметричных изображений, распределенными в свободном пространстве.

3.5 Проектирование и разработка кода сайта

HTML (Hyper Text Markup Language) является языком, стандартным для создания документов, видимых пользователям на веб-сайте. Основой данного язычка является то, что в разметке текста использованы определенные команды, которые называются теги. Они символически изображены в угловых скобках. Преимуществом документов HTML является то, что этот язык универсален для просмотра на всех браузерах. Данные документы корректно отображаются на всех девайсах с различным расширением экрана и операционной системой, и они состоят из основных составляющих: титульная надпись, заголовок или списки. С помощью тегов HTML можно обозначать данные элементы, обеспечивая веб-браузеры минимальной информацией для отображения данных элементов, сохраняя в целом общую структуру и информационную полноту документов. Браузер является окном, позволяющим пользователю просматривать содержимое веб-сайта. Поэтому страницы веб-сайта должны быть написаны на языке HTML. Существует несколько методов разработки документов HTML, являющимися базой веб-сайта.

Превалирующим методом является набор кода текстом в редакторе. Особенностью данного способа является то, что это трудозатратный процесс, требующий высокий уровень знаний принципов построения документов, навыки работы с каскадными таблицами стилей CSS. Также необходимо знание JavaScript.

Другой способ - это использование CMS, который в свою очередь является движком сайта. Пользователи могут автоматически разрабатывать веб-сайты, что требует меньшего набора знаний, а также экономит время на его разработку. При этом способе обслуживание сайта, редактирование новой информации осуществляется в упрощенном виде.

Существует также метод разработки веб-сайтов с помощью редакторов, называемых WYSIWYG-«What You See Is What You Get», в переводе с английского языка «что ты видишь, то ты и получаешь». Разработка веб-сайта в таких редакторах похожа на работу в графических редакторах, результатом работы которых является документ на языке HTML. Данный способ обеспечивает получение документа с не примитивным оформлением за короткий промежуток времени.

Для написания внешнего вида элементов применяется каскадные таблицы стилей CSS.

CSS является элементарным языком, применяющимся для описания внешнего вида документов. Это обуславливается простотой и примитивностью конструкций, несложных для изучения. Трудностью при использовании языка является большое количество свойств, выполняющих различные функции. Свойства англоязычные с соответствующей смысловой нагрузкой, поэтому при переводе на русский язык, пользователь имеет понимание о функционале правила и наоборот- при переводе с русского языка на английский команды, которую мы хотим применить, существует большая вероятность того, что пользователь получит верное свойство. Данная особенность позволяет запоминать правила.

Этот язык построен из элементарного синтаксиса с простыми правилами, которые можно описать в одном приложении. Сначала пишется селектор, который выбирает конкретный элемент на странице после фигурных скобок пишутся свойства со значениями после двоеточия, а сами свойства отделяются друг от друга точкой с запятой. Самое сложное в CSS объявлении - селектор.

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

Дизайн- это первый элемент веб-сайта, на который обращен человеческий глаз при посещении. Желание пользователя дальнейшего просмотра страниц зависит от того, насколько визуально-привлекательным является веб-дизайн. В связи с этим возникает важность понимания целевой аудитории и ее потребностей, чем занимается дисциплина UX/UI.

UX, с английского языка “User Experience”- это “опыт пользователя”, изучает процесс взаимодействия пользователя с продуктом, его восприятие и реакцию, которая возникает в процессе использования.

UI, с английского языка “User Interface”-это «пользовательский интерфейс», является определением того, как выглядит интерфейс и какие свойства он приобретает. Например, цвет, удобство при попадании пальцев на клавиши, читабельность текста и другое.

Разрабатываемый нами интерфейс должен отличаться простотой и легкостью использования, потому что возраст аудитории варьируется в пределах от 10 до 70 лет. Веб-сайт состоит из трех страниц: информативная часть, страница конвертера текстов и страница конвертера файлов.

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



Рисунок 3.10 – главная страница

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

```

<script type="text/javascript" src="/e9947f1ab0"/></script><script src="/nr-1167.min.js.see_mawamm"/></script><script type="text/javascript">window.NREUM||(NREUM=[]);NREUM.info = {agent:"","beacon":true,data.net,"teze
<link rel="stylesheet" href="/app.min.css">
<link rel="stylesheet" href="/osa">
<style rel="stylesheet">
  body { font-family: 'Roboto' }
</style>
<link id="omatom-css" rel="stylesheet" href="/7885084-5779-402b-8210-0b0027abb46f.oss">
</script>
<script type="text/javascript">
(function () {
  var appName = 'answers-widget';
  window.fedops = window.fedops || {};
  window.fedops.apps = window.fedops.apps || [];
  window.fedops.apps[appName] = { staticUrl: performance.now() };
  window.fedops.sessionId = window.localStorage.getItem('fedops_logger_sessionId') || 'xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx'.replace(/(x)/g, function(c) { var z = Math.random()*16|0, v = c == 'x' ? z : (z&3|0&3); return
(new Image()).src = '//log.wix.com/fed?appName=' + appName + '&src=72e618b-850-402b-a27e-1e8c8e8c07f1','parentId':'00000000-0000-0000-0000-000000000000','creationD
});
</script>
<link type="text/css" rel="stylesheet" href="/fontFace.oss"><link type="text/osa" rel="stylesheet" href="/iframe.oss"></head>
<body class="dashboard-answers-widget-business-manager-dashboard" style="display: block;">
<div id="root"><div class="router-container" data-reactroot=""><div class="module-container slide-down-page module-10 search-only"><div class="module"><span><div class="widget-module kb-container kb-home-page" data-item-id"
<script charset="UTF-8">
window._STATIC_URL = '//static.parastorage.com/services/answers-widget/1.844.0/';
window._BASE_PATH = '/apps/widget/v1/vix/388ed9c-2e9d-42ca-9402-a2d3c3ce439/view/ru/';
window._INITIAL_STATE = {locale:"ru","appId":1,"userId":1,"categoryTree":{"children":[{"id":"70b7e18b-850-402b-a27e-1e8c8e8c07f1","parentId":"00000000-0000-0000-0000-000000000000","creationD
</script>
<script src="/answers.ru.js.see_mawamm"/></script>
<script src="/app.bundle.min.js.see_mawamm"/></script>
<script id="custom-js" src="/file.js.see_mawamm"/></script>
</body></html>

```

Рисунок 3.11 - Код аналитики

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

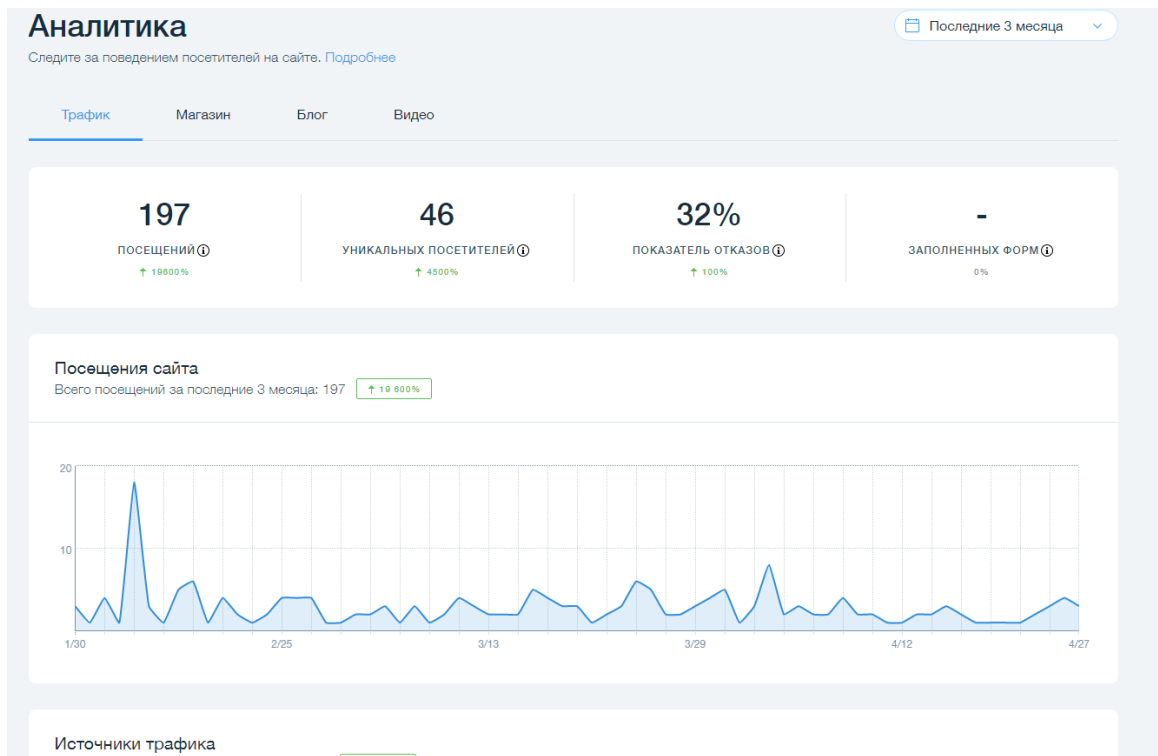


Рисунок 3.12 - Общая аналитика Сайта.

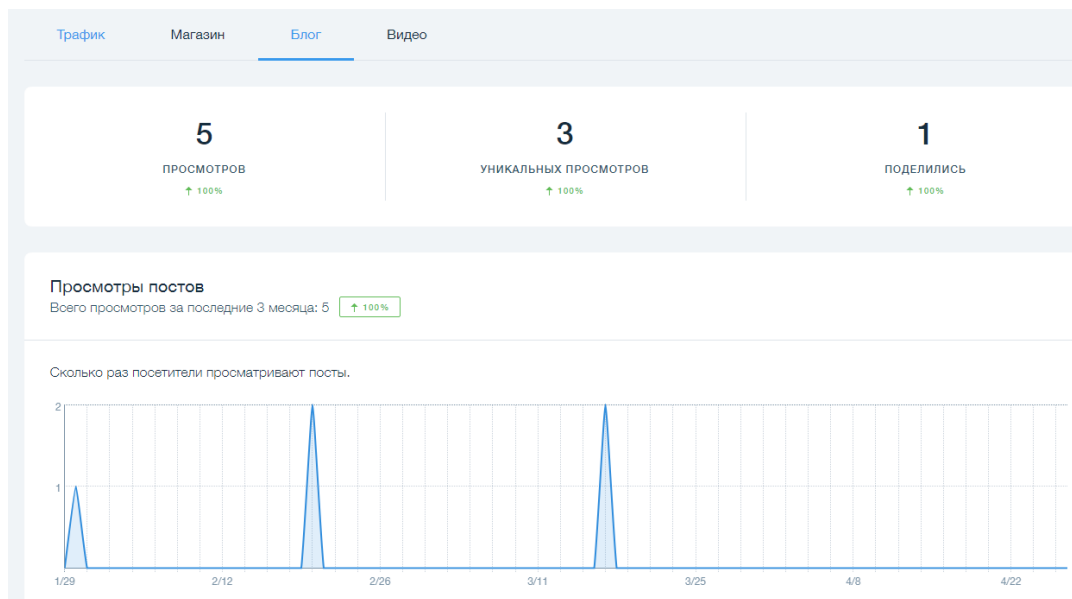


Рисунок 3.13 - Аналитика разных блоков сайта

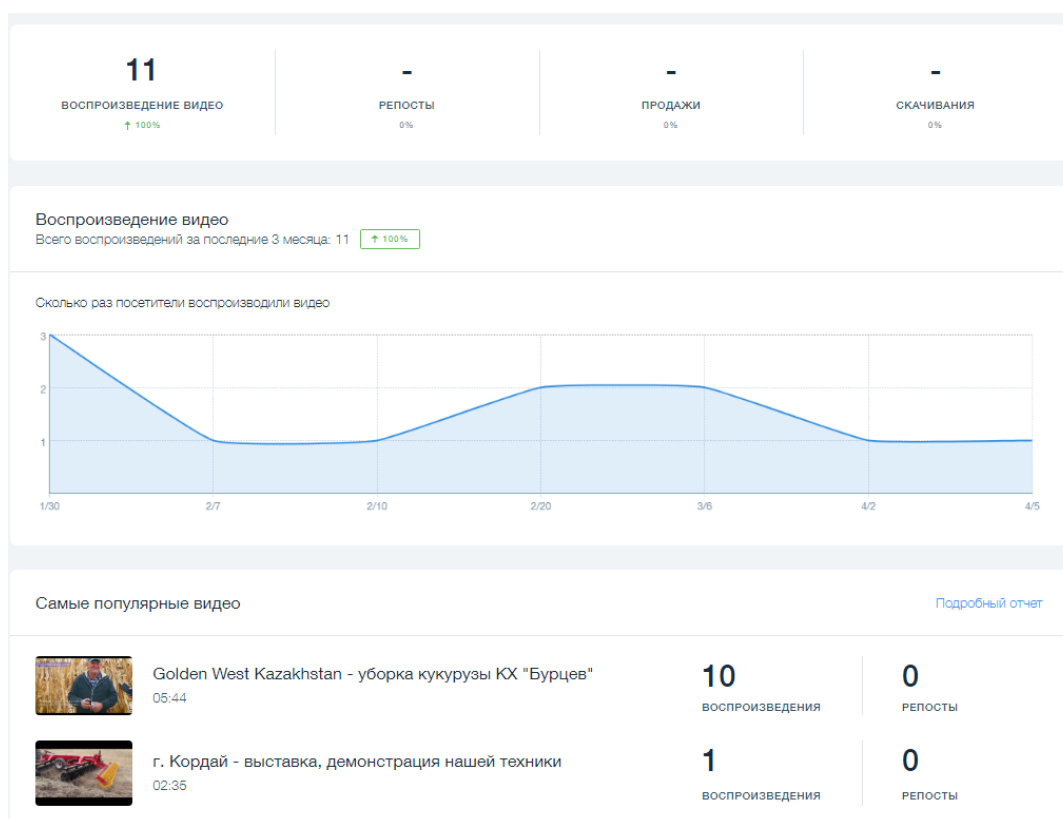


Рисунок 3.14 – Аналитика видео контента

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

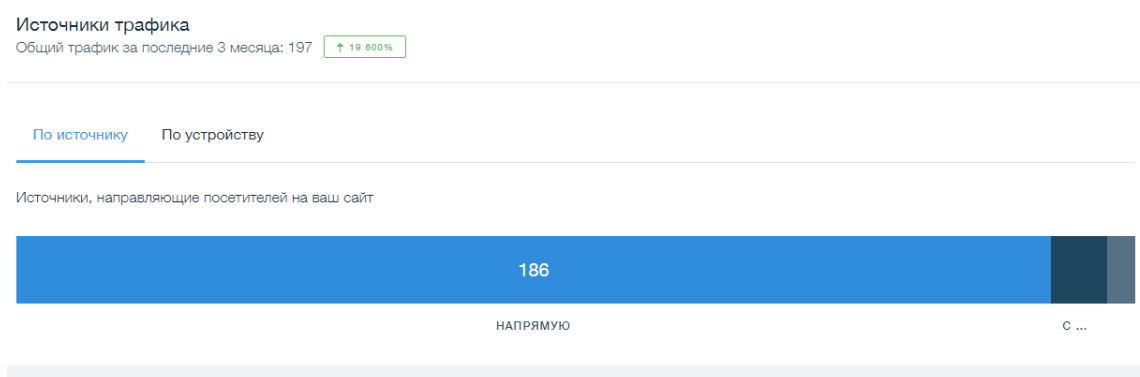


Рисунок 3.15 – Аналитика источника трафика

Источники трафика

Общий трафик за последние 3 месяца: 197 ↑ 19 000%

По источнику По устройству

Устройства, которые посетители используют для просмотра сайта

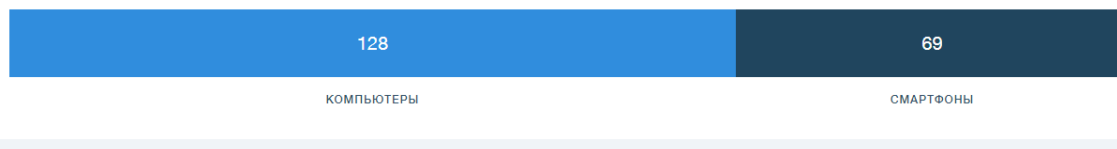


Рисунок 3.16 – Аналитика устройства посещения

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

```
44 <!-- META DATA -->
45 <script type="text/javascript">
46
47
48     var serviceTopology = {"cacheKillerVersion": "1", "staticServerUrl": "https://static.parastorage.com/",
49     var santaModels = true;
50     var isStreaming = true;
51     var rendererModel = {"metaSiteId": "618e077d-c7dd-4c32-b032-963f172f5981", "siteInfo": {"documentType":
52     var publicModel = {"domain": "kupiltraktor.kz", "externalBaseUrl": "https://www.kupiltraktor.kz/", "u
53
54     var isSAVRollout = "false";
55     var googleAnalytics = "";
56     var ipAnonymization = false;
57
58     var googleRemarketing = "";
59     var googleTagManager = "";
60     var facebookRemarketing = "";
61     var yandexMetrika = "";
62 </script>
63
64
```

Рисунок 3.17 – Код подвязки к сервисам

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

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

е > Рабочий стол > Diplom > Сайт > Аналитика_files

Поиск: Аналитика_fi

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

Рисунок 3.18 – фрагмент CSS файлов

е > Рабочий стол > Diplom > Сайт > О компании_files

Поиск: О компании_files

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

Рисунок 3.19 – фрагмент CSS файлов

4 Экономический расчет

4.1 Цели и задачи, решаемые в экономической части

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

Для нахождения себестоимости необходимо учесть:

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

4.2 Расчет трудоемкости разработки ПП

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

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

4.3 Расчет затрат на разработку ПП

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

- материальные затраты;

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

Таблица 4.3.1 - Затраты на материальные ресурсы

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

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

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

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

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

Общая сумма затрат на электроэнергию ($Z_э$) рассчитывается по формуле:

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

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

Затраты на электроэнергию находятся исходя из продолжительности периода разработки ПП, количества кВт/ч, затраченных на проектирование ПП и тарифа за 1 кВт/ч. Тариф по городу Алматы для юридических лиц в

2020 году составляет 19,18 тенге за 1 кВт/ч с учетом НДС (согласно данным представленным на официальном сайте ТОО «АлматыЭнергоСбыт»).

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

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

Таблица 4.3.2 - Затраты на электроэнергию

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

4.4 Расчет затрат на оплату труда

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

Рабочие часы сотрудника за месяц определяются по формуле:

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

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

$N_м$ — количество рабочих дней за месяц;

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

$$Ч_м = 21 \cdot 7 = 147 \text{ ч}$$

Ставка работника за час работы рассчитана по формуле:

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

Инженер-разработчик:

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

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

Для определения трудоемкости разработки ПП используются данные из таблицы 4.2.1.

Трудоемкость разработки Middle Full stack разработчика

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

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

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

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

Инженер-разработчик:

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

Таблица 4.4.1 - Затраты на оплату труда

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

Дополнительная заработная плата:

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

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

Фонд заработной платы:

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

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

4.5 Расчет затрат по социальному налогу

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

Таблица 4.5.1 – Формулы расчета социального налога

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

Расчет амортизационных основных фондов

Полная сумма амортизационных отчислений определяется по формуле:

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

где, Φ_i - стоимость i -го ОФ, тг;

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

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

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

i - вид ОФ;

n - количество ОФ.

Расчет годовой нормы амортизации ОФ:

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

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

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

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

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

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

Оборудование:

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

Таблица 4.5.2 - Амортизация основных фондов (ОФ)

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

Таблица 4.5.3 - Смета затрат на разработку ПП

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

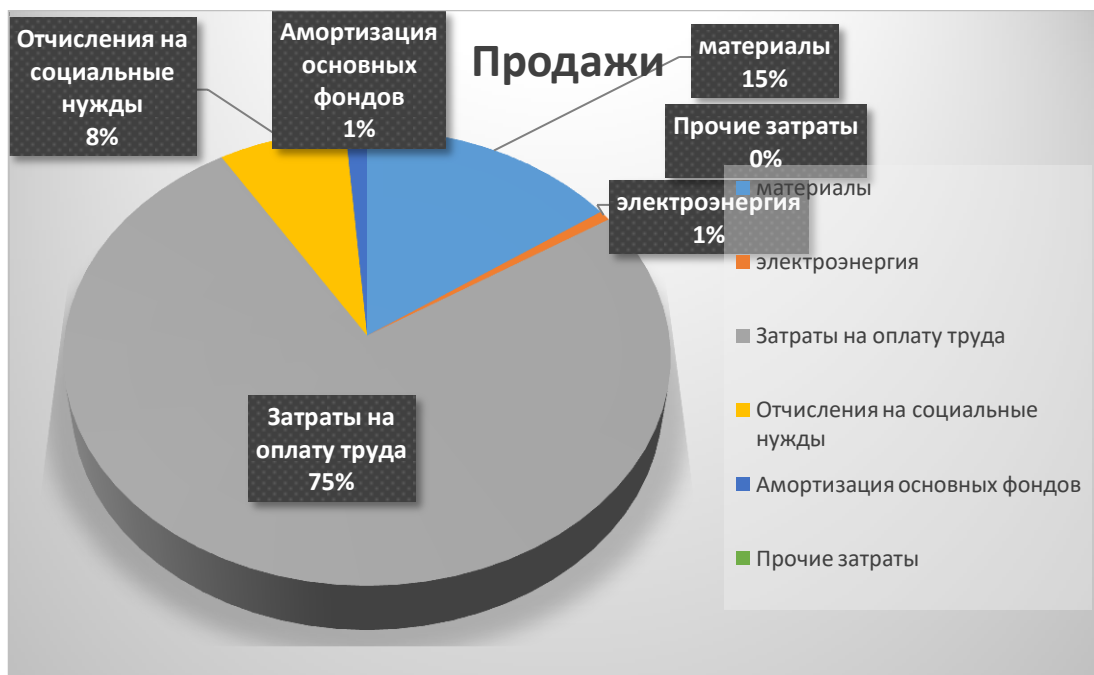


Диаграмма 1 – Смета затрат на разработку проекта

4.6 Определение возможной (договорной) цены ПП

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

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

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

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

P - средний уровень рентабельности ПП - 20 %

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

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

Цена реализации с учетом НДС рассчитывается по формуле:

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

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

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

4.7 Расчет эксплуатационных затрат

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

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

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

ОТ - отчисления по социальному налогу, тенге;

А - затраты на амортизацию, тенге;

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

НР - накладные расходы, тенге.

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

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

где O_c - оклад специалиста, тенге/мес.;

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

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

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

K_d - коэффициент дополнительной заработной платы.

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

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

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

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

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

$$t_{\text{общ}} = 2 \cdot 60 + 2 \cdot 30 = 120 + 60 = 180 \text{ часов}$$

$$ЗП = \frac{240\,000 \cdot 1 \cdot 12}{1920} \cdot 180 \cdot 12 \cdot 1 = 3\,240\,000 \text{ тенге}$$

Материальные затраты определяются по формуле:

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

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

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

В условиях функционирования разрабатываемой ИС, не предполагается материальных затрат.

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

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

где N_a — норма амортизации (25 %);

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

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

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

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

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

Таблица 4.7 – Эксплуатационные затраты

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

C = 5 904 404 тенге

4.8 Расчет результатов от создания и использования ИС

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

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

Статьи затрат при применении ИС включают в себя:

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

- накладные расходы.

Данная система не предполагает расхода каких-либо материалов.

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

$$ЗП = 5\,040\,000 \text{ тенге}$$

$$СО = 527\,184$$

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

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

$$А = (300\,000 \cdot 0,25) + (150\,000 \cdot 0,25) + (50\,000 \cdot 0,25) = 125\,000 \text{ тенге}$$

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

Общие накладные расходы составят:

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

Статьи затрат без применения ИС включают в себя:

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

Таблица 5.8 – Годовые эксплуатационные затраты

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

Всего	9 720 184	5 904 404
-------	-----------	-----------

Ожидаемая условно-годовая экономия определяется по формуле:

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

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

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

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

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

4.9 Расчет основных показателей экономической эффективности

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

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

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

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

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

K — капитальные вложения, тенге;

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

Нормативный коэффициент экономической эффективности капитальных вложений определяется по формуле:

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

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

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

$$E_H = \frac{1}{4} = 0,25$$

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

Расчетный коэффициент экономической эффективности капитальных вложений составляет:

$$E_p = \frac{\mathcal{E}_r}{K} \quad (4.21)$$

где E_p - расчетный коэффициент экономической эффективности капитальных вложений;

\mathcal{E}_r — ожидаемая годовая экономия, тенге;

K — капитальные вложения на создание системы, тенге.

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

Расчетный срок окупаемости капитальных вложений составляет:

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

где E_p - коэффициент экономической эффективности капитальных вложений.

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

Таблица 5.9 – Показатели сравнительной экономической эффективности от внедрения программного продукта

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

Таким образом, разработанная информационная система позволила не только упростить процесс управления работой проектной группы организации и процесс принятия решений руководством, но и значительно сэкономить на различных факторах. Во-первых, автоматизация процесса управления позволит больше не использовать услуги дополнительных кураторов проектов по филиалам, так как теперь отсутствует необходимость осуществлять контроль непосредственно на месте филиала, вся работа организации сосредоточена в ИС. Во-вторых, значительно снизится расход материалов, за

счет того, что назначение на задания и все изменения по заданиям можно сделать прямым в системе, не прибегая к формированию документов по назначению и их распечатке. Ожидаемый годовой экономический эффект составил 1 079 582 тенге. Приложение окупится в первые 5,3 месяцев использования.

5 Безопасность жизнедеятельности

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

Автоматизирование вышеизложенных услуг позволяет работникам рекламных компании больше времени уделять улучшению качества самих программ. Этими приложениями абоненты смогут воспользоваться посредством компьютеров и мобильных телефонов. На сегодняшний день больше людей предпочитают применение через мобильные телефоны. Создание этого проекта проводилась в бизнес центре.

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

В используемом офисе расположены руководитель, менеджеры, дизайнер находящиеся за своими рабочими местами. Во время создания программного продукта главную роль играет безошибочная организация условий труда на рабочем месте.

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

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

Наиболее характерными факторами риска для работников офиса (согласно закону N 528-2 «о безопасности и охране труда») являются перенапряжение зрения и внимания; превышение норм электромагнитного,

рентгеновского, ультрафиолетового и инфракрасного излучения; чрезвычайно высокий уровень статического электричества и показатель запыленности воздуха рабочей зоны, длительное нахождение в сидячем положении, монотонность труда, высокая концентрация в воздухе рабочих мест двуокиси углерода, озона, аммиака, фенола и формальдегида. Часто работники офисных помещений жалуются на следующие симптомы: жжение в глазах, покраснение век и конъюнктивы, чувство инородного тела или песка под веками, боли в области глазниц и лба, затуманивание зрения, замедленная перефокусировка с ближних объектов на дальние. Причиной тому является переутомление зрительного аппарата при длительной работе с экраном без перерывов. Данные симптомы имеют наименование данное Всемирной Организацией Здравоохранения, и именуется как «компьютерный зрительный синдром».

Потому как написание работы зависело от постоянного пребывания за компьютером, стал вопрос анализа систем пожаротушения, для безопасности.

Рассмотрим пример типового производственного помещения в бизнес центре. В помещении, кроме оборудования и рабочих мест, могут находиться кондиционеры и обогреватели. Офис в котором располагались 12 компьютеров. Выполняют работу 12 сотрудников: 2 разработчика, 2 администратора, 2 контент менеджера, 1 оператор по видео, 1 аудитор, 1 HR-менеджер и 3 начальника.

Так как сотрудники находились в не благоприятных условиях труда, это может подвергнуть к возникновению профессиональных заболеваний.

Офис имеет различные параметры:

а) размеры рабочего офиса:

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

б) общая площадь помещения составляет 153м^2

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

К тому же необходимо соблюдение главных критериев для организации рабочего места пользователя ПЭВМ

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

Свойства применяемого оборудования

При использовании и разработке в будущем программного продукта применяются ниже записанные приборы:

- А) компьютеры – 12 шт.;

Б) Люминесцентные лампы Т8/18W, 15 светильников по 4 лампы E=400лк;

В) Кондиционер;

Г) Система пожаротушения.

В таблице 5.1 указаны показатели идеальной нормы параметров микроклимата, беря во внимание сезонность года. Приборы, расположенные в офисе, не относятся к источникам выделяемого тепла.

Таблица 5.1 – Идеальные нормы микроклимата для помещений с компьютеров

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

Одним из главных факторов устройства труда является пожарная безопасность.

В случаях неверной организации труда это повлечёт за собой к плохим результатам в чрезвычайных ситуациях. В случае в отказе в установке пожарной системы и сигнализации, это приведет к тому что при возникновении пожара не все сотрудники будут оповещены (без сигнализации) и не будут предприняты все необходимые меры безопасности. В нашем помещении пожар может возникнуть при следующих критериях:

- 1)небольшое замыкание;
- 2)открытые провода;
- 3)искры огня из плиты;

Проанализируем следующие виды систем пожаротушения и их особенности.

Благодаря тому, что в нашем помещении уже оборудовано пожарной системой, нам остается проанализировать их виды. Бывают следующие виды: сигнализации, автоматические системы пожаротушения, схемы пожаротушения по кнопке. К категории сигнализаций входят датчики, реагирующие на температуру (что может быть не удобно), либо на дым (что становится высокоэффективным). Рассмотрим также системы пожаротушения. Система пожаротушения имеет несколько подвидов: водяные, порошковые, газовые, аэрозольные и пенные. Отличим их становится такие критерия как: общая площадь покрытия, непрерывность работы и конечно употребляемость воды.

Проанализировав разные виды пожаротушения, можно сказать что эффективная, к тому же часто встречаемым видом является водяная. В ценовой политике она также имеет преимущества по сравнению с другими видами, так как является дешевым, но в тот же момент позволяет справиться с пожаром и закрытых помещениях.

Если сравнить водяную и пенную систему тушения в эффективности, то преимущество переходят пенной системе. Потому что пенная может побороться с более сложным пожаром за быстрое время расходуя меньше воды.

Подведя итоги могу сказать что данный анализ помогает понять и выбрать более эффективную систему пожаротушения, которую и мы учтем в своей работе.

5.2 Методика расчета установок водяного и пенного пожаротушения

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

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

В нижеуказанной таблице 5.2 приводятся характеристики спринклерных установок внутри офиса.

Таблица 5.2. Параметры спринклерных установок

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

Гидравлический расчет выполняется при условии водоснабжения от основного источника водоснабжения, в нашем случае центральное

водоснабжение. Максимально давление на узле управления должно быть 1,0 Мпа.

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

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

где k – коэффициент производительности распылителя по таблице 5.3;

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

Таблица 5.3. Коэффициенты производительности оросителя

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

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

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

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

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

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

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

b – наибольшая ширина совмещенных столов, м;

n – количество техники;

q_n – интенсивность орошения, принимается по таблице 5.2.

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

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

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

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

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

B – характеристика трубопровода, рассчитываемая по формуле 5.4:

$$B = \frac{k_1}{l}, \quad (5.4)$$

где k_1 – коэффициент, параметр труб, принимается по таблице 5.4;

l – длина расчетного участка трубопровода, м.

Таблица 5.4. Параметры труб

Трубы	Диаметр условного прохода, мм	Диаметр наружный, мм	Толщина стенки, мм	Значение k_1
Стальные электро-сварные (ГОСТ 10704-76)	15	18	2,0	0.0755
	20	25	2,0	0.75
	25	32	2,2	3.44

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

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

Далее по формуле 5.3 рассчитываем потери:

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

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

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

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

Таблица 5.5. Коэффициент потерь напора в узле управления

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

По формуле 5.5 рассчитываем:

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

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

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

где k_2 – коэффициент разрушения пены по таблице 5.6;

k_3 – кратность пены;
 V – объем помещения, м³.

Таблица 5.6. Коэффициент разрушения пены

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

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

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

Далее рассчитаем количество одновременно работающих генераторов пенного раствора:

$$n = \frac{V_1}{Q_d \cdot t'} \quad (5.7)$$

где Q_d – примерная производительность одного генератора пенообразующего раствора, для генератора средней кратности 20 м³/мин;
 t – продолжительность работы установки с пеной средней кратности, мин. Для пены средней кратности и помещения с твердыми горючими материалами продолжительность работы установки будет равна 25 мин.

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

Одновременно будет работать 0,184 генераторов пенного раствора.

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

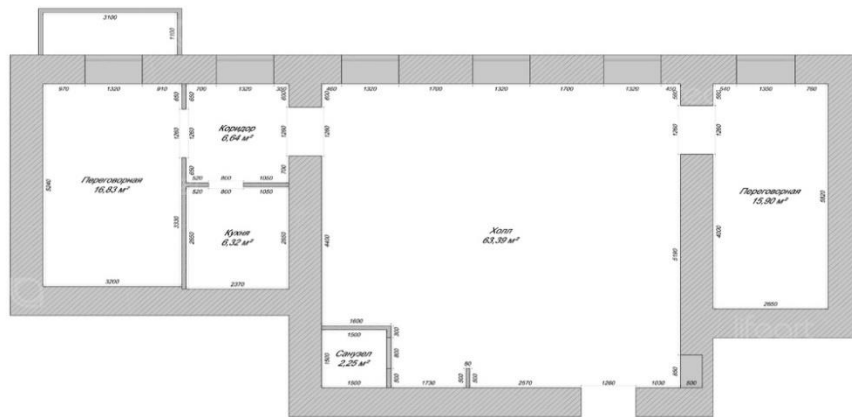


Рисунок 5.1 – схема офиса

5.3 Расчет искусственного освещения

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

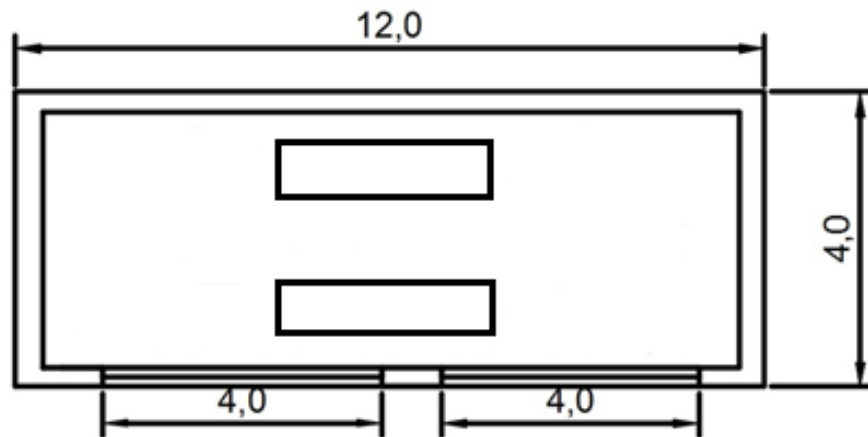


Рисунок 5.3.1 – Схема расчета освещенности.

Таблица 5.3 – Технические характеристики ламп PLE-TPRO23/230

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

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

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

где, $S_{\text{ос}}$ – площадь помещения = 153;

k_z – коэффициент запаса = 1,2;

N – количество светильников = 2;

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

n – количество ламп в светильнике = 4;

$\Phi_{\text{л}}$ – световой поток выбранной лампы, $\Phi_{\text{л}} = 1500$ лм

η – коэффициент использования, $\eta = 60\%$

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

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

Коэффициенты отражения от потолка стен и пола соответственно равны:

$$\rho_{\text{пот}} = 70 \%$$

$$\rho_{\text{ст}} = 50 \%$$

$$\rho_{\text{пол}} = 30 \%$$

Вычислим высоту подвеса светильника над рабочей поверхностью по формуле 5.6:

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

где: h_c – расстояние от светильника до перекрытия, $h_c = 0,1$ м;
 h_p – высота рабочей поверхности над полом, $h_p = 0,6$ м;
 h – высота помещения, $h = 3$ м.

$$H = 3 - 0,1 - 0,6 = 2,3 \text{ м}$$

Лучшее расстояние от окна до светильника определяется по формуле 5.8:

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

где, $\lambda = 0,6 \div 1,5$

$$L = 1,5 \cdot 2,3 = 3,45 \approx 3,5 \text{ м}$$

Расстояние между светильниками:

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

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

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

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

Определяем индекс помещения по формуле 5.10

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

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

Коэффициент использования в данном случае равен $\eta = 60\%$, коэффициент запаса равен $k_3 = 1,2$

Определим количество люминесцентных ламп по формуле 5.10

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

где, S_{OC} – площадь помещения;

k_z – коэффициент запаса;

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

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

n – количество ламп в светильнике;

$\Phi_{л}$ – световой поток выбранной лампы, $\Phi_{л} = 3570$ лм

η – коэффициент использования, $\eta = 60\%$

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

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

Заключение

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

При реализации дипломного проекта были выполнены следующие задачи:

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

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

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

Список литературы

- 1 Основы безопасности жизнедеятельности: Учебное пособие / сост.: М.К.Дюсебаев, Ж.С.Абдимуратов. -Алматы: АУЭС, 2013.-79 с
- 2 Калюжный Е.А. Безопасность жизнедеятельности: Учебное пособие / Е.А. Калюжный, С.В. Михайлова, С.Г. Напреев, Д.Г. Сидоров. - Арзамас:
- 3 Абрамов В.В. Безопасность жизнедеятельности: Учебное пособие для вузов. – Спб.: Питер, 2013. – 365 с.
- 4 Вершина, Г.А. Охрана труда: учебник / Г. А. Вершина и др. – Минск: ИВЦ Минфина, 2017. – 412 с
- 5 ДП ФАИТ ММиПО Алимбетова АС 0806
http://aipet.kz//student/diplom/2017/fait/kt/kt_en/alimbetova_as.pdf.
- 6 <http://unlimsoft.com/native>
- 7 <https://megabook.ru/>
- 8 ГОСТ 12.1.004-91 «Система стандартов безопасности труда.»
- 9 Г. Боканова Методические указания по выполнению экономической части дипломных работ Алматы, АУЭС, 2020 – 35с.
- 10 Техничко-экономическое обоснование дипломных проектов Брест, БГТу, 2014 – 15с.
- 11 Методические указания к выполнению расчетно-графической работы для студентов специальности 080801 «Прикладная информатика (по областям)» Уфа, 2010 – 12с.
- 12 Методические указания по выполнению экономической части дипломных работ Москва, Московский университет им. С.Ю. Витте, 2016 – 21с.
- 13 Симионов Ю.Ф., Боромотов В.В. Информационный менеджмент. — Ростов н.Д: Феникс, 2013, 250с.
- 14 Бардзелл, Джеффри Macromedia Dreamweaver MX 2004 с ASP, ColdFusion и PHP. Из первых рук (+ CD-ROM) / Джеффри Бардзелл. - М.: Эком, 2017. - 560 с.
- 15 Бенкен, Е. PHP, MySQL, XML. Программирование для Интернета / Е. Бенкен. - М.: БХВ-Петербург, 2016. - 352 с. Wikipedia: Парковочные часы / URL: https://en.wikipedia.org/wiki/Parking_meter
- 16 Прохоренок, Николай HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера (+ CD-ROM) / Николай Прохоренок. - М.: БХВ-Петербург, 2012. - 912 с.
- 17 Andy, Harris HTML, XHTML and CSS All-In-One For Dummies® / Andy Harris. - Москва: Наука, 2014. - 173 с.
- 18 Ben, Henick HTML & CSS – The Good Parts / Ben Henick. - Москва: СИНТЕГ, 2013. - 350 с.
- 19 Ed, Tittel HTML, XHTML & CSS For Dummies® / Ed Tittel. - Москва: Гостехиздат, 2012. - 416 с.

20 Информационные технологии управления: Учебное пособие/ М.В. Бастриков, О.П. Пономарев; Институт «КВШУ». – Калининград: Изд-во Ин-та «КВШУ», 2005.– 140 с.

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

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

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

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

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

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

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

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

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

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

Приложение А

Техническое задание

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

Общие требования:

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

Технические требования:

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

Продолжение приложения А

- тестирование системы продаж на корректность;
- тестирование платформы на появление ошибок и перегруженности;

Экономические требования:

- расчет стоимости системы и стоимости разработки программного обеспечения (подлежит обсуждению):
- стоимость готового продукта 5 904 404тг;
- стоимость разработки 1 513 040тг.
- потенциальные клиенты и области применения:
- веб-сайты и приложения по всему миру;

Приложение Б (ЛИСТИНГ ПРОГРАММЫ)

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

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

Продолжение приложения Б

```
wixBiSession.caching = undefined;
}

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

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

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

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

function isCached(part) {
```


Продолжение приложения Б

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

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

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

if (performance.measure) {
```

Продолжение приложения Б

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

Продолжение приложения Б

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

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

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

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

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

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

Продолжение приложения Б

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

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

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

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

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

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

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

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

```
<meta property="og:image:height" content="3543">
<meta property="og:url" content="https://www.kupiltraktor.kz">
<meta property="og:site_name" content="УТО AGRO Казахстан">
<meta property="og:type" content="website">
<meta name="keywords" content="купить трактор, купить трактора, трактор в
алматы">
<meta name="google-site-verification" content="-
xMbwBqIf2RaKNxXyyAI25a2HDg8Geli0MJ0rYRkXA">
```

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

Продолжение приложения Б

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

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

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

Продолжение приложения Б

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

var clientSideRender = false;

</script>

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

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

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

Продолжение приложения Б

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

``

``

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

``

``

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

``

``

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

``

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

``

``

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

``

``

Продолжение приложения Б

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

``

``

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

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

```
}
```

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

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


Приложение В Акт внедрения

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



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

АКТ ВНЕДРЕНИЯ

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

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

Макеева Ж.М.

Подпись _____

