

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН  
Некоммерческое акционерное общество  
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»  
Кафедра IT-инжиниринг

**ДОПУЩЕН К ЗАЩИТЕ**

Заведующий кафедрой

PhD, доцент

Т.С. Картбаев

« \_\_\_\_ » \_\_\_\_\_ 2018 г.

**ДИПЛОМНАЯ РАБОТА**

На тему: Разработка и управление web-ориентированной информационной системы для предприятия «КазЭнергоПромТорг»

Специальность 5B060200 – «Информатика»

Выполнила Джамалова Д.Н.                      Группа ИНФ-14-2  
Научный руководитель ст. преподаватель Альмуратова К.Б.

Консультанты:

по экономической части: к.э.н., доцент \_\_\_\_\_ А.И. Бекишева  
« 29 » 05 \_\_\_\_\_ 2018 г.

по безопасности жизнедеятельности: к.т.н., доцент \_\_\_\_\_ А.А. Абикенова  
« 31 » 05 \_\_\_\_\_ 2018 г.

по применению  
вычислительной техники: ст. преп. \_\_\_\_\_ А.М. Рамазанова  
« 22 » 05 \_\_\_\_\_ 2018 г.

Нормоконтролер: ст. преп. \_\_\_\_\_ Ш.Д. Толыбаев  
« 31 » 05 \_\_\_\_\_ 2018 г.

Рецензент: к.т.н., ассистент профессора \_\_\_\_\_ С.Т. Аманжолова  
« \_\_\_\_ » \_\_\_\_\_ 2018 г.

Алматы 2018

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН  
Некоммерческое акционерное общество  
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»

Институт систем управления и информационных технологий

Кафедра IT-инжиниринг

Специальность 5В060200 – «Информатика»

**ЗАДАНИЕ**

на выполнение дипломной работы

Студенту Джамаловой Денизе Низамидиновне

Тема работы: Разработка и управление web-ориентированной информационной системы для предприятия «КазЭнергоПромТорг»

Утверждена приказом по университету № 155 от «23» октября 2017 г.

Срок сдачи законченной работы «01» июня 2018 г.

Исходные данные к работе (требуемые параметры результатов исследования (проектирования) и исходные данные объекта): Руководство качества системы на предприятии; данные о предприятии «КазЭнергоПромТорг», данные преддипломной практики.

Перечень вопросов, подлежащих разработке в дипломной работе, или краткое содержание дипломной работы:

- а) современное состояние системы менеджмента качества на предприятии;
- б) исследование предметной области;
- в) проектирование web-ориентированной информационной системы;
- г) разработка программного продукта;
- д) вопросы безопасности жизнедеятельности и охраны труда;
- е) экономическая эффективность работ по стандартизации.

Перечень графического материала (с точным указанием обязательных чертежей): представлены 15 таблиц, 20 иллюстрации.

Основная рекомендуемая литература:

1 Борисов, Е. Ф. Основы экономики: Учебное пособие / Е. Ф. Борисов. – М.: Юрайт – Издат, 2009. – 316 с.

2 Экономическая теория: Учебник/ под ред. В.Д. Камаева, Е.И.Лобачевой. – М.: Юрайт-Издат, 2010. – 557с.

3 Экономика и анализ деятельности предприятий / Т.П. Елисеева, М.Д.Молев, Н.Г. Трегулова. - Ростов-на-Дону: Феникс, 2011. - 480 с.

4 Стивен Хольцнер. PHP в примерах. / Стивен Хольцнер. М.: «Бином-Пресс», 2011 г. Пер. с англ. 352 с

5 Гутманс Э., Баккен С, Ретанс Д. PHP 5. Профессиональное программирование. / Пер. с англ. СПб: Символ- Плюс, 2009. 704 с., ил.

6 Веллинг Л. PHP и MySQL Web-разработчикам.- М.: SAMS, 2012. - 893с.

7 Томсон Лаура. Разработка Web-приложений на PHP и MySQL: Пер. сангл. /Лаура Томсон, Люк Веллинг. – 2-е изд., испр. – СПб: ООО «ДиаСофтЮП», 2013. – 672 с.

8 Дубаков М.А. Создание Web-страниц: искусство верстки. – Мн.:Новое знание, 2014. – 287 с.

9 Ларри Ульман. Ульман Л. Основы программирования на PHP:/Ларри Ульман. Пер. с англ. -М.: ДМК Пресс, 2011. -288 с.: ил. (Самоучитель).

10 Нидерст Б. Web-мастеринг для профессионалов. «Настольный справочник». - М.: БИНОМ, 2013. - 313 с.

Консультации по работе с указанием относящихся к ним разделов работы

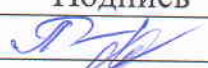



Раздел	Консультант	Сроки	Подпись
Экономическая часть	Бекишева А.И.	23.05 - 29.05.18	
Безопасности жизнедеятельности	Абикенова А.А.	12.04 - 31.05.18	
Программная часть	Рамазанова А.М.	05.05 - 22.05.18	
Нормоконтролер	Толыбаев Ш.Д.	05.05. - 31.05.18	

График  
подготовки дипломной работы

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
Исследование предметной области	01.11.2017- 20.12.2017	<i>Выполнено</i>
Проектирование web-ориентированной информационной системы	21.12.2017- 20.02.2018	<i>Выполнено</i>
Разработка программного продукта	21.02.2018- 30.04.2018	<i>Выполнено</i>

Дата выдачи задания «25» октября 2017 г.

Заведующий кафедрой \_\_\_\_\_ *Т.С. Картбаев* Т.С. Картбаев

Научный руководитель работы \_\_\_\_\_ *К.Б. Альмуратова* К.Б. Альмуратова

Задание принял к исполнению студент \_\_\_\_\_ *Д.Н. Джамалова* Д.Н. Джамалова

## **Аннотация**

Данная дипломная работа посвящена разработке веб-сайта для предприятия «КазЭнергоПромТорг». Для реализации сайта были использованы языки программирования CSS, HTML, JavaScript и др. Были произведены различные расчеты, которые подтвердили его целесообразность. В главе обеспечения безопасности жизнедеятельности был проведен анализ условий труда, освещения, микроклимата, расчет вентиляции, нормирование шума, электромагнитное и ионизирующее излучения, а также пожарная безопасность. В экономической части дипломной работы был произведен расчет затрат на оплату труда, оборудования, программные обеспечения, амортизационные отчисления, а также была составлена итоговая смета затрат на разработку веб-сайта.

Дипломная работа состоит из введения, пяти глав и заключения. Актуальность и основная идея по созданию сайта были раскрыты во введении. В главах описывается поэтапный процесс создания сайта с нуля до функционирующего сайта. Основные итоги о проделанной работе описаны в заключении работы.

## **Андатпа**

Бұл дипломдық жұмыс «КазЭнергоПромТорг» компаниясының веб-сайтын құруға арналған. Жобаны іске асыру үшін JavaScript, HTML, CSS және т.б. бағдарламалау тілдері қолданылған. Оның мақсаттылығын растайтын түрлі есептеулер жүргізілген. Тіршілік әрекетінің қауіпсіздігі тарауында еңбек шарттарының, жарықтандырудың, микроклиматын, желдеткіш есептеулері, шу деңгейі, электромагниттік және иондаушы радиация және өрт қауіпсіздіктің есебі жүргізілді. Дипломдық жұмыстың экономикалық бөлімінде еңбек төлемінің, жабдықтардың, бағдарламалық қамтамасыз етудің, амортизациялық есептердің шығындардың есептеулері жүргізілген.

Дипломдық жұмыс кіріспеден, бес бөлімнен және қорытындыдан тұрады. Кіріспеде сайтты құру бойынша өзектілігі және негізгі идеясы ашылған. Тарауларда сайттың толық даму кезеңдерін сипаттайды. Жұмыстың жасалғаны туралы негізгі қорытындылар жұмыстың қорытындысында сипатталған.

## **Annotation**

This diploma project is devoted to the creation of a web site for “КазЭнергоПромТорг” company. For site’s implementation were used programming languages such as JavaScript, HTML, CSS, etc. Was made a difficult calculations of the project, which confirmed its economic viability. In the life safety part was made analysis of working conditions, lighting, microclimate, calculation of

ventilation, noise level, electromagnetic and ionizing radiation and fire safety. In the economic part was made salary's, equipments's, software's, depreciation deductions's calculations of expenses.

The diploma project consists of an introduction, five sections, conclusion. The introduction reveals the main purpose of the work, also urgency of creating a website. Sections describe the stages from the design structure to the created fullfeatured site. In conclusion, the main conclusions on the work done and summarized the study.

## Содержание

	Введение	9
1	Исследование предметной области	10
1.1	Исследование объекта и актуальность разработки	10
1.2	Этапы планирования разработки веб-сайта	11
1.3	Анализ методов и технологий создания сайта	15
2	Проектирование web-ориентированной информационной системы	28
2.1	Описание административной части	28
2.2	Структура сайта	29
2.3	Разработка дизайна сайта	33
2.4	Разработка логотипа предприятия «КазЭнергоПромТорг»	37
3	Разработка программного продукта	41
3.1	Разработка веб сайта на основе html	41
3.2	Дополнительные инструменты разработки	53
4	Технико-экономическое обоснование	63
4.1	Описание работы и обоснование необходимости	63
4.2	Трудоемкость разработки ПП	63
4.3	Расчет затрат на разработку ПП	64
4.4	Материальные затраты	65
4.5	Затраты на электроэнергию	65
4.6	Затраты на оплату труда	66
4.7	Социальный налог	67
4.8	Амортизация основных фондов	67
4.9	Прочие затраты	68
4.10	Смета затрат на разработку ПП	69
4.11	Определение возможной (договорной) цены ПП	69
4.12	Оценка социально-экономических результатов функционирования ПП	70
5.	Безопасность жизнедеятельности	72
5.1	Анализ условий труда	72
5.2	Параметры микроклимата	75
	Заключение	79
	Список литературы	80

## **Введение**

В современном мире ни один бизнес-проект не осуществляется без его технических исследований, одним из важных разработок является создание веб-сайта. Сайт - это место во всемирной сети которое имеет свой адрес, собственного хозяина и состоит из отдельных веб-страниц, которые мы видим как одно целое. Существует множество сайтов различных тематик и предназначений, но для продвижения того или иного бизнеса необходим бизнес сайт, который объединяет в себе всю информацию об организации, как для клиентов, так и для своих сотрудников. Бизнес сайт обладает широкими возможностями для размещения и управления информационными материалами.

Актуальность дипломной работы заключается в удобном для фирм и их клиентов использовании веб-сайта, так как человечество переходит на новый уровень – уровень новых возможностей развития в условиях четвертой промышленной революции.

Целью моей дипломной работы является создания сайта для предприятия «КазЭнергоПромТорг», который должен содержать в себе как необходимые данные для клиентов, так и быть удобным в использовании.

При создании сайта, необходимо обдумать все достоинства и недочеты. Определиться с целью, назначением сайта, задуматься о его функциональной нагрузке.

Данная дипломная работа посвящена разработке веб - сайта для предприятия. Для реализации сайта были использованы такие языки программирования как JavaScript, HTML, CSS и др. Данная работа состоит из пяти глав. Первая глава – общая, посвящена раскрытию идеи целесообразности и обоснование необходимости создания сайта . Вторая глава является аналитической, которая описывает все начальные этапы по созданию сайта, а именно – структура сайта и разработка дизайна сайта. Третья глава - проектно-конструкторская часть, где описываются схемы функционирования сайта и техническая часть, а именно – основы html. В четвертой главе обеспечения безопасности жизнедеятельности был проведен анализ условий труда, освещения, микроклимата, расчет вентиляции, нормирование шума, электромагнитное и ионизирующее излучения, а также пожарная безопасность. В последней, пятой главе - экономической части дипломной работы был произведен расчет затрат на оплату труда, оборудования, программные обеспечения, амортизационные отчисления, а также была составлена итоговая смета затрат на разработку веб-сайта.

Основными задачами моей дипломной работы является создание удобного просмотра информации для клиентов стройфирмы.



## **1 Исследование предметной области**

### **1.1 Исследование объекта и актуальность разработки**

Стройфирма «КазЭнергоПромТорг» является лидером в своем деле. Основная деятельность организации – это строительство и ремонт зданий, сооружений промышленного и гражданского назначения. Также фирма имеет дополнительные функции: планирование строительства, архитектурное моделирование и управление строительством.

Компания «КазЭнергоПромТорг» была создана в 2012 году. В течение пяти лет были выполнены строительные работы. За очень небольшой промежуток времени строительная фирма "КазЭнергоПромТорг" зарекомендовала себя в положительном свете на строительном рынке г. Алматы, наработала постоянных клиентов-заказчиков. В составе предприятия работают специалисты высокой квалификации, отвечающие всем требованиям современных стандартов. Ежегодно обновляется техническое оснащение персонала, а также повышается квалификация работающих. Состав предприятия остаётся постоянным на протяжении длительного времени. ТОО "КазЭнергоПромТорг" выступало как единственным подрядчиком, так и выполняла функции генерального подрядчика. В связи с этим имеется возможность осваивать объёмы работ с высокой скоростью, но в тоже время, делая упор на качество выполняемых работ. В данный момент компания твёрдо стоит на ногах, так как имеет мощную материально-техническую базу, обученный высококвалифицированный персонал, безупречную репутацию.

Основными проектами компании являются общеобразовательная школа в г. Сары-Озек, мкр. Нуркент 1, мкр Нуркент 2, ЖК Евразия в г. Алматы. Также на данный момент ведется строительство пляжного волейбола.

Фирма имеет немало партнеров, являющихся лучшими в своем деле, такие как БАЗИС-А, Артемида ЛТД, VI GROUP, ASU company group, DAMIO group и др.

Обоснованием необходимости создания веб сайта для предприятия является его отсутствие. Компания набирала клиентскую базу путем разглашения информации о себе через различные источники, что не являлось эффективным и удобным как для фирмы, так и для ее клиентов. Предназначение сайта заключается в том, что вся доступная информация о компании, о проектах, о способах связи, адрес, каталог товаров будет находиться на сайте, что облегчит работу сотрудникам и клиентам.

Сайт будет не только существовать, но и улучшаться в ногу со временем. Он будет являться основным источником информации, ссылка на который будет указываться во всех социальных сетях. Именно поэтому создание сайта для предприятия «КазЭнергоПромТорг» является необходимостью первой степени для компании.

## 1.2 Этапы планирования разработки веб-сайта

Определение целей веб-сайта и его позиционирование. На этом этапе необходимо определить, для чего нужен сайт, т.е. какие задачи он должен решать: предоставить общее представление о компании или многосторонне осветить какую-либо сторону человеческой деятельности, увеличить продажи по традиционным каналам или организовать веб-торговлю, провести рекламную или маркетинговую кампанию.

Цели веб-сайта, в большинстве случаев, должны ставиться заказчиком, а затем, вместе с исполнителем они уточняются и корректируются.

Это один из самых важных этапов не только создания веб-сайта как такового, но важнейший этап интернет-маркетинга.

Если заказчик не понимает, для чего ему нужен веб-сайт, с 99% вероятностью он будет недоволен работой исполнителя и будет считать, что деньги, потраченные на создание веб-сайта, просто потеряны. В итоге, компания не будет использовать методы Интернет-маркетинга, что негативно скажется на ее конкурентных позициях на рынке.

После определения целей сайта надо совершенно четко и как можно подробнее представить и описать целевую аудиторию сайта, т.к. это влияет на то, в каком виде будет представлена информация.

Определение целевой аудитории веб-сайта - это не менее важный, чем определение целей веб-сайта, этап. Не всегда целевая аудитория компании из офлайн будет прямо проецироваться на он-лайн. Зная целевую аудиторию и аудиторию российской части Интернета можно сделать некие предположения о том, кто будет являться основными посетителями веб-сайта.

Определение и как можно более подробное описание целевой аудитории сайта дает возможность разработать правильный дизайн для проекта, а также выбрать правильное направление для написания текстов. Очень важно говорить с аудиторией на понятном ей языке. И это требование в разы важнее в Интернете, чем в реальной жизни, т.к. в онлайн вас и вашего конкурента разделяет только клик мыши.

На заключительной стадии этого этапа примерно определяют, по каким поисковым запросам сайт должен появляться в результатах поискового запроса и посмотреть интернет-проекты конкурентов будущего веб-проекта.

Определение поисковых запросов или другими словами составление семантического ядра процесс достаточно сложный и в идеале должен делаться профессионалами. Однако, составить приблизительный список запросов под силу каждому руководителю или владельцу бизнеса. Для этого просто необходимо выписать все те слова и фразы, которые вы лично используете для поиска конкурентов и опросить менеджеров отдела продаж – какие вопросы возникают у клиентов вашей компании наиболее часто. После этого будет очень полезно сверить ваши догадки с картиной запросов, которую дают нам поисковики.

Дополнительную информацию о положении вещей в Интернете на вашем рынке даст обзор сайтов конкурентов. Необходимо иметь в виду, что совсем необязательно конкуренты из реальной жизни будут являться таковыми в Интернете.

Создание Технического Задания (ТЗ) на разработку веб-сайта. В ТЗ необходимо как можно более подробно описать:

- цели создания сайта и его целевую аудиторию;
- структуру веб-сайта и количество страниц в каждом разделе;
- работу динамических модулей;
- пожелания по дизайну (цвета, использование фирменного стиля, соотношение графика/текст и т.д.
- используемые технологии (HTML, Flash, PHP и проч.);
- порядок предоставления, обработки или создания графической и текстовой информации;
- технические требования к сайту.

ТЗ является основным документом, на основе которого осуществляются все последующие этапы разработки веб-сайта.

Очень часто создается ситуация, при которой заказчик ждет ТЗ от исполнителя, а исполнитель от заказчика. Еще раз хотелось бы повторить, что создание веб-сайта – это сотрудничество. В одиночку, ни заказчик, ни исполнитель не могут составить ТЗ. Как правило, заказчик должен четко описать цели веб-сайта, его целевую аудиторию и пожелания по функционалу. Далее за дело берется исполнитель, который техническим языком описывает, как будет работать будущий сайт.

На этапе создания дизайн-макета веб-сайта дизайнер в специальной графической программе создает дизайн страниц будущего веб-сайта с прорисовкой всех графических (банеров, кнопок, фотографий) и текстовых элементов. Дизайнер создает дизайн веб-страниц с учетом пожеланий заказчика и задания, прописанного в ТЗ.

Говоря о визуальном представлении веб-сайта, нельзя не остановиться на таком моменте, как различия между фиксированными и резиновыми сайтами.

Современные мониторы имеют различные разрешения. В разговоре о веб-сайтах нас особенно интересует ширина. Размер монитора по ширине в пикселях может быть 800, 1024, 1280, 1600 и даже больше. Более того, пользователь может просматривать сайт в полускрученном окне.

Фиксированный дизайн сайта предполагает одну единую ширину сайта для всех разрешений экрана и для всех ширин окон, т.е. если ширина сайта 1000 пикселей, то на экране 800 пикселей будет появляться горизонтальная прокрутка, а на экране 1280 пикселей – пустые поля слева и справа.

Такое построение сайта позволяет четко управлять композицией сайта и однозначно знать, где будет каждый элемент навигации на каждом экране. Этот тип сайта позволяет использовать более насыщенные графические элементы.

Резиновый дизайн, в отличие от фиксированного «подстраивается» под ширину экрана.

Это приводит к тому, что сайт без горизонтальной прокрутки увидит большинство пользователей, но на разных экранах композиция сайта будет непредсказуемой, что сильно не по душе дизайнерам.

Зачем знать эти различия? Затем, чтобы правильно выбрать подход к графическому представлению сайта для ЦА, а дизайнер, прежде чем начать работу должен точно знать, какой будет сайт: фиксированный или резиновый.

После того, как Заказчик утвердил дизайн-макет (в письменном виде или по электронной почте) за работу принимается верстальщик – это тот человек, который переводит дизайн-макет на язык, понятный компьютеру с использованием языка HTML.

Очень часто этап программирования и верстки объединяют в один. На мелких и средний проектах оба действия в состоянии выполнить один человек. На крупных проектах в силу специфики работ эти этапы разделяют.

На этапе программирования (как правило, с использованием DHTML, PHP, Perl, ASP и баз данных) происходит создание всех страниц сайта, определяется порядок работы меню, расставляются гипер-ссылки, создается динамика на сайте, программируются такие составляющие, как гостевая книга, форум, новостная лента и проч.

Если сайт должен иметь администраторский интерфейс то он создается именно на этапе программирования.

Очень важно на этапе программирования определить, на какой системе администрирования (Content Management Site) будет работать ваш сайт. На сегодня на рынке существует великое множество систем администрирования. Для того, чтобы правильно выбрать систему надо ответить себе на ряд вопросов.

В результате работы верстальщика и программиста получается так называемая «рыба» сайта, т.е. сайт без информационного наполнения. Физически сайт в таком виде представляет из себя набор файлов.

Сроки выполнения работ по верстке и программированию зависят от сложности проекта и могут составлять от недели до 2-х месяцев.

На этапе наполнения сайта информацией информация, предоставленная Заказчиком, размещается на сайте, т.е. путем перевода в специальный формат текст и графика располагаются на сайте на определенных страницах, и эта информация становится доступной для просмотра.

Срок исполнения опять же зависит от сложности проекта, объема информации, который надо расположить на сайте, и того вида, в котором Заказчик ее представил. Если информация представлена в электронном виде, а графика не требует дополнительной обработки (например, изменение размера, добавление дополнительных элементов и проч.), то наполнение сайта происходит достаточно быстро. Если необходима дополнительная работа, как например, поиск или набор текста, сканирование фотографий или создание рисунков, то этап наполнения сайта может стать одним из самых длительных.

Этап расположения сайта в сети Интернет. необязательно выполняется после проведения всех вышеперечисленных работ. Он может проводиться параллельно с любым из этапов. Он заключается в том, что файлы сайта располагаются на хостинге.

Тестирование сайта можно осуществить как до, так и после размещения сайта по его «родному» адресу. На этом этапе выявляются все ошибки и недочеты в программировании и написании текстов. Срок тестирования зависит от сложности проекта, но, как правило, не превышает 1 месяца.

Для реализации сайта необходимо применять упорядоченные этапы, которые в конечном итоге приведут к запуску готового сайта.

Перед созданием веб-сайта рекомендуется выделить время на анализ, планирование и проектирование системы, которая будет отвечать за обслуживание веб-сайта. План должен охватывать не только веб-сайт, но и систему веб-материалов, отвечающую за управление веб-сайтом, а также инфраструктуру, необходимую для обеспечения работы системы. Кроме того, необходимо заранее определить роли и пользователей, принимающих участие в создании веб-сайта и установке инфраструктуры.

Визуально этапы планирования разработки веб-сайта показано на рисунке 1

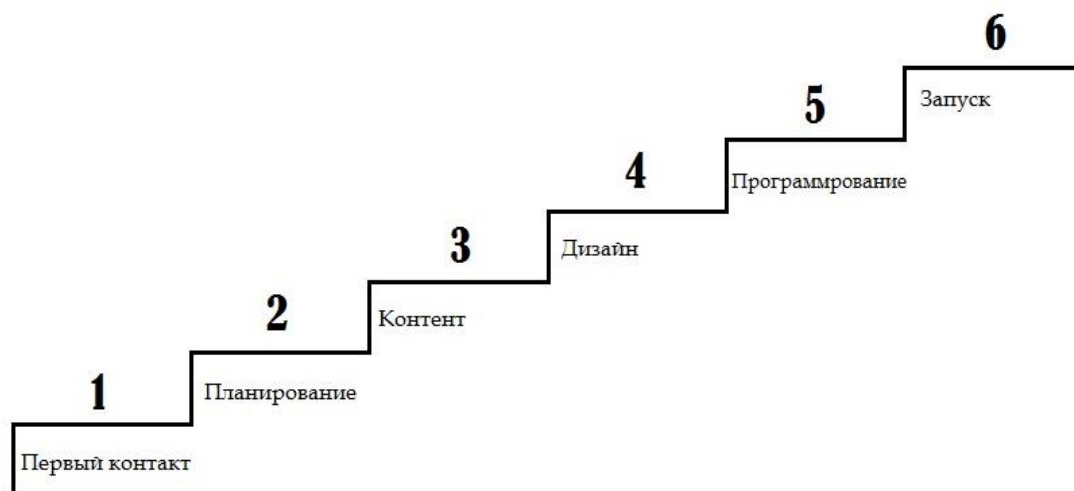


Рисунок 1 – Этапы разработки веб-сайта

Первый контакт – постановка цели и задач;

Планирование – как, кем, чем и в какие сроки будут проектироваться отдельные части сайта;

Контент – содержимое сайта, информация которая будет находиться в нем;

Дизайн – разработка интерфейса;

Программирование – реализация сайта на язык html, css, JavaScript и др.;

Запуск – активация готового сайта.

### 1.3 Анализ методов и технологий создания сайта

Сейчас, у разработчиков сайта есть немало способов создания сайта, а именно, при вопросе, какой же язык программирования применить для разработки вебсайта. Есть большое количество разновидностей: Perl, PHP, ASP, ASP.Net, JSP и др. Наиболее распространенными сейчас считаются PHP и HTML.

Свое начало World Wide Web положил в марте 1989 г., когда Тим Бернерс-Ли из CERN предложил руководству этого международного европейского научного центра концепцию новой распределенной информационной системы, которую назвал World Wide Web.

Программное обеспечение WWW предусматривало наличие двух основных составляющих: WWW-сервера и WWW-клиента. Последний называется «браузером».

В начале 1992 г. Марк Андрессен из Национального Центра Суперкомпьютерных Приложений (NCSA) построил на принципах W3 программу MOSAIC - мультипротокольный переносимый интерфейс, который обеспечил бурный рост популярности этого вида сервиса.

К 1995 году были разработаны программы-браузеры Netscape, Arena, Chimera. В последствии к ним добавилась Microsoft Internet Explorer. С начала 1995 г. практически все компьютерные издания обратили внимание на этот вид сервиса: наконец появился удобный интерфейс для работы в Internet.

При создании новых браузеров цель их создания была расширена - они должны были реализовать многопротокольную работу: не только в WWW, но и в FTP, Telnet, Gopher, WAIS и др. Благодаря этому в более поздних браузерах, работая в WWW, можно использовать практически все сервисы Internet.

Бернерс-Ли считал, что информационная система, построенная на принципах гипертекста, должна объединить все множество информационных ресурсов CERN, которое состояло из базы данных отчетов, компьютерной документации, списков почтовых адресов, информационной реферативной системы, наборов данных, полученных в экспериментах, и т.п. Гипертекстовая технология должна была позволить легко перемещаться из одного документа в другой.

Для реализации гипертекстовой технологии был выбран язык HTML. В WWW программа-клиент - это интерпретатор HTML. В зависимости от команд HTML программа выполняет различные функции: размещение текста на экране; обмен информацией с сервером по мере анализа полученного HTML-текста или графических образов; запуск внешних программ для работы с документами в форматах, отличных от HTML (GIF, JPEG, MPEG, Postscript и др.).

Наибольшее распространение получили Web-браузеры Microsoft Internet Explorer и Netscape Navigator (Netscape Communicator).

Сервер в WWW работает по протоколу HTTP. Он обеспечивает работу WWW, управляет имеющимися базами данных документов в формате HTML и программным обеспечением, разработанным в стандарте CGI. Получили известность такие серверы HTTP, как Netsite (Netscape), Apache (Apache Group), Open Market Secure Web Server (Open Market), NCSA HTTP (NCSA), Netscape Enterprise Server (Netscape), O'Reilly Website (O'Reilly), Internet Information Server (Microsoft), Personal Web Server (Microsoft).

Сервис WWW позволяет организовать огромное количество информации в компактное хранилище с легким доступом к имеющимся материалам.

Наименьшей единицей информации в сервисе W3 является Web-страница. Она может иметь разные размеры - от одного экрана до бесконечного их количества. Главная особенность Web-страницы - наличие на ней гипертекстовых ссылок, активизация которых может перевести на другую часть Web-страницы или на другую страницу.

Совокупность Web-страниц, объединенных единой темой и организованных для совместной работы, образует Web-сайт, или Web-представительство, или просто «Web».

Каждая Web-страница хранится в файле, имя которого имеет расширение «htm» (для MS DOS и Windows - платформ) или «html» (для Unix - платформ). По этому расширению находят Web-страницы программы-клиенты сервиса WWW-браузеры (просмотрщики Web-страниц). Причем, браузеры понимают оба эти расширения страницы делятся на два вида: пассивные и активные. Пассивные Web-страницы содержат только информационное наполнение (текст, графику). Причем для хранения в ЭВМ графика оформляется в виде отдельных файлов (в основном - в формате GIF, JPEG или PNG). Сама же Web-страница (т.е. файл с расширением «htm») содержит только размеченный текст.

Активные Web-страницы кроме информационного наполнения содержат программы, с помощью которых принимается информация от посетителей (интерактивная Web-страница), осуществляется связь с СУБД (динамически обновляемая Web-страница), выводится на экран в пространстве Web-страницы баннерная реклама (Web-страница с баннерами), реализуются некоторые спецэффекты, например, использующие скроллинг, и др. сайт кроме информационных страниц обязательно содержит корневую страницу - home page («домашнюю страницу»). Она имеет в Интернет стандартное имя: index.html или index.htm сайт, который кроме домашней страницы не имеет информационных страниц, а вместо домашней страницы создана только ее заготовка, называется «пустой Web».

Создание Web-страницы заключается в разметке содержащегося на ней текста, подготовленного любым текстовым редактором. Другие виды информации (графика, звук, анимация) готовятся соответствующими

редакторами и оформляются в виде файлов с соответствующими расширениями, стандартными для используемого браузера. Связь между файлами, используемыми на Web-странице, осуществляется с помощью гиперссылок.

Разметка заключается в форматировании текста - определении его места на странице и способа отображения, в указании, в какое место страницы должна быть вставлена графика, как она соотносится с текстом (вставляется на свободное место, является фоном, на который наносится текст, «обтекается» текстом и т.д.), и вставке гиперссылок, осуществляющих переход на другие Web-страницы или на используемые программы. Для разметки используется язык HTML.

Важнейшими элементами Web-страниц являются информационная часть и гиперссылки.

Каждая гиперссылка состоит из двух частей: названия и адреса. Названием гиперссылки может быть текст или рисунок. При наведении курсора мыши на название гиперссылки, курсор меняет форму на указательный палец. Адресная часть гиперссылки становится видна, если после изменения формы курсора нажать правую кнопку мыши. Щелчок левой кнопкой приводит к переходу по адресу, указанному в гиперссылке.

Одностраничные сайты почти никогда не используются в бизнесе, так как выводимая на сайт информация обычно занимает более одного экрана, и для ознакомления с нею в одностраничных сайтах приходится пользоваться полосами прокрутки, а это крайне нежелательно для экранных документов.

Чтобы создать многостраничный сайт, надо создать несколько страниц и соединить их гиперссылками. Гиперссылки позволяют создавать навигационную структуру многостраничного сайта, т.е. определять, куда, в какие части сайта можно попасть с каждой его страницы.

В сети Internet объединено множество компьютеров различных типов. Эти компьютеры могут использовать разные операционные системы, но все они должны поддерживать принятый для обмена данными в Internet стандарт реализованный на базе стека протоколов TCP/IP (Transmission Control Protocol / Internet Protocol - Протокол управления передачей / Протокол Internet). Стек протоколов - разделенный на уровни набор протоколов, которые работают совместно, реализуя определенную коммуникационную архитектуру. Обычно задачи того или иного уровня реализуются одним или несколькими протоколами. Стек протоколов TCP/IP называют набор сетевых протоколов, используемых в технологии интернет. В этом стеке различают несколько уровней, и протоколы высокого уровня всегда базируются на протоколах более низких уровней. Нижними являются протоколы физического и канального уровней. Например, протокол Ethernet, описывающий передачу данных витой паре.

Выше идёт сетевой уровень, где находится протокол IP, описывающий структуру сети и доставку пакетов. Ещё выше - транспортный уровень, где находится протокол TCP, использующийся для передачи данных. На самом



верху находится множество протоколов прикладного уровня, выполняющих конкретные прикладные задачи. Обычно они программируются в отдельных приложениях. Например: HTTP, FTP, Telnet и т.д.

Язык гипертекстовой разметки HTML был предложен Тимом Бернерсом-Ли в 1989 г. К моменту создания HTML существовал стандарт языка разметки печатных документов - SGML (Standart Generalised Markup Language), который и был взят в качестве основы HTML. Предполагалось, что такое решение поможет использовать существующее программное обеспечение для интерпретации нового языка.

В качестве элемента гипертекстовой базы данных для HTML был выбран обычный текстовый файл, который хранится средствами файловой системы операционной среды ЭВМ.

Таким образом, гипертекстовая база данных в концепции WWW - это набор текстовых файлов, написанных на языке HTML, который определяет:

- форму представления информации (разметка);
- структуру связей файлов между собой (гипертекстовые ссылки).

Такой подход предполагает наличие еще одной компоненты технологии - интерпретатора языка. В WWW функции интерпретатора разделены между сервером гипертекстовой базы данных и интерфейсом пользователя. Сервер, кроме доступа к документам и обработки гипертекстовых ссылок, осуществляет также препроцессорную обработку документов, тогда как интерфейс пользователя осуществляет интерпретацию конструкций языка, связанных с представлением информации.

В 1990 г. появилась нулевая версия HTML 0. Первая версия (HTML 1.0) была направлена на представление языка как такового, где описание его возможностей носило скорее рекомендательный характер.

Вторая версия (HTML 2.0) фиксировала практику использования конструкций языка и явилась стандартом для построения WWW.

Версия ++HTML++ представляла новые возможности по представлению научной информации и таблиц, улучшения стиля компоновки текста и изображений.

Третья версия (HTML 3.0) упорядочила все нововведения и формализовала интерфейс пользователя гипертекстовой распределенной системы.

В 1996 г. выпущена новая версия языка HTML 3.2, включающая средства для создания таблиц, конструкции языка Java, карты-изображения (располагающиеся на машине-клиенте) и новые типы форматирования текста.- это текстовый язык, в котором инструкции по форматированию, называемые тэгами (tags), встроены в текст документа. Процесс создания HTML-документа заключается во включении тэгов внутрь неформатированного текста, несущего информационную нагрузку.

Тэги HTML - это последовательности символов, начинающиеся знаком «меньше» (<) и заканчивающиеся знаком «больше» (>).

Большинство функций форматирования в HTML специфицирует логические, а не физические стили. Например, тэги заголовков, которые обычно обозначают больший размер букв, не указывают, какой именно размер надо использовать. Браузер (который выполняет роль клиентской программы) выбирает размер текста для заголовков, превышающий обычный. Этот подход позволяет избавиться от платформы - документ, созданный на Macintosh может быть использован UNIX или Windows. Недостатком является то, что на разных платформах документ выглядит по-разному.

Браузеры WWW при анализе предложений HTML работают по следующим правилам:

- пробелы и другие «невидимые» символы игнорируются;
- все тэги могут быть разделены на группы: базовые тэги, тэги форматирования, структурные тэги, указатели и т.д.;
- тэги форматирования могут быть написаны строчными и / или прописными буквами;
- большинство тэгов форматирования пишутся парами. Открывающий тэг активирует данный эффект, а закрывающий выключает его. Пара тэгов иногда называется контейнером, поскольку эффект, который они включают и выключают проявляется на тексте, расположенном внутри них. Например, для того, чтобы указать, что строка должна быть напечатана жирными символами, надо написать: `<B>` Эта строка будет напечатана жирным текстом `</B>` - текст, находящийся внутри закрывающего тэга, всегда предваряется символом «обратная косая черта». Среди базовых тэгов HTML исключением из этого правила являются: `<BASE>` (основная информация), `<BR>` (конец строки), `<HR>` (горизонтальная линейка), и `<IMG>` (изображение). Эти тэги существуют в единичном экземпляре, т.е. не требуют закрывающего тэга.

Каждый документ HTML имеет граничные тэги: `<HTML>` и `</HTML>`. Эта пара тэгов заключает в себе все другие тэги Web-страницы и все информационное содержание документа. Фактически они декларируют, что все содержимое контейнера является кодом HTML. Тэг `<HTML>` должен располагаться в первой строке файла, а тэг `</HTML>` - в последней.

Каждый HTML-документ содержит заголовок документа и его тело.

Заголовок документа идентифицирует документ. Он должен идти сразу после тэга `<HTML>` и начинаться с `<HEAD>`. Заканчивается заголовок тэгом `</HEAD>`.

В заголовке документа обязательно должны присутствовать название документа и его базовый URL-адрес.

Название документа располагается между тэгами `<TITLE>` и `</TITLE>`. Оно используется браузером и Web-сервером. Длина названия не должна превышать 40 символов. Название документа должно быть информативным, отражать содержание документа. Название встречается в строке заголовка Internet Explorer, в списках журнала (History) и «Избранных страницах» (Favorites).

Базовый URL должен быть полным (абсолютным) и содержать протокол, адрес компьютера в Internet и имя файла. Он указывается в тэге `<BASE HREF = «базовый_URL»>`. Все остальные URL в документе могут указываться относительно базового. Например, если базовый URL имеет адрес: `http://www.microsoft.com/products/office/word/datesheets.html` и необходимо указать URL файла `features.html`, расположенного в каталоге `/products/office/excel` на том же сервере, можно указать: `../excel/features.html`, а не набирать полный адрес.

Символы указывают, что нужно перейти на один уровень каталогов вверх относительно базового адреса.

Наличие базового адреса облегчает перенос документа на другой сервер, так как если все остальные URL в документе относительные, то при переносе необходимо изменить только имя сервера в базовом URL.

Тело документа содержит весь текст, несущий информацию и все тэги HTML, используемые для форматирования текста. Тело документа начинается сразу же после заголовка и ограничивается тэгами `<BODY>` и `</BODY>`.

Пример HTML-документа:

```
<HTML>
<HEAD>
<TITLE> Название документа </TITLE> <BASE HREF =
«базовый_URL» > </HEAD> <BODY>
</BODY> </HTML>
```

Тэги, используемые в теле документа, позволяют указывать следующие элементы разметки текста:

- начало абзаца и конец строки;
- стили заголовков;
- физические стили;
- логические стили;
- списки;
- специальные символы.

Гипертекстовые ссылки состоят из двух частей: указателя и URL. Указатель - это текст, по которому пользователь должен щелкнуть для того, чтобы куда-то переместиться. URL указывает адрес, с которого браузер будет загружать документ, когда пользователь щелкнет по указателю.

Текстовые указатели в графических браузерах обычно подчеркнуты и выделены цветом. В чисто текстовых браузерах они выделяются жирным шрифтом.

В качестве указателя может использоваться любой текст, независимо от его размеров и особенностей форматирования. Указатель может состоять из нескольких букв, слов или даже строк текста.

HTML - это язык программирования, который предназначен для создания сайта. Веб-сайты, созданные на языке html могут просматривать все пользователи всемирной сети Internet. Этот язык является наиболее легким и доступным в использовании для большинства людей. Этот язык не стоит на

месте, он постоянно улучшается и эволюционирует, чтобы удовлетворять требованиям современного информационного мира под руководством «W3C» организации, которая занимается разработкой и поддержкой языков программирования.

HTML - это гипертекстовый язык разметки.

HyperText - это метод, с помощью которого можно переходить из одной страницы в сети в другую - нажав на специальный текст, называемый гиперссылками, который приведет вас к следующей странице. По одному клику можно посетить различные места в интернете, нет какого-либо определенного установленного порядка, который может запретить вход на какую-нибудь страницу.

Метка - это символьное имя, которое используется в языке, путем проставления тегов. Она отмечают его как определенный тип текста (например, курсивом).

HTML состоит из тегов - серии кодов, введенных в текстовый файл автором сайта. Затем текст сохраняется как html-файл и просматривается через браузер, например Internet Explorer, Google Chrome, Opera, Mozilla Firefox и др. Страница в браузере открывается, показывая весь написанный код в визуальном виде, точнее то, что автор и хотел получить прописывая коды.

HTML-теги – основа языка HTML. Теги используются для обозначения начала и конца элементов в разметке. Теги ограничиваются <угловыми скобками>. Теги позволяют обозначать в коде изображения, таблицы, абзац, параметры шрифта и прочее. Существует ряд тегов, предназначенных для различных функций соответственно.



Рисунок 2 – Взаимосвязь HTML с CSS и JavaScript

Каскадные таблицы стилей CSS появились гораздо позже HTML - только в 1997 году. Несмотря на то, что CSS работает с HTML, это не HTML. Более того, CSS представляет собой отдельный код, который расширяет

возможности HTML, позволяя переопределять уже существующие HTML-теги.

Связано это с тем, что WWW была создана как средство для обмена текстовыми документами, а HTML был языком, с помощью которого создавались эти документы. Для военных и ученых главным было содержимое документ, а не его оформление. Поэтому первые версии HTML не включали даже средств для размещения на веб-страницах графических изображений, тем более для сложного форматирования текста. Но со временем в интернет пришли остальные люди и потребовали, чтобы им «сделали красиво».

Так возникли каскадные таблицы стилей, созданные для облегчения работы веб-дизайнеров. В настоящее время приняты CSS1 и CSS2. CSS третьего уровня (CSS3) еще находится в стадии разработки. Но даже после ее выпуска браузеры только через несколько лет начнут ее поддерживать. Самым важным отличием CSS3 от предыдущих версий станет масштабируемая векторная графика. То есть все преимущества векторной графики будут доступны теперь и в интернете.

Но даже таблицы стилей первого уровня (CSS1) предлагают удивительные возможности. Куда там HTML с его примитивным форматированием!

Каскадные (многоуровневые) таблицы стилей - cascading style sheets (CSS) - это мощный стандарт на основе текстового формата, определяющий представление данных в браузере.

Если формат HTML предоставляет информацию о составе документа, то таблицы стилей сообщают как он должен выглядеть. Таким образом каскадные таблицы стилей дают возможность хранить содержимое отдельно от его представления.

Стиль включает все типы элементов дизайна: шрифт, фон, текст, цвета ссылок, поля и расположение объектов на странице разрабатывались так, чтобы обеспечить больший уровень контроля над размещением текста и графики.

Каскадные таблицы стилей обеспечивают должный уровень единства оформления, организации и контроля во время разработки узла, который является недостижимым с помощью одного только HTML предполагает 3 типа таблиц стилей - встроенные, внедренные (внутренние) и связанные (внешние).

Впервые идея форматирования HTML-документов с помощью CSS была рекомендована Консорциумом W3C в 1996 году. Эта рекомендация, которая была обновлена в 1998 году, используется Web-разработчиками и по сей день.

Термин «каскадный» означает, что в одной странице HTML могут использоваться разные стили. Браузер, поддерживающий таблицы стилей, будет следовать их порядку (как по каскаду), интерпретируя информацию стилей. Это означает, что вы можете использовать все три типа стилей, и браузер будет интерпретировать сначала связанные, затем внедренные и, наконец, встроенные стили. Даже если ко всему узлу будут применены

образцы стилей, можно будет управлять отдельными аспектами страниц с помощью внедренных стилей, а отдельными областями внутри этих страниц - с помощью встроенных стилей.

Другой аспект каскадирования - наследование (inheritance). Наследование означает, что если не указано иное, то конкретный стиль будет унаследован другими элементами страницы HTML. Например, если вы примените определенный цвет текста в теге <p>, то все теги внутри этого абзаца наследуют этот цвет, если не оговорено иное.

Существует ряд методов, с помощью которых таблицы стилей могут применяться к документу HTML. Синтаксис соответствует реальной структуре информации, содержащейся внутри таблицы стиля.

Существует три метода для применения таблицы стилей к документу HTML:

- Встроенный (Inline). Этот метод позволяет взять любой тег HTML и добавить к нему стиль. Использование встроенного метода предоставляет максимальный контроль над всеми свойствами Web-страницы. Предположим, вы хотите задать внешний вид отдельного абзаца. Вы можете просто добавить атрибут style к тегу абзаца, и браузер отобразит этот абзац с помощью параметров стиля, добавленного в код.

- Внедренный (Embedded). Внедрение позволяет контролировать всю страницу HTML. При использовании тега <style>, помещенного внутри раздела <head> страницы HTML, в код вставляются детализированные атрибуты стиля, которые будут применяться ко всей странице.

- Связанный (Linked или External). Связанная таблица стилей - мощный инструмент, который позволяет создавать образцы стилей (master styles), которые можно затем применять ко всему узлу. Основным документом таблицы стилей (расширение.css) создается Web-дизайнером. Этот документ содержит стили, которые будут едиными для всего Web-узла (неважно, содержит одну страницу или тысячи страниц). Любая страница, связанная с этим документом, будет использовать указанные стили.

Таблицы стилей строятся в соответствии с определенным порядком, в противном случае они не могут нормально работать.

Синтаксис всех методов, используемых для применения стилей к документу HTML, практически одинаков. Таблицы стилей состояются из определенных частей. Эти части включают следующие элементы:

- Указатель (Selector). Указатель является элементом, к которому будут применяться назначаемые вами атрибуты. Это может быть просто тег типа заголовка (H1) или абзаца (P). Таблицы стилей позволяют использовать различные объекты, включая классы, которые будут кратко обсуждаться далее.

- Свойство (Property). Свойство определяет указатель. Например, если в качестве указателя выбран абзац, вы можете включить свойства, определяющие этот указатель. В свойства входят такие элементы, как поля,

шрифты и фоновые изображения. В таблицах стилей существует много свойств, которые можно использовать для того, чтобы определить указатель.

– Значение (Value). Значения определяют свойства. Предположим, что у вас есть заголовок первого уровня H1 (указатель) и вы включаете свойство `type family` (семейство шрифта). Шрифт, который на самом деле будет применен к указанному фрагменту, задается значением этого свойства.

Итак, какскадная таблица стилей - это набор правил форматирования тегов HTML. Например, для того, чтобы цвет фона Web-страницы сделать черным, необходимо объявить следующее правило форматирования: `{background:black}`. Свойства CSS должны находиться в фигурных скобках.

Для каждого тега HTML можно указать не одно, а несколько свойств стиля.

Изменим с помощью CSS не только цвет фона Web-страницы, но и цвет шрифта (на белый): `body {background:black; color:white}`.

Формат самого правила не имеет значения, главное - правило должно читаться удобно и легко. Например, вышеприведенное правило можно записать и так: `{:black;:white}`.

Одно и то же правило стиля можно применить сразу к нескольким различным тегам HTML-страницы. Например: `td, h1 {:black;:white}`.

Встроенный стиль применяется к любому тегу HTML с помощью атрибута `style` следующим образом:

`<P style= «font: 12pt Courier New»> The text in this line willas 12 point text using the Courier New font.`

`</P>`

Или:

`<p style= «font: 12pt Arial»>`

The text in this line will display as 18 point text using the

Arial font.

`</p>`

Можно добавлять встроенный стиль в любой тег HTML, в котором эта операция будет иметь смысл. Среди таких тегов можно назвать абзацы, заголовки, горизонтальные линии, якоря и ячейки таблиц. Ко всем этим элементам логично применять встроенные стили.

Существуют два тега, которые помогают применять встроенные стили к разделам страницы. Это теги `<div>` (division - раздел) и `<span>` (промежуток). Эти теги определяют диапазон текста, так что все, находящееся между ними, будет оформлено с помощью нужного стиля. Единственное различие между `<div>` и `<span>` состоит в том, что `<div>` создает принудительный разрыв строки, а `<span>` - нет.

Следовательно, нужно использовать `<span>` для изменения стиля любой части текста, меньшей абзаца.

Внедренный стиль

Внедренные стили используют тег `<style>`, расположенный между тегами `</head>` и `<body>` в стандартном документе HTML.

Рассмотрим пример внедренного стиля:

```
<html>
<head>
<title>Embedded Style Sheet Example </title>
</head>
<style>{: #FFFFFF;: #000000;
} {: 14pt verdana; color: #CCCCCC;} {font: 12pt times;} {color: #FF0000;
text-decoration: none}
</style>
```

Как видно из примера, приведенного выше, таблица стилей теперь отличается от кода стандартной страницы HTML, но все же логику проследить нетрудно. В нашем случае для основной части страницы (body) указаны цвет фона, цвет текста и верхнее, левое и правое поля в дюймах.

Для заголовка первого уровня (H1) указывается шрифт (название шрифта и размер в пунктах). В этом и состоит удобство каскадных таблиц стилей - можно задавать размеры не только в пунктах, но и пикселях (px), процентах (75%) и сантиметрах (см).

Существует и несколько новых единиц измерения, самой примечательной из которых является m.

В отличие от пунктов или пикселей, представляющих абсолютные размеры различных объектов, 1m - ширина строчной буквы m в том шрифте о котором идет речь. Правда, большинство браузеров об этом не догадываются и поэтому определяют 1m просто как размер, заданный по умолчанию. Например, в Internet Explorer 4.0 и выше для гарнитуры Verdana размер 1m соответствует 12 пунктам. Точнее сказать «соответствует значению, которое пользователь установил в качестве размера шрифта по умолчанию. Таким образом, если установить в браузере размер шрифта по умолчанию 16 пунктам, то и 1m тоже станет равна 16 пунктам, и все остальные размеры будут соответственно увеличены.

JavaScript – это язык программирования, который добавляет интерактивность на веб-сайт (например: игры, отклик при нажатии кнопок или при вводе данных в формы, динамические стили, анимация).

JavaScript – язык программирования, являющийся прототипно-ориентированным. Он отражает язык ECMAScript, чьим прототипом изначально и являлся. Первая вариация появилась ещё в 1995 году и с тех пор постоянно совершенствовалась, пока не пришла к нынешнему виду.

Чаще этот язык используется в разработке приложений и браузерах с целью придания им интерактивности и «живости».

Базовой особенностью этого языка отмечается то, что на него повлияли другие (Python, Java и др.) языки программирования с целью придания максимального комфорта JavaScript и лёгкости в понимании его теми пользователями, которые не имеют соответствующего образования и глубинных знаний – не программистами. JavaScript – официально зарегистрированная торговая марка компании Oracle.



С помощью него доступны к исполнению следующие функции:

- возможность изменять страницы браузеров;
- добавление или удаление тегов;
- изменение стилей страницы;
- информация о действиях пользователя на странице;
- запрос доступа к случайной части исходного кода страницы;
- внесение изменений в этот код;
- выполнение действия с cookie-файлами.

Область применения этого языка удивительно обширна и ничем не ограничена: среди программ, которые используют JS, присутствуют и тестовые редакторы, и приложения (как для компьютеров, так и мобильные и даже серверные), и прикладное ПО.

Преимущества JavaScript:

- Ни один современный браузер не обходится без поддержки JavaScript.
- С использованием написанных на JavaScript плагинов и скриптов справится даже не специалист.
- Полезные функциональные настройки.
- Постоянно совершенствующийся язык – сейчас разрабатывается бета-вариация проекта, JavaScript2.
- Взаимодействие с приложением может осуществляться даже через текстовые редакторы – Microsoft Office и Open Office.
- Перспектива использования языка в процессе обучения программированию и информатике.

Недостатки JavaScript:

- Пониженный уровень безопасности ввиду повсеместного и свободного доступа к исходным кодам популярных скриптов.
- Множество мелких раздражающих ошибок на каждом этапе работы. Большая часть из них легко исправляется, но их наличие позволяет считать этот язык менее профессиональным, сравнительно с другими.
- Повсеместное распространение. Своеобразным недостатком можно считать тот факт, что часть активно используемых программ (особенно приложений) перестанут существовать при отсутствии языка, поскольку целиком базируются на нем.

JavaScript - новый язык программирования, который используется как часть HTML страницы с целью повышения функциональности и взаимодействия с пользователем. Она была разработана Netscape в сотрудничестве с Sun Microsystems на базе Java от Sun. С другой стороны, JavaScript - это текстовый язык программирования, предназначенный для работы как часть веб-приложения. Когда он был впервые разработан, он должен был стать дополнением к Java. Но JavaScript занял свою собственную позицию, как один из трех этапов веб-разработки, а два других - HTML и CSS. В отличие от приложений Java, которые необходимо скомпилировать, прежде чем они смогут запускаться в веб-среде, JavaScript был специально разработан для интеграции в HTML. Все основные веб-браузеры

поддерживают JavaScript, хотя большинство предоставляют пользователям возможность отключить поддержку для него.

Этот интерпретируемый язык не требует специальной программы для создания кода. Для написания JavaScript можно использовать текстовый редактор Блокнот, Markdown Editor и др.

Исследование возможностей ввода и вывода информации - это не только замечательный повод научиться создавать сценарии JavaScript, обеспечивающие двустороннюю связь с пользователем, но и прекрасное начало вашего профессионального пути.

В вычислительной технике все основано на вводе и выводе данных. Без этого ничего не происходит. Текстовый процессор не выполняет никаких действий, пока пользователь не введет какую-либо информацию (символы, набираемые на клавиатуре), и лишь затем эта информация отображается на экране, распечатывается или сохраняется на жестком диске.

Говоря простым языком, объект (object) - это какой-либо предмет.

Что касается JavaScript, его объекты находятся внутри браузера. Это, в частности, окно браузера, формы и их части, например кнопки и текстовые окна. В JavaScript также имеется собственная группа встроенных объектов, к которым относятся массивы, данные и т.д. Сейчас вам не обязательно фиксировать на этом внимание, поскольку все эти объекты будут рассмотрены позже. Пока вы должны усвоить лишь необходимые определения.

Именно благодаря наличию объектов язык JavaScript считается объектно-ориентированным. Язык организован вокруг объектов, а не действий, или, иначе говоря, ориентирован на данные, а не на логику. При объектно-ориентированном программировании первоочередное внимание уделяется объектам, с которыми производятся некоторые манипуляции, а не логическим правилам, необходимым для таких манипуляций. Преимуществом такого подхода является не только облегчение программирования (или написания сценария), но и в то, что каждое действие можно выполнить разными способами.

## **2 Проектирование web – ориентированной информационной системы**

### **2.1 Описание административной части**

Для создания и полного функционирования сайта было использовано следующее программное обеспечение:

- операционная система Windows 7 Профессиональная;
- текстовый редактор NotePad++ версии 6.6 для создания и редактирования HTML, CSS и JavaScript кода;
- Wordpress – система управления содержимым сайта;
- графический редактор Adobe Photoshop CS5;
- браузер Google Chrome 43.0.2357.81m.

Программная реализация кода была написана на серверном языке HTML. Конструкция основана на языке HTML, каскадные таблицы стилей (CSS) и библиотек JavaScript.

Windows 7 - операционная система, которая принадлежит к семейству операционных систем Microsoft Windows. Согласно Microsoft, Windows 7 работает быстрее, надежнее и совместима с предыдущей версией под названием Windows Vista.

Бета-версия Windows 7 была выпущена 9 января 2009 года, и релиз был выпущен 5 мая. Окончательная версия Windows 7 была отправлена разработчикам ПК 22 июля 2009 года. Официально Windows 7 была презентована 22 октября 2009 года.

С июля 2011 до марта 2017 года Windows 7 занимала лидирующее положение по количеству пользователей в мире. По состоянию на октябрь 2017 года доля Windows 7 составляет 26,1 % среди используемых в мире операционных систем для доступа к сети Интернет и занимает второе место в мире по популярности после Windows 10.

Notepad ++ - это текстовый редактор и редактор исходного кода для использования с Microsoft Windows. Он может открывать несколько вкладок, что позволяет работать с несколькими открытыми файлами в одном окне. Название проекта происходит от оператора инкремента C.

Notepad ++ распространяется как бесплатное программное обеспечение. Сначала проект был размещен на SourceForge.net, откуда он был загружен более 28 миллионов раз, и дважды выиграл награду SourceForge Community Choice Award за лучший инструмент для разработчиков. Проект был организован на TuxFamily с 2010 по 2015 год; с 2015 года Notepad ++ был размещен на GitHub. Notepad ++ использует компонент редактора Scintilla.

Adobe Photoshop - это редактор растровой графики, разработанный и опубликованный Adobe Systems для MacOS и Windows.

Photoshop был создан в 1988 году Томасом и Джоном Кнолом. С тех пор он стал фактическим отраслевым стандартом в редактировании растровой графики. Он может редактировать и компоновать растровые изображения в

нескольких слоях и поддерживает маски, альфа-композицию и несколько цветовых моделей, включая RGB, CMYK, CIELAB, цвет пятна и дуот. Photoshop имеет обширную поддержку форматов графических файлов, но также использует собственные форматы файлов PSD и PSB, которые поддерживают все вышеупомянутые функции. В дополнение к растровой графике он имеет ограниченные возможности для редактирования или рендеринга текста, векторной графики (особенно путем обтравочного контура), 3D-графики и видео. Набор функций Photoshop можно расширить с помощью плагинов Photoshop, разработанных и распространяемых программ независимо от Photoshop, которые могут работать внутри него и предлагать новые или улучшенные функции.

## **2.2 Структура сайта**

Архитектура сайта - это часть систематизации и навигации по ней. Главная идея заключается в том, чтобы клиенты успешно находили нужную им информацию, а также могли связаться с сотрудниками компании. Хорошо продуманная архитектура сайта гарантирует, что клиенты будут тратить наименьшее количество своего времени на поиск нужной информации.

Архитектура сайта – понятие, которое неразрывно связано с контентом и оптимизацией. Под архитектурой понимают систему организации файлов, страниц и просто информации, содержащейся на сайте. Если такой системы нет или она была составлена неправильно еще при создании ресурса, в дальнейшем это может серьезно усложнить жизнь посетителям сайта.

Главная цель архитектуры сайта – это создание ресурса, который бы позволил пользователю быстро найти необходимую информацию. Даже если посетитель впервые заглянул на сайт, он должен в течение нескольких минут освоиться и найти нужное. Больше времени на Ваш ресурс посетитель вряд ли потратит: он не будет загружать десятки страниц, чтобы найти нужную информацию, и просто-напросто уйдет на другой сайт. Ведь гораздо проще и быстрее набрать интересующий запрос в поисковой системе, чем искать что-то на одном сайте.

Оптимизация архитектуры сайта включает в себя систематизацию страниц, распределение внутренних ссылок, проставление различных тегов, необходимых для поискового продвижения.

Главная цель оптимизации архитектуры сайта – сделать информацию, хранящуюся на сайте, доступной пользователям и сделать так, чтобы посетитель смог быстро найти необходимые ему сведения. В случае, если речь идет о коммерческом ресурсе, появляются и дополнительные задачи. Например, сделать страницы, приносящие наибольший доход, более доступными и видными посетителям, чем другие.

Работа над оптимизацией архитектуры сайта может идти совместно с разработкой нового дизайна и с раскруткой сайта. Но можно сделать

страницы более доступными для пользователей и без изменения внешнего вида и функционала. Для этого достаточно грамотно распределить ссылки.

На поисковое продвижение влияют как внешние факторы (а именно – количество внешних ссылок), так и внутренние. В число внутренних входит не только качество оптимизации текстового контента, но и архитектура сайта.

В SEO существует такое понятие, как внутренняя перелинковка, т.е. проставление внутренних ссылок (ссылок, ведущих на страницы сайта) на разных страницах. Чем лучше сделана перелинковка, тем быстрее проходит индексация и тем выше показатели остальных страниц (например, PR). Внутреннюю перелинковку можно назвать одной из составляющих архитектуры сайта, но главной ее частью все же являются те ссылки, которыми посетители пользуются, чтобы перейти с одной страницы на другую.

Чем «выше» находится ссылка, тем лучше. В данном случае мы говорим об уровне вложенности страницы, где она находится, а не о самом месторасположении на странице. Так, ссылки, ведущие с главной, считаются поисковыми роботами наиболее важными. Ссылки, которые находятся на второстепенных страницах, уже менее важны, а значит, дают меньшие показатели.

Сайт состоит из отдельных страниц, которые являются частями единого целого. Первоочередной задачей архитектуры сайта является создание удобного интерфейса для пользователя, чтобы ему не составило особого труда найти необходимую информацию, соответствующую тематике сайта.

Грамотная архитектура сайта предполагает наличие в документе ссылок, ведущих на документы этой же тематики. Пользователь всегда сможет легко перейти с поиска документа на его просмотр и обратно.

Архитектура сайта помогает пользователю увидеть все разделы сайта. Ему сразу видно, из каких разделов состоит сайт, и какая информация содержится в каждом из разделов.

Правильно выстроенная архитектура сайта позволяет выделять в глазах поисковых систем наиболее важные страницы. Это могут быть и «Услуги», и «Цены», и «Прайс-лист», а также все те страницы, на которых размещены наиболее востребованные и прибыльные товары и с которых пользователи чаще всего оформляют заказ. Ссылки на такие страницы, как правило, размещают на главной. Это влечет за собой повышенное внимание со стороны поисковых систем и более высокие позиции на страницах результатов поиска.

Необходимо создать и внедрить сайт архитектуры-проекта так, чтобы информативный контент сайта выполнялся на наивысшем уровне удобства использования, то есть связь между пользователем и сайтом, а также в то же время содержать информационную суть. В процессе структуры нового сайта или оптимизации уже существующего клиенты должны заострить свое внимание на управлении важными разделами сайта, которая также будет позволять находить наибо́льший способ поиска той или иной информации.

Структура сайта – основа для выстраивания последовательности и формы отображения имеющихся данных на сайте. При правильной структуре сайта пользователям максимально удобно переходить от одной странички к другой и изучать необходимые для них сведения. Если вы еще не определились какую структуру выбрать для своего сайта — узнайте о разных видах структур сайта и выберите наиболее подходящую для вас.

Самым первым видом структуры сайта является линейный. Он называется так потому, что в нем вся информация представлена в линейном виде. Это как будто цепочка, где все элементы послушно следуют друг за другом. Здесь не может быть разделов, и пользователь не может перескакивать с одной страницы на другую в любой последовательности, он получает возможность лишь плавно переходить по заданной траектории, так, как будто он читает книгу. Данный вид структуры сайта отлично бы подошел для создания презентаций либо онлайн-книг, его очень просто реализовать. Навигация здесь также присутствует минимальная, на каждой странице нужно обозначить лишь три основные ссылки: на главную страницу, на предшествующую и последующую.

В качестве расширенной формы этого вида структуры сайта представлен линейный с ответвлениями. Он содержит несколько веток и на главной странице посетитель может выбрать, по какой из них идти дальше, и в зависимости от выбора ему будет представлена та или иная цепочка сведений.

Но самым многофункциональным и развитым можно назвать древовидный вид структуры сайта. Это форма представления данных, выраженная в форме многоуровневой иерархии. То есть, вся структура может содержать большое количество разделов и подразделов, внутренних страниц любой вложенности и в произвольной последовательности. В таком формате можно выполнить совершенно любой веб-ресурс, причем вне зависимости от того, посвящен он одной тематике или сразу нескольким различным областям. Однако здесь следует быть осторожным, ведь вы можете запутать своих посетителей в лабиринте составных частей проекта. Чтобы этого не случилось, при выборе подобного вида структуры сайта следует хорошо позаботиться о подробной и удобной навигации, понятном интерфейсе.

Существует еще один вид структуры сайта – решетчатый. Он выполняется практически по тем же стандартам, что и древовидный, но в его составе есть особая функция, которая помогает в один клик переместиться с одной части веб-проекта в совершенно противоположную. В большинстве случаев такой ход лишь усложняет разработку сайта, так что данный вид структуры сайта применяется лишь в среде разработчиков каталогов.

Структура сайта - это связь документов, принадлежащих ему, между собой. Правильная структура позволит поисковым роботам быстро совершать обход ресурса, а посетителям легко перемещаться между его страницами.

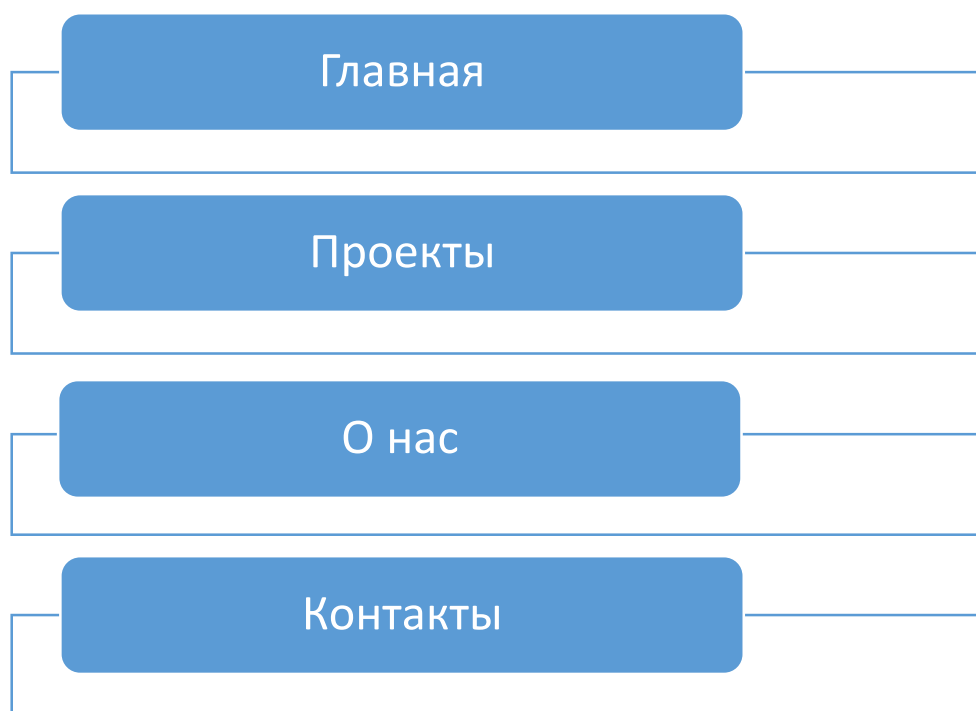


Рисунок 3 – Структура сайта

На рисунке 3 представлена структура сайта. Имеется четыре категории, каждая из которой имеет свою страницу с соответствующей информацией.

Все четыре страницы имеют общие элементы. Например, такие как: логотип, контактные данные, время работы, партнеры и способ связи через электронную почту.

Главная страница содержит в себе:

- меню навигации для перехода в нужный раздел, что сэкономит время клиента;

- краткую информацию о компании, чтобы клиент, пройдя по ссылке на сайт, сразу увидев небольшой текст из достаточно крупного шрифта, понял какова деятельность компании;

- основные функции, предназначенные для того, чтобы также сэкономить время клиента, который увидев эту часть, понял соответствуют ли эти функции его требованиям;

- лозунг. У всех крупных компаний есть свой лозунг, которого они придерживаются. Это дает понять клиенту степень надежности компании;

- каталог товаров, который предназначен для того, чтобы клиент мог увидеть все товары в наличии и их сумму.

Категория «Проекты» посвящена ряду выполненных работ и достижений, по которым клиент может определить качество и уровень стройфирмы. Категория состоит из следующих элементов:

- кнопка «Главная», которая при нажатии, позволяет вернуться на главную страницу;

- список выполненных работ с изображениями, датой выполнения и необходимой информацией об объекте;

- галерея, предназначенная для просмотра изображений компании.

Категория «О нас» является страницей с биографией компании. Она состоит из:

- кнопка «Главная», которая при нажатии, позволяет вернуться на главную страницу;

- информация о строительной фирме, так как у каждой компании есть своя история с даты появления;

- наши проекты – небольшая ссылка на предыдущую категорию;

- галерея, предназначенная для просмотра изображений компании.

Категория «Контакты» это страница с необходимой контактной информацией для того, чтобы клиент мог связаться или пройти по адресу указанной в ней. На странице присутствует:

- кнопка «Главная», которая при нажатии, позволяет вернуться на главную страницу;

- контактная информация, которая содержит телефон, факс, электронную почту и адрес компании;

- google-карта, предназначенная для визуального представления местонахождения фирмы;

- блок «заполните форму» для того, чтобы клиент мог оставить свое сообщение, которое отправится напрямую в адрес электронной почты компании.

### **2.3 Разработка дизайна сайта**

Photoshop - это программа для создания и обработки растровой графики, проще говоря - графический редактор. Растровая графика - это любое изображение, которое состоит из отдельных пикселей разного цвета - фотографии в фотоаппарате, картинки на сайтах, спрайты в играх - вообще, основная масса всей графики относится к растровой. Все это Photoshop прекрасно открывает и предоставляет невероятное количество инструментов для работы с изображением. Сам по себе Photoshop - всего лишь набор инструментов, которые собраны воедино. Но все сделано настолько продумано и аккуратно, что создается настоящая рабочая лаборатория для художника или дизайнера. Есть инструменты по рисованию - различные кисти и заливки, инструменты для вставки и работы с текстом, для векторной графики... Последние версии позволяют даже загружать 3D-модели в популярных форматах, созданных в редакторах трехмерной графики.- настолько удобная и мощная программа, что сегодня практически любой человек, так или иначе имеющий отношение к графике, имеет эту программу на своем компьютере. Веб-дизайнеры разрабатывают в ней всю графику сайтов, программисты - все радующие наш глаз красивые иконки и игровых персонажей, фотографы - корректируют неудачные фотографии, убирая



«красные глаза» и исправляя яркость, контраст или цветовой баланс Photoshop нужен часто и практически всем.

В связи с этим видно, что знать основы Photoshop для того, кто пользуется компьютером не только для игр - так же важно, как уметь набирать текст в Word-е. Эти две программы - одни из самых необходимых в наборе любого мало-мальски грамотного «юзера». А посему требуется осваивать и эту программу. Благо, своим удобством Photoshop завоевал такую любовь пользователей, что созданы тысячи сайтов, посвященных ему, написано сотни книг, как для начинающих, так и для профессионалов, снято сотни часов обучающего видео. Все это легко доступно каждому, притом с любым уровнем подготовки. Даже тот, кто совсем ничего не знает о графике, может освоить ее и многому научиться. Нужно только попробовать познакомиться с этой программой поближе и мир компьютерной графики может увлечь вас надолго.

Растровые изображения формируются из совокупности крошечных элементов, называемых пикселями. Пиксель является основным кирпичиком растровых изображений и это единица принята в компьютерной графике, подобно тому, как метр, килограмм и литр приняты для измерений в повседневной жизни. Количество пикселей в изображении определяет его разрешение. Пиксели часто называют точками, тогда разрешение измеряется в dpi (dot per inch), то есть в количестве точек на дюйм.

Чем выше разрешение, тем большее количество пикселей содержит изображение и тем большим количеством деталей (то есть - качеством) такое изображение характеризуется. С другой стороны, более высокое разрешение изображения прямо связано с большим размером файла такого изображения. Поэтому установка величины разрешения зависит от целей и задач компьютерного художника и для конкретной работы будет разной. Например, веб-дизайнеры обычно работают с изображениями 72-96 dpi, в то время как полиграфисты предпочитают разрешения изображений от 300 dpi и выше (рис.

В черно-белых изображениях уровни яркости представляются в виде оттенков серого цвета, а в цветных изображениях эти уровни проявляются в виде различных цветовых тонов. При этом очевидно, что черно-белая фотография воспринимается как менее качественная по сравнению с фотографией цветной. Иначе говоря, чем больше оттенков цвета в изображении, тем выше его яркостное (цветовое) разрешение, называемое глубиной цвета, и тем большее число уровней яркости (цветов) будет содержать файл такого изображения.

Глубина цвета характеризует число воспроизводимых градаций яркости пикселя в черно-белых изображениях и количество отображаемых цветов в цветном изображении.

С точки зрения цветовой глубины растровые изображения можно разбить на несколько типов:

Для монохромного черно-белого (Black and White) изображения используются только два типа ячеек: черные и белые. Поэтому для запоминания каждого пикселя требуется только 1 бит памяти компьютера. Такие изображения часто называются 1-битовыми изображениями. Соответственно, их цветовая разрешающая способность будет равна 1 бит/пиксель.

В другом типе растровых изображений, называемом оттенки серого (Grayscale), на каждый пиксель выделяется до 8 бит информации. Это позволяет оперировать с комбинацией из 256 градаций яркости, перекрывающей весь диапазон оттенков серого от черного до белого.

Если разрешение цифрового изображения, полученного посредством матрицы сканера (или цифровой фотокамеры) совпадает с числом светочувствительных элементов сканера (или камеры), то говорят о фактическом (аппаратном или физическом) разрешении. Однако как в аппаратуре оцифровки изображений, так и в графических программах предусмотрена возможность использования операции интерполяции (Resampling), то есть изменения разрешения, которая может быть реализована разными способами. Например, в программе Adobe Photoshop (рис. 4) реализованы три способа интерполяции - по соседним, билинейная и бикубическая.

При интерполяции по соседним (Nearest Neighbor) для добавляемого программой пикселя берется значение пикселя соседнего с ним. То есть, если соседний пиксель красный, то и программа увеличивает разрешение изображения добавлением красного пикселя.

Термины "маска" и "выделение" взаимосвязаны: выделенная область доступна для редактирования, а область изображения закрытая маской от редактирования защищена.

Маски - это один из базовых инструментов профессиональных растровых редакторов. Всякая маска включает в себя два типа областей: непрозрачные и прозрачные. Первые (непрозрачные) используются для защиты закрываемых ими частей изображений или объектов от нежелательных изменений. Они, собственно, и выполняют функцию маскирования. Прозрачные области можно рассматривать как отверстия в маске. Их используют для выделения фрагментов изображения или объекта, которые собираются модифицировать.

Маска является изображением, которое создается в модели Grayscale (Градации серого) и помещается поверх другого изображения, защищая его от изменений. Для любого пикселя маски значение оттенка серого цвета можно изменять в пределах 256 градаций серого (от 0 до 255). Область маски со значением цвета пикселей, равного 0 (черный), полностью защищает изображение от изменений (она и служит маской). Область, пиксели которой имеют значение 255 (белый), полностью открыта для проведения изменений. Остальная (серая) часть маски частично прозрачна.

В растровом редакторе есть два типа каналов: цветовые и альфа-каналы. Количество цветовых каналов определяется количеством базовых цветов в используемой цветовой модели. Так, изображение в формате Grayscale имеет один канал, в цветовой модели RGB - три канала, а в модели CMYK - четыре канала. В Photoshop доступ к ним реализуется с помощью палитры Channels (Каналы), для отображения которой необходимо выбрать команду Window Channels.

Наряду с цветовыми каналами в растровых редакторах возможно использование альфа-каналов, тесно связанных с понятием маски - каждый такой канал представляет собой маску. Говоря иными словами, создание маски приводит к одновременному и автоматическому созданию альфа-канала, в который помещается изображение этой маски. Для работы с альфа-каналами предусмотрен ряд инструментальных средств, доступ к которым осуществляется с помощью набора кнопок, размещенных в нижней части палитры Channels.

Фильтры представляют собой небольшие программы, которые (подобно макросам и скриптам) выполняют заранее установленную последовательность команд. При этом они автоматически вычисляют значения и характеристики каждого пикселя изображения и затем модифицируют их в соответствии с новыми значениями. Большинство фильтров (filters) предназначено для имитации реальных эффектов. Например, группа художественных эффектов позволяет имитировать самые разнообразные виды живописи (масло, акварель и т. п.) и стили разных художников. Большинство современных графических программ поддерживает подключаемые фильтры других фирм (plug-ins). Их использование расширяет функциональные возможности графического редактора. Вероятно, что слово "альфа канал" все же пока вам не понятно и пугает вас.

Программное обеспечение Adobe Photoshop представляет собой отраслевой стандарт в области графического дизайна. Содержит самые современные наработки и средства для редактирования растровых изображений и создания различных композиций в области matte painting и не только.

Имеет широкий спектр совместимости с другими программами компании Adobe, таких как Adobe After Effects, Adobe Premier, Adobe Illustrator.

Предоставляет пользователю на выбор большой выбор инструментов, начиная от самых простых (карандаш, ластик) заканчивая сложными инструментами с различными настройками (кисть, градиентная заливка и т.д.)

Имеет поддержку цветовой модели изображения, а также включает в себя удобный интерфейс и систему слоёв, что в свою очередь упрощает работу пользователя при создании того или иного эффекта или изображения.

Все инструменты Photoshop условно можно разделить на четыре группы:

- инструменты рисования и закрасивания;

- инструменты для выделения и перемещения выделенных областей;
- инструменты редактирования и ретуши изображений;
- текстовые инструменты.

Предварительный прототип сайта был произведен при помощи программы графического редактора Adobe Photoshop.

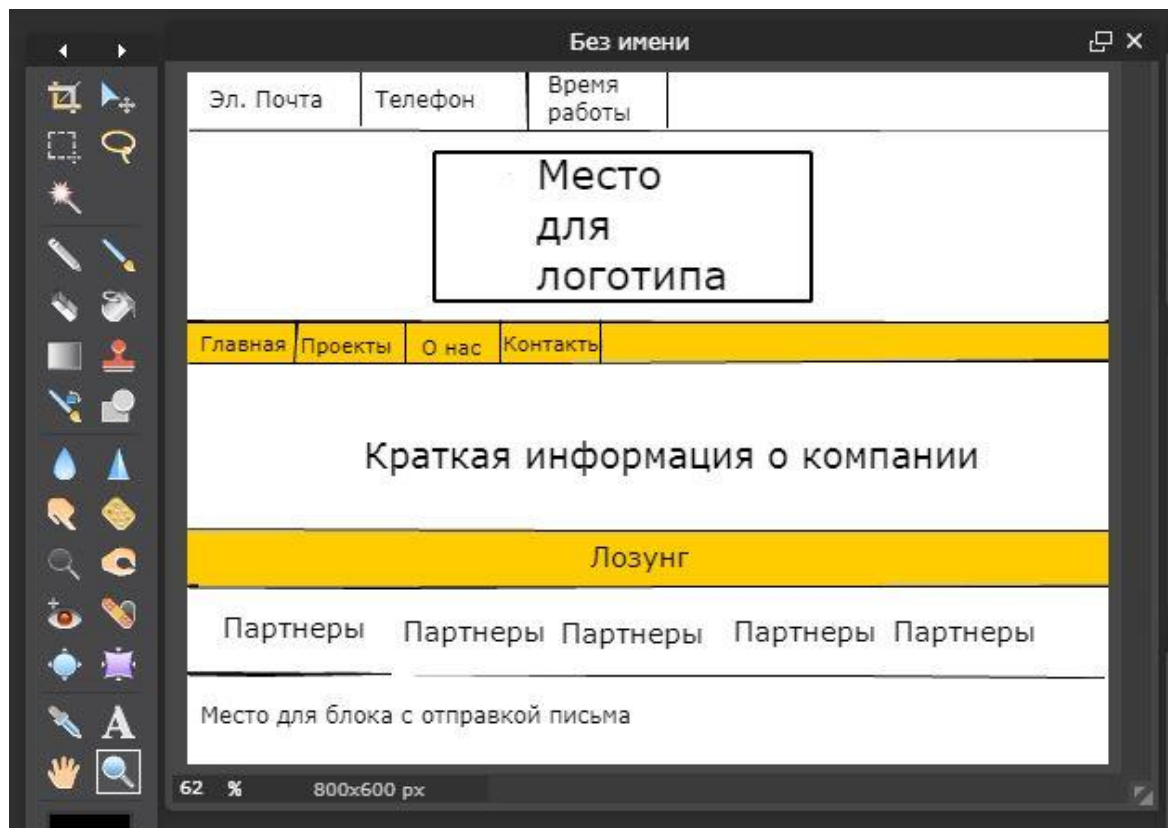


Рисунок 4 – Предварительный макет дизайна сайта

Для удобного использования должен быть реализован крайне понятный интерфейс, распределенный по блокам, каждый из которых содержит необходимые данные или изображение. Также необходимо разработать логотип компании, так как это одна из важнейших составляющих, для формирования ассоциативного поведения.

#### **2.4 Разработка логотипа для предприятия «КазЭнергоПромТорг»**

Логотип для сайта – это эмблема, уникальное изображение, по которому сайт отличают от других.

Фактически, это картинка, с которой будут ассоциировать вашу компанию (это в идеале, если все сделано правильно). Например, у Nike логотип «правильный», потому что узнаваемый.

Вебсайты разрабатывают, как представительство бизнеса в интернете, поэтому логотип на сайте – это логотип организации, часть ее фирменного стиля.

У этого знака есть несколько брендинговых и маркетинговых функций:

- Узнаваемость, отстройка на фоне конкурентов.
- Защита торговых знаков и марок от подделки, копирования. Важно зарегистрировать и оформить все необходимые патенты на всех рынках, на которых планируете работать. Если этого не сделать, может обойтись очень дорого.

- Реклама, формирование лояльности и доверия к бренду. Качественный логотип отработает любые средства, вложенные в его создание.

Технические требования к логотипам на сайтах:

- Размещается вверху страниц, чаще всего в хэдере (шапке).
- Лучше использовать картинку с прозрачным фоном (допустимо с любым). Оптимально выбирать svg – векторный формат. Поскольку растровый (популярный jpg) при масштабировании теряет качество отображения.

- Если никак нельзя избежать размещения слогана вместе с логотипом, оформляем его текстом либо в векторе. Буквы в растре нечеткие, а при малейшем изменении размера станут нечитабельными.

- Само изображение нужно положить в корневую папку на сервере, название должно быть logo, чтобы было легко искать.

- На внутренних страницах сайта используют как элемент навигации. По умолчанию принято и интуитивно используется многими интернет-пользователями, при клике возвращает на титульную страницу.

У каждой компании есть логотип, который в свою очередь отличается своей уникальностью и посвящен деятельности компании. Логотип нужен для того, чтобы человек, зайдя на страницу и увидев его, оставил в памяти ассоциации с компанией. То есть, при появлении мысли о компании, ассоциативно должно вырисовываться изображение логотипа.

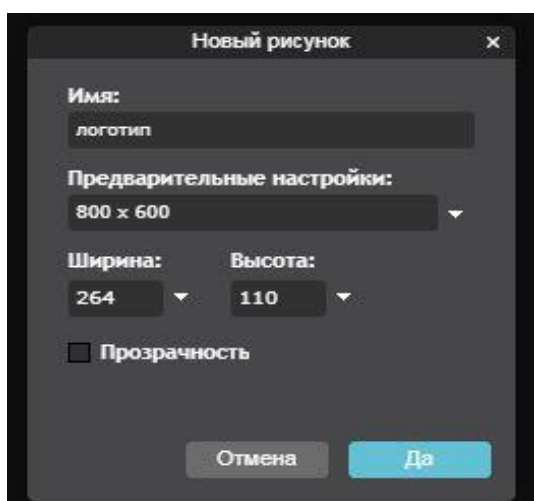


Рисунок 5 – Параметры создания логотипа

На рисунке 5 были произведены настройки параметра создания логотипа. По расчёту логотипа из предварительного макета разработки дизайна сайта были определены ширина и высота логотипа.

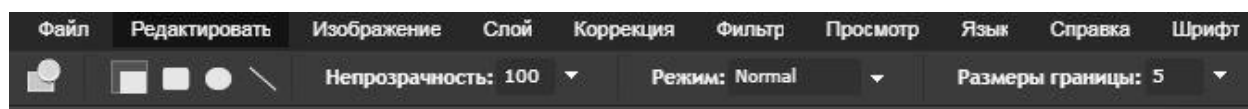


Рисунок 6 – Параметры границ логотипа

Далее на рисунке 6 представлены параметры границ логотипа. Выбираем фигуру прямоугольник для создания квадрата, режим normal, уровень непрозрачности 100 и размеры границы равные 5.

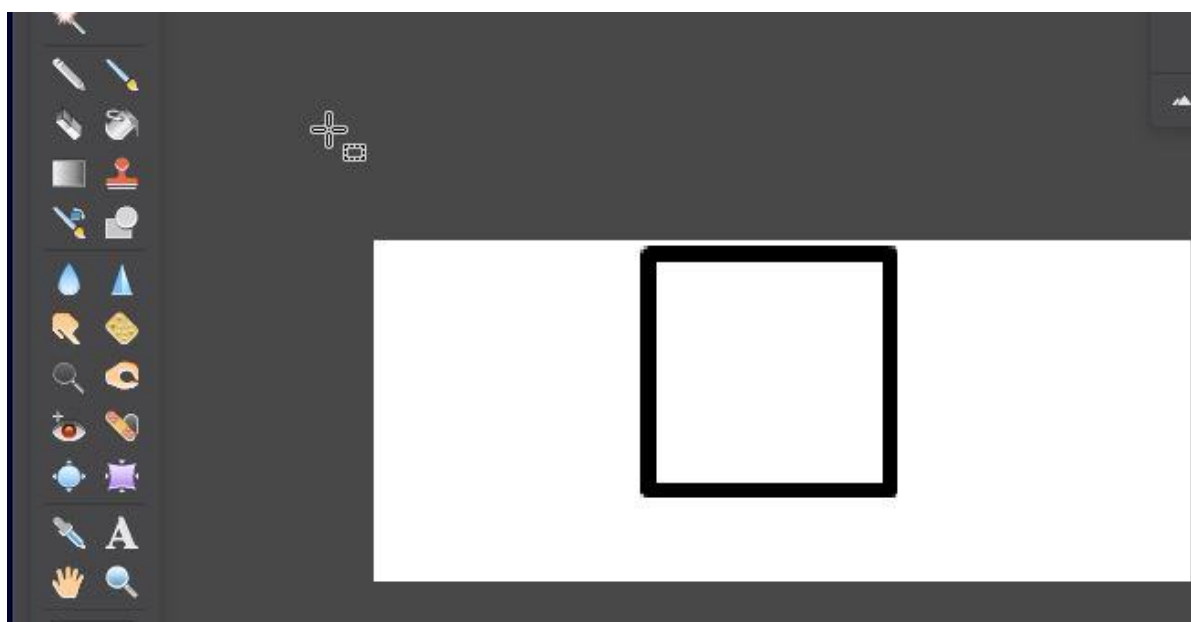


Рисунок 7 – Создание квадрата для логотипа

На рисунке 7 был произведен квадрат для логотипа по вышеуказанным параметрам.

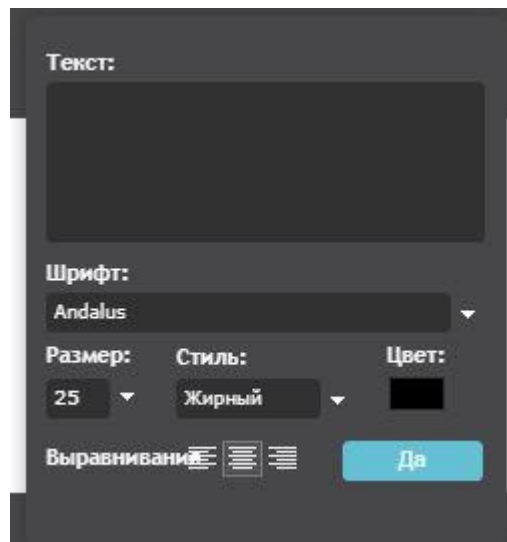


Рисунок 8 – Параметры стилей текста для логотипа

Для создания текста с названием компании настраиваем стили текста. Выбираем шрифт Andalus, стиль текста жирный, размер 25 и выравнивание по центру.



Рисунок 9 – Готовый логотип

Исходя из рисунков логотип для компании был поэтапно создан. На мой взгляд, особенность и уникальность логотипа должен состоять в его простоте. Именно по этой причине я создала логотип, который показан на рисунке 9.

## 3 Разработка программного продукта

### 3.2 Разработка веб – сайта на основе HTML

Тема состоит из папки с набором файлов из html. Каждый файл будет соответствовать отдельной части темы. Не зависимо от этого на какой странице вы находитесь, показываемая вам страница сайта формируется файлами index.html, project.html, aboutus.html, contact.html.

Разберем элементы кодов и их значения с соответствующими изображениями.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>КазЭнергоПромТорг</title>
  <meta name="description" content="КазЭнергоПромТорг">
  <meta name="author" content="Дениза">
  <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
  <link rel="stylesheet" href="css/zerogrid.css">
  <link rel="stylesheet" href="css/style.css">
  <link rel="stylesheet" href="css/responsiveslides.css">
  <link href="font-awesome/css/font-awesome.min.css" rel="stylesheet" type="text/css">
  <link href="owl-carousel/owl.carousel.css" rel="stylesheet">
  <link rel="stylesheet" href="css/menu.css">
  <script src="js/jquery183.min.js"></script>
  <script src="js/script.js"></script>
  <script src="js/jquery-latest.min.js"></script>
  <script src="js/responsiveslides.min.js"></script>
  <script>
    $(function () {
      $("#slider").responsiveSlides({
        auto: true,
        pager: false,
        nav: true,
        speed: 500,
        namespace: "callbacks",
        before: function () {
          $('#events').append("<li>before event fired.</li>");
        },
        after: function () {
          $('#events').append("<li>after event fired.</li>");
        }
      });
    });
  </script>
</head>
```

Рисунок 10 – Начало программы с содержанием стилей CSS и JavaScript

На рисунке 10 представлен скрин с кодом начала программы с содержанием стилей CSS и JavaScript. В этой части задается название программы, страницы, описываются стили CSS с указанием источников из отдельной папки с названием CSS, и используется язык JavaScript также с указанием источников из папки js. (Функции CSS и JavaScript описываются в разделе 1.3. )



```

<body class="home-page">
  <div class="wrap-body">
    <header>
      <div class="top-bar">
        <div class="wrap-top zerogrid">
          <div class="row">
            <div class="col-2-3">
              <ul class="list-inline">
                <li class="mail"><span><i class="fa fa-envelope"></i> KazEnergPromTorg@mail.ru</span></li>
                <li class="phone"><span><i class="fa fa-phone"></i>+7 (727) 3485334</span></li>
                <li class="clock"><span><i class="fa fa-clock-o"></i>Пон-Пт | 8:00 - 20:00</span></li>
              </ul>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

Рисунок 11 – Создание контактных данных в шапке сайта



Рисунок 12 – Результат создания контактных данных в шапке сайта

На рисунке 11 показано создание тела программы, а именно создание контактных данных в шапке сайта.

При помощи тега `<li>` создаем маркированный список с использованием тега `<span>` для для строчных элементов, который можно использовать для форматирования отрывков текста, например, выделения цветом отдельных слов.

```

<div class="header-logo">
  <a href="index.html"></a>
</div>

```

Рисунок 13 – Вставка логотипа в сайт



Рисунок 14 – Результат вставки логотипа в сайт

На рисунке 13 показан метод вставки логотипа в сайт, путем указания источника `<a href="index.html"></a>`.

```

<div id='cssmenu' >
  <ul>
    <li class="active"><a href='index.html'><span>Главная</span></a></li>
    <li><a href='project.html'><span>Проекты</span></a></li>
    <li><a href='aboutus.html'><span>О нас</span></a></li>
    <li class='last'><a href='contact.html'><span>Контакты</span></a></li>
  </ul>
</div>

```

Рисунок 15 – Описание кода для создания навигационного меню



Рисунок 16 – Результат создания навигационного меню

На рисунке 15 представлено описание кода для создания навигационного меню. При помощи тега `<li>` создаем маркированный список с использованием тега `<span>` для строчных элементов, который можно использовать для форматирования отрывков текста, например, выделения цветом отдельных слов и с использованием тега `<a>` для создания гипертекстовых ссылок.

```

<div class="slider">
  <div class="callbacks container">
    <ul class="rslides" id="slider">
      <li>
        
        <div class="caption">
          <h1>СТРОЙФИРМА "КАЗЭНЕРГОПРОМТОРГ"</h1>
          <span>
            Основная деятельность компании: Строительство, реконструкция и ремонт;
            благоустройство территории; общестроительные работы.
          </span>
        </div>
      </li>
    </ul>
  </div>
</div>

```

Рисунок 17 – Описание кода для создания центрального основного отдела



Рисунок 18 – Результат центрального основного отдела

На рисунке 17 этап по созданию центрального основного отдела с использованием изображения в качестве фона. В строке `` указывается адрес источника изображения. Для создания заголовка с крупным шрифтом используется тег `<h1>`. Также используем тег `<span>` для строчных элементов, который можно использовать для форматирования отрывков текста, например, выделения цветом отдельных слов.

```

<section id="container">
  <div class="wrap-container">
    <section class="content-box box-1">
      <div class="zerogrid">
        <div class="row wrap-box">
          <div class="col-1-3">
            <div class="wrap-col">
              <div class="box-item">
                
                <h2>Планирование строительства</h2>
              </div>
            </div>
          </div>
          <div class="col-1-3">
            <div class="wrap-col">
              <div class="box-item">
                
                <h2>Архитектурное моделирование</h2>
              </div>
            </div>
          </div>
          <div class="col-1-3">
            <div class="wrap-col">
              <div class="box-item">
                
                <h2>Управление строительством</h2>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</section>

```

Рисунок 19 – Описание кода для создания отдела для дополнительных функций компании



Планирование строительства



Архитектурное моделирование



Управление строительством

Рисунок 20 – Результат создания отдела для дополнительных функций компании

На рисунке 19 показан код, которым описывается создание отдела для дополнительных функций компании. В строках ``, ``, `` указываются источники изображений для заголовков, которые создаются при помощи тегов `<h2>`, при этом текст получается меньше, чем текст с использованием тега `<h1>`.

```

<section id="container">
<div class="wrap-container zerogrid">
<div class="crumbs">
<center>
<h3>Каталог товаров</h3>
</center>
</div>
</div>
<center>
<table cellspacing="2" border="1" cellpadding="5" width="600">
<tr>
<td align="center"> N </td>
<td align="center"> Наименование товара </td>
<td align="center"> Цена, тр </td>
</tr>
<tr>
<td align="center"> 1 </td>
<td align="center"> Труба DN 57мм. стальная </td>
<td align="center"> 1395 </td>
</tr>
<tr>
<td align="center"> 2 </td>
<td align="center"> Заглушка d 57 электросварная </td>
<td align="center"> 400 </td>
</tr>
<tr>
<td align="center"> 3 </td>
<td align="center"> Заглушка d 76 электросварная </td>
<td align="center"> 600 </td>
</tr>
<tr>
<td align="center"> 4 </td>
<td align="center"> Отвод кованный d25 </td>
<td align="center"> 155 </td>
</tr>
</tr>

```

Рис. 21 – Описание кода для создания таблицы с каталогом товаров

На рисунке 21 представлен код для создания таблиц с каталогом товаров. Таблица создается с использованием тега `<table>`, на нее задаются параметры `cellspacing="2" border="1" cellpadding="5" width="600"`. Далее для создания ячейки таблицы используем тег `<td>`, а для того, чтобы выравнивание таблицы было по центру используем этот тег с использованием тега `<td align="center">`. Для создания новой строки таблицы используется тег `<tr>`.

## Каталог товаров

N	Наименование товара	Цена, тг
1	Труба DN 57мм. стальная	1395
2	Заглушка d 57 электросварная	400
3	Заглушка d 76 электросварная	600
4	Отвод кованный d25	155
5	Отвод кованный d32	170
6	Резьба d25	85
7	Резьба d32	125
8	Сгон обутый d25	380
9	Сгон обутый d32	545
10	Вен. шар. 15"WARMA"	785
11	Вен. шар. 20"WARMA"	1150
12	Вен. шар. 25"WARMA"	1915
13	Вен. шар. 32"WARMA"	2970
14	Вен. шар сварной (Навал) 80	36500
15	Фильтр водяной 25мм."WARMA"	1520
16	Фильтр водяной 32мм."WARMA"	2805
17	Автомат сброса воздуха"WARMA"	1190
18	Вен. рр шар 20 мм.	765
19	Переход н.20х1.2"	240
20	Кран латунный	1530

Рисунок 22 – Результат создания таблицы с каталогом товаров

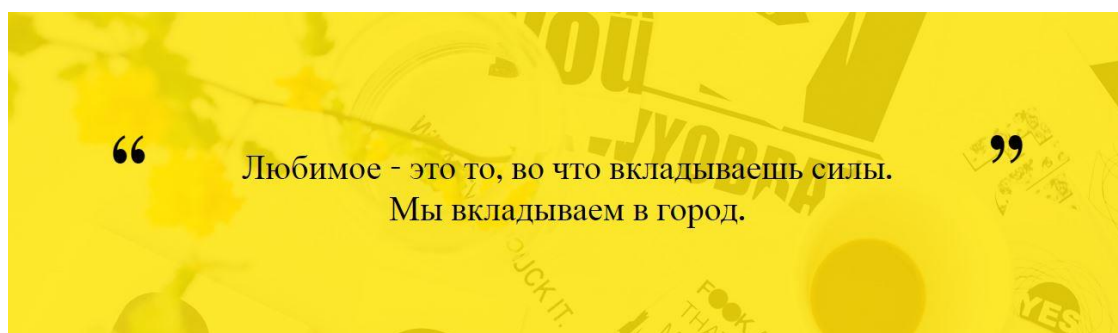
На рисунке 21 представлен код для создания таблиц с каталогом товаров. Таблица создается с использованием тега `<table>`, на нее задаются параметры `cellspacing="2" border="1" cellpadding="5" width="600"`. Далее для создания ячейки таблицы используем тег `<td>`, а для того, чтобы выравнивание таблицы было по центру используем этот тег с использованием тега `<td align="center">`. Для создания новой строки таблицы используется тег `<tr>`.

```

<section class="content-box boxstyle-1 box-2">
  <div class="zerogrid">
    <div class="row wrap-box">
      <div class="box-item">
        <blockquote><p>Любимое - это то, во что вкладываешь силы.<br>Мы вкладываем в город.</p></blockquote>
      </div>
    </div>
  </div>
</section>

```

Рисунок 23 – Описание кода для создания блока с логотипом



## Рисунок 24 – Результат создания блока с логотипом

На рисунке 23 представлено описание кода для создания блока с логотипом. Для этого используется тег `<blockquote>`.

```
<footer>
  <div class="top-footer">
    <div id="owl-brand" class="owl-carousel">
      <div class="item">
        <a href="single.html"></a>
      </div>
      <div class="item">
        <a href="single.html"></a>
      </div>
      <div class="item">
        <a href="single.html"></a>
      </div>
      <div class="item">
        <a href="single.html"></a>
      </div>
      <div class="item">
        <a href="single.html"></a>
      </div>
      <div class="item">
        <a href="single.html"></a>
      </div>
      <div class="item">
        <a href="single.html"></a>
      </div>
    </div>
  </div>
</div>
```

## Рисунок 25 – Описание кода для создания карусели-блока для партнеров компании



## Рисунок 26 – Результат описание кода для создания карусели-блока для партнеров компании

На рисунке 25 представлено описание кода для создания карусели-блока для партнеров компании. Используется тег `<div>` с указанием на источник изображения. Тег `<div>` это тег-контейнер для разделов HTML-документа. Он используется для группировки блочных элементов с целью форматирования.

```

    <div class="col-1-4 col-footer-4">
      <div class="wrap-col">
        <h3 class="widget-title">Напишите нам!</h3>
        <h3 style="color: rgb(255, 255, 255);">Хотите у нас работать?</h3>
        <h4 style="color: rgb(255, 255, 255);">Пожалуйста, отправьте резюме и сопроводительное письмо по адресу: kazenergopromtorg@mail.ru</p>
        <p>Email address:</p>
        <form action="#" method="post">
          <input type="text" name="your-name" value="" size="40" placeholder="Your Email" />
          <input type="submit" value="отправить" class="button button-subscribe" />
        </form>
      </div>
    </div>
  </div>
</div>
</div>
</div>
</div>
</div>
</div>

```

Рисунок 27 – Описание кода для создания дополнительного блока с полем для отправки письма на адрес компании

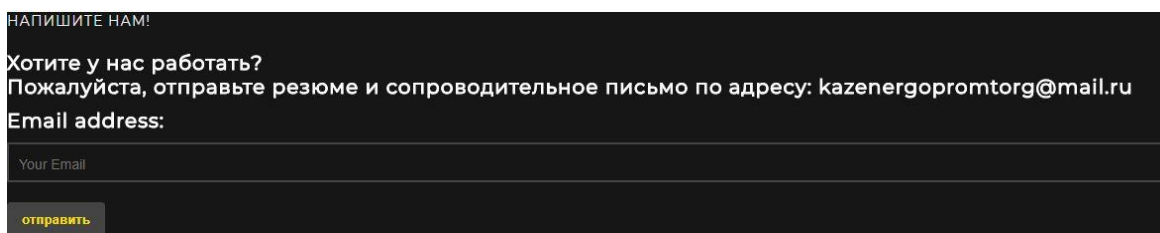


Рисунок 28 – Результат создания дополнительного блока с полем для отправки письма на адрес компании

На рисунке 27 представлено описание кода для создания дополнительного блока с полем для отправки письма на адрес компании. Для строки `<h3 class="widget-title">Напишите нам!</h3>` используем размер шрифта тега `<h3>`, а для строки `<h4 style="color: rgb(255, 255, 255);">Пожалуйста, отправьте резюме и сопроводительное письмо по адресу: kazenergopromtorg@mail.ru</p>` используем тег `<h4>` с указанием цвета шрифта. Для создания поля для ввода используем тег `<input type="text">`, а для создания кнопки используем этот же тег, но с использованием параметра кнопки `<input type="submit">`.

```

    <div id="main-content" class="col-2-3">
      <div class="wrap-content">
        <article>
          <div class="art-header">
            <a href="#"><h3>Микрорайон Нуркент, г. Алматы </h3></a>
            <div class="info">Posted on March 13, 2018 </div>
          </div>
          <div class="art-content">
            
            <p>Расположение Новостройки Алматы, Алатауский р-н <br> ЖК «Нуркент» в г.
            Та часть города, в которой осуществляется строительство, уже сейчас является весьма удобной для жизни, но
            </div>
          </div>
        </div>
      </div>
    </div>

```

Рисунок 29 – Описание кода для файла project.html

## Микрорайон Нуркент, г. Алматы

Posted on March 13, 2018



Расположение Новостройки Алматы, Алатауский р-н

ЖК «Нуркент» в г. Алматы – это несколько 9-этажных жилых домов, которые возводятся в Алатауском районе города в рамках программы «Доступное жилье». Та часть города, в которой осуществляется строительство, уже сейчас является весьма удобной для жизни, но дальнейшие перспективы ее развития позволяют с полной уверенностью утверждать, что жилой комплекс «Нуркент» и квартиры в нем будут пользоваться высоким спросом и

Рисунок 30 – Результат выполнения файла project.html

На рисунке 29 представлен код описания файла project.html для создания страницы с данными о проделанной работе. Для создания данной страницы описывается часть кода из основного файла index.html, а также создается сам блок с информацией о работе и изображением. Для этого необходимо использовать самые простые теги для создания заголовков, вставки текста и изображения - `<h3>`, `<img>`, `<p>`. Используется тег `<div>` с указанием на источник изображения. Тег `<div>` это тег-контейнер для разделов HTML-документа. Он используется для группировки блочных элементов с целью форматирования. Также используем тег `<span>` для строчных элементов, который можно использовать для форматирования отрывков текста, например, выделения цветом отдельных слов.



```

</div>
<div id="sidebar" class="col-1-3">
  <div class="wrap-sidebar">
    <div class="widget wid-gallery">
      <div class="wid-header">
        <h5>Галерея</h5>
      </div>
      <div class="wid-content">
        <a href="#"></a>
        <a href="#"></a>
        <a href="#"></a>
        <a href="#"></a>
        <a href="#"></a>
        <a href="#"></a>
      </div>
    </div>
  </div>
</div>
</div>
</section>

```

Рисунок 31 – Описание кода для создания блока «Галерея»



Рисунок 32 – Результат создания блока «Галерея»

Для создания блока «Галерея» из рисунка 30 необходимо использовать теги `<img>` и `<a>`.

```

<div id="main-content" class="col-2-3">
  <div class="wrap-content">
    <article>
      <div class="art-header">
        <div class="entry-title">
          <h2>Немного о нас</h2>
        </div>
      </div>
      <div class="art-content">
        
        <div class="excerpt">
<p>Стройфирма «КазЭнергоПром Торг» была создана в 2012 году. Основная деятельность организации – это строительство и ремонт
в течение пяти лет были выполнены строительные работы. За очень небольшой промежуток времени строительная фирма "КазЭнерго
в составе предприятия работают специалисты высокой квалификации, отвечающие всем требованиям современных стандартов. Ежегод
<h5>Наши проекты</h5>
</div>
<div class="wid-content">
  <div class="row">
    <div class="col-1-3">
      <div class="wrap-col">
        <a href="#"></a>
        <h4><a href="#">Школа №17</a></h4>
      </div>
    </div>
  </div>

```

Рисунок 33 – Описание кода для создания страницы файла aboutus.html

## Немного о нас



Стройфирма «КазЭнергоПром Торг» была создана в 2012 году. Основная деятельность организации – это строительство и ремонт зданий, сооружений промышленного и гражданского назначения. В течение пяти лет были выполнены строительные работы. За очень небольшой промежуток времени строительная фирма "КазЭнергоПром Торг" зарекомендовала себя в положительном свете на строительном рынке г. Алматы, наработала постоянных клиентов-заказчиков. В составе предприятия работают специалисты высокой квалификации, отвечающие всем требованиям современных стандартов. Ежегодно обновляется техническое оснащение персонала, а также повышается квалификация работающих. Состав предприятия остаётся постоянным на протяжении длительного времени. ТОО "КазЭнергоПром Торг" выступало как единственным подрядчиком, так и выполняла функции генерального подрядчика. В связи с этим имеется возможность осваивать объёмы работ с высокой скоростью, но в

Рисунок 34 – Результат создания страницы файла aboutus.html

На рисунке 33 представлен код для создания блока «О нас» файла aboutus.html. Для этого необходимо использовать самые простые теги для создания заголовков, вставки текста и изображения - `<h3>`, `<img>`, `<p>`.

```

<div id="main-content">
  <div class="row">
    <h1 class="t-center" style="margin: 40px 0;color: #212121;letter-spacing: 2px;font-weight: 500;">Связаться</h1>
    <iframe src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d2904.937418394863!2d76.94723421515171!3d43.273686879136">
  </div>
  <div class="col-1-3">
    <div class="wrap-col">
      <h3 style="margin: 20px 0">Контактная информация</h3>
      <p>Если у вас возникли какие-то вопросы, обращайтесь по контактному адресу:</p>
      <p>Казахстан, г. Алматы<br>
      ул. Суюнбая 17</p>
      <p>kazenergopromtorg@mail.ru</p>
      <p>Телефон: +7 (727) 3485334</p>
      <p>Факс: +7 (727) 3485330 <br></p>
    </div>
  </div>
</div>

```

Рисунок 35 – Описание кода для создания блока «Контакты» файла contact.html

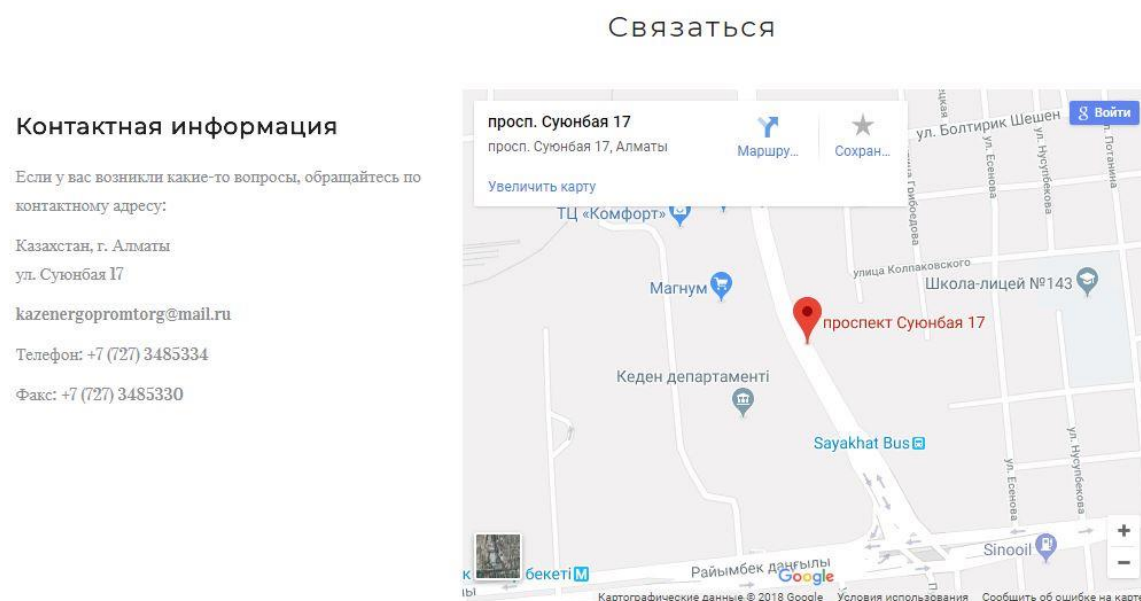


Рисунок 36 – Результат создания блока «Контакты» файла contact.html

На рисунке 35 представлено описание кода для создания блока «Контакты» файла contact.html с использованием google-карты. В строке `<h1 class="t-center" style="margin: 40px 0;color: #212121;letter-spacing: 2px;font-weight: 500;">Связаться</h1>` указываем заголовок с использованием тега для текста с крупным шрифтом `<h1>` с использованием параметров шрифта. Далее вставляем ссылку на google-карту с использованием тега `<iframe>`, который создает встроенный фрейм, загружая в текущий HTML-документ другой документ. Также для дальнейшего указания способов связи используем простые теги `<p>`, `<h3>`, `<br>`. Используется тег `<div>` с указанием на источник изображения. Тег `<div>` это тег-контейнер для разделов HTML-документа. Он используется для группировки блочных элементов с целью форматирования. Также используем тег `<span>` для строчных элементов, который можно использовать для форматирования отрывков текста, например, выделения цветом отдельных слов.



Опция `itemsDesktopSmall` используется для того, чтобы проверить, соответствует ли размеры экрана устройства указанным параметрам. Если ширина экрана меньше чем 979, то карусель будет вмещать в себя только 2 элемента.

Опция `itemsDesktop` активирует возможность пролистывания карусели влево и направо при нажатии на соответствующие навигационные элементы карусели.

Следующая опция, а именно `navigationText` назначает текст для кнопок навигаций.

Опция `pagination` это булева опция, которая отвечает за включения или отключение пагинации.

```

1           <h3>Хотите у нас работать?</h3>
2 <h4>Пожалуйста, отправьте резюме и сопроводительное письмо по адресу: kazenergopromtorg@mail.ru</h4>
3 <p>Email address:</p>
4 <form action="#" method="post">
5     <input type="text" name="your-name" value="" size="40" placeholder="
6       Your Email" />
7     <input type="submit" value="отправить" class="button button-subscribe"
8       />
9 </form>
10 </div>
11 </div>
12 </div>
13 </div>
14 </div>
15 </div>
16 </div>
17 </div>
18 </div>
19
20 <!-- carousel -->
21 <script src="owl-carousel/owl.carousel.js"></script>
22 <script>
23 $(document).ready(function() {
24   $("#owl-brand").owlCarousel({
25     autoPlay: 3000,
26     items : 6,
27     itemsDesktop : [1199,4],
28     itemsDesktopSmall : [979,2],
29     navigation: true,
30     navigationText: ['<i class="fa fa-chevron-left fa-5x"></i>', '<i class="fa fa-chevron-right
31       fa-5x"></i>'],
32     pagination: false
33   });
34 });
35 </script>
```

## Рисунок 38 – Описание JavaScript

Файл `aboutus.js` содержит такой же код, как и в `index.html` для отображения карусели в конце страницы. Функция вызывается только тогда, когда наша страница уже подгрузилась и готова показаться конечному пользователю, то есть для того, кто просматривает данную страницу.



Страница contact.html содержит в себе JavaScript код, который отвечает за подгрузку и отображение Google карты, где показывается местоположение компании.

В коде содержатся такие переменные как marker и image. Вторая переменная нужна, чтобы указать, где именно находится фотография нашей метки, а первая - непосредственно сам маркер на Google карте.

Функция initMap() - функция, где указываются все начальные параметры отображения карты, включая стили карты, цвет, насыщенность и т.д. Тут же настраивается и отображение лейблов.

```
        styles: [
            { visibility: "off" }
        ]
    }
};

var map = new google.maps.Map(document.getElementById('map'), {
    center: myLatLng,
    scrollwheel: false,
    // Apply the map style array to the map.
    styles: styleArray,
    zoom: 7
});

var directionsDisplay = new google.maps.DirectionsRenderer({
    map: map
});

// Create a marker and set its position.
marker = new google.maps.Marker({
    map: map,
    icon: image,
    draggable: true,
    animation: google.maps.Animation.DROP,
    position: myLatLng
});
marker.addListener('click', toggleBounce);
}
```

### Рисунок 41 – Описание JavaScript

Дальше в коде инициализируется сама карта с заданными прежде параметрами. Здесь указывается и координаты (широта и долгота), по которым центрируется карта, то есть куда будет указывать карта изначально.

Параметр zoom указывается для приближения отображения карты до нужных размеров.

Дальше мы передаем переменную map в класс DirectionsRenderer чтобы он нарисовал карту на странице.

Следом за этим мы создаем маркер (метку), которая будет указывать на точку, где находится компания. В параметр map мы указываем уже выше созданную переменную карт, в параметре icon указываем ссылку на картинку, которая будет показана для маркера.

Параметр draggable позволяет делать карту перемещаемой, либо обратно.

Параметр animation указывает на тип анимации при работе с картой.

Параметр position - непосредственно координаты описываемого маркера (метки).

Следующая строка кода добавляет слушатель (действие) на нажатие поверх созданной метки. В функцию передается другая функция, где описано, что должно происходить тогда, когда происходит событие (в данном случае - нажатие на маркер).

```
// Create a marker and set its position.
marker = new google.maps.Marker({
  map: map,
  icon: image,
  draggable: true,
  animation: google.maps.Animation.DROP,
  position: myLatLng
});
marker.addListener('click', toggleBounce);
}

function toggleBounce() {
  if (marker.getAnimation() !== null) {
    marker.setAnimation(null);
  } else {
    marker.setAnimation(google.maps.Animation.BOUNCE);
  }
}
</script>
<script src="https://maps.googleapis.com/maps/api/
js?key=AIzaSyB7V-mAjEzzmP6PCQda8To0ZW_o3UOCVCE&callback=initMap" async defer></script>

</body>
</html>
```

Рисунок 42 – Описание JavaScript

Как уже упоминалось выше, в функцию добавления слушателя на событие передается другая функция, которая выступает в роли callback-функции, где описан код, который будет выполняться при нажатии на вышеупомянутый маркер (метку).

В случае с этим кодом, при нажатии на какую-либо метку на карте происходит смена основной метки на позицию нажатой метки. Во время выполнения этой операции происходит анимации, которая описывается в функции toggleBounce().

Следует заметить, что Google карты подгружаются только в конце загрузки страницы, чтобы не увеличивать время ожидания конечного пользователя. Загрузка и отображение карты происходит асинхронно, то есть выполняется в отдельном потоке процесса.

### 3.2.2 CSS

Каскадные стили CSS необходимо использовать для описания внешнего вида документа. Разберем элементы кодов и их значения с соответствующими изображениями.



```

<link rel="stylesheet" href="css/zerogrid.css">
<link rel="stylesheet" href="css/style.css">
<link rel="stylesheet" href="css/responsiveslides.css">

```

Рисунок 43 – Связка CSS с HTML

На рисунке 43 представлено связки файлов CSS с HTML для дальнейшей работы с CSS для создания внешнего вида веб-сайта.

```

* {
  -webkit-box-sizing: border-box;
  -moz-box-sizing: border-box;
  box-sizing: border-box;
}
::-moz-selection{background: #FFE401 none repeat scroll 0 0;color: #000;text-shadow:none}
::selection{background: #FFE401 none repeat scroll 0 0;color: #000;text-shadow:none}

html { -webkit-text-size-adjust: none;}
.video embed,.video object,.video iframe { width: 100%; height: auto;}
img{max-width:100%; height: auto;width: auto\9; /* ie8 */}

a{color: #000;text-decoration: none;-webkit-transition: 0.3s;
  -moz-transition: 0.3s;
  -ms-transition: 0.3s;
  -o-transition: 0.3s;
  transition-duration: 0.3s;}
a:hover {text-decoration: underline;}

h1, h2 {font-size: 25px;line-height: 30px;}
h3, h4 {font-size: 20px;line-height: 20px;}
h5, h6 {font-size: 16px;line-height: 16px;}

h1, h2, h3, h4, h5, h6{clear: both;hyphens: auto; color: #222;line-height: 1.2;font-family: 'Montserrat'}

p{margin: 10px 0;}

.t-center{text-align: center;}
.t-left{text-align: left;}
.t-right{text-align: right;}
.f-right{float: right;}
.f-left{float: left;}

```

Рисунок 44 – Описание стилей

На рисунке 44 представлены стили для создания блоков. Согласно спецификации CSS ширина блока складывается из ширины контента width, значений отступов margin, полей padding и границ border. Аналогично обстоит и с высотой блока. Свойство box-sizing позволяет изменить этот алгоритм, чтобы свойства width и height задавали размеры не контента, а размеры блока. Строка box-sizing: border-box означает, что свойства width и height включают в себя значения полей и границ, но не отступов margin. Эта модель используется браузером Internet Explorer в режиме несовместимости.

В строке -moz-selection{background: #FFE401 none repeat scroll 0 0;color: #000;text-shadow:none} описывается определенный выделенный блок, где элемент -moz-selection задает фону блока цвет #FFE401(желтый) и отменяет добавление тени.

Опция a{color: #000;text-decoration: none;-webkit-transition: 0.3s; задает цвет #000 и отменяет все эффекты, в том числе и подчеркивания у ссылок,

которое задано по умолчанию. transition-duration: 0.3s;} задает время в 0,3 секунды для того, чтобы установить длительность анимации перехода до её завершения.

Опция h1, h2, h3, h4, h5, h6{clear: both;hyphens: auto; color: #222;line-height: 1.2;font-family: 'Montserrat', sans-serif;letter-spacing: 1px;} задает параметры стилей текста, а именно межстрочный интервал 1,2, семейство шрифта 'Montserrat', черный цвет текста, снимает обтекание элемента, добавляет автоматический перенос слов.

```
.post{}
.post:after, .post:before, article:after, article:before, section:after, section:before{clear: both; content: '\0020'; }
.clear{content: "\0020"; display: block; height: 0; clear: both; visibility: hidden; }
.clearfix:after, .clearfix:before{clear: both; content: '\0020'; display: block; visibility: hidden; width: 0; height: 0; }
/* --Line-- */
hr.line { margin: 15px auto; padding: 0; max-width: 80px;border: 0; border-top: solid 6px #ccc;text-align: center;border-radius: 6px;}
.uppercase{text-transform: uppercase;}
.widget-title{font-weight: 500;text-transform: uppercase;line-height: 1.38;margin-bottom: 20px;font-size: 14px;color: #ccc;}
ul.list-inline li{display: inline-block;}

html, body {width:100%; padding:0; margin:0;}
body {font: 15px/25px Arial;font-family: Vidaloka,serif;background:#fff;color: #8a8c8e;}
body .wrap-body{}

@media all and (max-width: 800px), only screen and (-webkit-min-device-pixel-ratio: 2) and (max-width: 1024px), only screen and (min-width: 1024px)
body{padding: 0;}
}

header{}
header .top-bar {background-color: #fff;border: 1px solid #ccc;}
header .top-bar .wrap-top{}

header .top-bar ul li {border-right: 1px solid #ddd;padding: 13px 25px;}
header .top-bar ul li:first-child {border-left: 1px solid #ddd;}
header .top-bar .mail span, header .top-bar .phone span{color:#999;font-size:14px;}
header .top-bar i.fa{margin: 3px 6px 0 0;font-size: 17px;}

header .top-bar .top-social {float: right;text-align: center;}
header .top-bar ul.top-social li{border: none;padding: 5px;}
header .top-bar .top-social a{display: block;width: 38px;height: 38px;border-radius: 50%;font-size: 20px;line-height: 38px;}
header .top-bar .top-social a:hover{border: 1px solid #000;color: #FFE401;background: #000;}
header .top-bar .top-social i.fa{margin: 0;font-size: 19px;}
```

## Рисунок 45 – Описание стилей

На рисунке 45 представлены стили описания блоков и текстов.

В строке hr.line { margin: 15px auto; padding: 0; max-width: 80px;border: 0; border-top: solid 6px #ccc;text-align: center;border-radius: 6px;}uppercase{text-transform: uppercase;} описывается отступ в 15 пикселей, нулевой отступ полей вокруг содержимого элемента, ширину элемента в 80 пикселей, отсутствие толщины элемента, центральная ориентация текста, радиус границ 6 пикселей.

Правило @media позволяет указать тип носителя, для которого будет применяться указанный стиль, в данном случае выбираются все типы носителя, для которых описываются различные параметры.

В строке header-logo{text-align: center;padding: 50px 0; min-height: 100px;background: #fff;} описывается положение логотипа по центру, отступ полей в 50 пикселей, белый цвет фона.

```

.content-box {}
.content-box .wrap-box {padding: 70px 60px; position: relative;}
.content-box .header { margin: 0 0 60px;text-align: center;}
.content-box .header h2{font-size: 30px;margin: 0;padding: 0;line-height: .9;text-transform: upperca}
.content-box .header .wrapper{padding: 5px; }
.content-box .header .intro {font-size: 18px;}

@media only screen and (max-width: 767px) {
  .content-box .wrap-box {padding: 40px 20px;}
}

.content-box.boxstyle-1{background: url(../images/background1.jpg) no-repeat fixed 0 0;}
.content-box.boxstyle-2{background: #f7f7f7;}
.content-box.boxstyle-3{background: url(../images/action-bg.jpg) no-repeat fixed;}

.content-box.box-1 h2{font-size: 16px;margin-top: 10px;}
.content-box.box-1 .box-item{}

.content-box.box-2 .box-item{color: #000;padding: 60px 0;}
.content-box.box-2 blockquote {position: relative;padding: 20px 120px 0;text-align: center;font-size}
.content-box.box-2 blockquote:before, .content-box.box-2 blockquote:after {position: absolute;width:}
.content-box.box-2 blockquote:before {top: 0;left: 0;content: "\201C"; }
.content-box.box-2 blockquote:after {top: 0;right: 0;content: "\201D"; }
@media only screen and (max-width: 1100px) {
  .content-box.box-2 blockquote {font-size: 30px; }
}
@media only screen and (max-width: 960px) {
  .content-box.box-2 blockquote {padding: 30px 60px;font-size: 22px; }
  .content-box.box-2 blockquote:before, .content-box.box-2 blockquote:after {width: 30px;height: 30px}
}
@media only screen and (max-width: 690px) {
  .content-box.box-2 blockquote {padding: 10px 30px;font-size: 20px; }
  .content-box.box-2 blockquote:before, .content-box.box-2 blockquote:after {font-size: 35px; }
}

```

## Рисунок 46 – Описание стилей

Опция `content-box` применяется для изменения алгоритма расчета ширины и высоты элемента. Согласно спецификации CSS ширина блока складывается из ширины контента `width`, значений отступов `margin`, полей `padding` и границ `border`. Аналогично обстоит и с высотой блока. В данном случае описывается отступ полей в 70 пикселей, параметры текста, положение и способ позиционирования элемента относительно его исходного места. Добавление свойств `left`, `top`, `right` и `bottom` изменяет позицию элемента и сдвигает его в ту или иную сторону от первоначального расположения.

Опция `background` задает цвет заднего фона, либо источник изображения, который будет служить в качестве фона.

Тег `<blockquote>` предназначен для выделения длинных цитат внутри документа. Текст, обозначенный этим тегом, традиционно отображается как выровненный блок с отступами слева и справа (примерно по 40 пикселей), а также с отбивкой сверху и снизу.

Согласно спецификации CSS ширина блока складывается из ширины контента `width`, значений отступов `margin`, полей `padding` и границ `border`. Аналогично обстоит и с высотой блока. Свойство `box-sizing` позволяет изменить этот алгоритм, чтобы свойства `width` и `height` задавали размеры не контента, а размеры блока.

```

#main-content{margin-top: 40px;}
#main-content .wrap-content{padding: 0 30px 20px;border-right: 1px solid #ddd;}
#main-content .wrap-content h3{margin-bottom: 10px;}
#main-content .wrap-map{background: #fff;padding: 5px;border: 1px solid #bbb;max-width: 520px;}

article{display: inline-block;border-bottom: 1px solid #ddd;padding-bottom: 30px;margin-bottom: 30px;}
article:last-child{border-bottom: none;padding-bottom: 0;}
article:after, article:before{clear: both; content: '\0020'; display: block; visibility: hidden; width: 0; height: 0;}
article .art-header h3{font-size: 25px;}

article .art-content a{ color: #FE9B13;font-weight: bold;}
article .art-content a:hover{ color: #000;}
article .art-content blockquote {border-left: 3px solid #1c7791;font-size: 18px;margin: 0 50px;font-style: italic}
article .art-content ol {float: left;text-align: justify;margin: 10px 50px;font-size: 16px;list-style-type: circle}

article .entry-title {margin: 10px 0;}
article .entry-title h2{line-height: 1.2;font-size: 38px;}
article .info a{}

article img{display: block;width: 100%;margin-top: 10px;}

/* ---Sidebar-Widget--- */
#sidebar{margin-top: 40px;}
#sidebar .wrap-sidebar{margin: 0px 10px ;}

.widget {margin-bottom: 10px;padding: 5px 25px 10px;}
.widget ul li { line-height: 2.8;border-bottom: 2px solid #DDD;}
.widget:after, .widget:before{clear: both; content: '\0020'; display: block; visibility: hidden; width: 0; height: 0;}

.widget .wid-header {}
.widget .wid-header h5 {display: inline-block;padding: 20px 10px 20px 0;margin-bottom:-4px;font-size: 23px;}
.widget .wid-content{padding: 0 0 20px;border-bottom: 1px solid #ddd;}

.widget.wid-about {}

```

## Рисунок 47 – Описание стилей

Опция `margin-top` устанавливает величину отступа от верхнего края элемента. Отступом является расстояние от внешнего края верхней границы текущего элемента до внутренней границы его родительского элемента. В данном случае величина отступа в 40 пикселей.

Многоцелевое свойство `display` определяет, как элемент должен быть показан в документе. Это значение генерирует блочный элемент, который обтекает другими элементами веб-страницы подобно встроенному элементу `.br`. Фактически такой элемент по своему действию похож на встраиваемые элементы (вроде тега `<img>`). При этом его внутренняя часть форматируется как блочный элемент, а сам элемент – как встроенный.

Также, в данном коде описывается свойство `border-bottom`, которое позволяет одновременно установить толщину, стиль и цвет границы внизу элемента размеров в один пиксель. Далее опции `padding-bottom` и `margin-bottom` устанавливают значения поля и величину отступа от нижнего края содержимого элемента размером в 30 пикселей.

Опция `visibility: hidden` делает элемент невидимым или правильней сказать, полностью прозрачным, поскольку он продолжает участвовать в форматировании страницы.

```

#sidebar{margin-top: 40px;}
#sidebar .wrap-sidebar{margin: 0px 10px ;}

.widget {margin-bottom: 10px;padding: 5px 25px 10px;}
.widget ul li { line-height: 2.8;border-bottom: 2px solid #DDD;}
.widget:after, .widget:before{clear: both; content: '\0020'; display: block; visibility: hi

.widget .wid-header {}
.widget .wid-header h5 {display: inline-block;padding: 20px 10px 20px 0;margin-bottom:-4px;
.widget .wid-content{padding: 0 0 20px;border-bottom: 1px solid #ddd;}

.widget.wid-about {}

.widget.wid-tag {}
.widget.wid-tag a{color: #a9a9a9; background: #202020; padding: 5px 10px;line-height: 2.3;
.widget.wid-tag a:hover{color: #fff;background: #FFE401;}

.widget.wid-post {}

.widget.wid-gallery {}
.widget.wid-gallery img{margin: 0 10px 10px 0; max-width: 150px;}
.widget.wid-gallery .wid-content{border-bottom: none;}

.widget.wid-archive {}
.widget.wid-archive select {margin: 5px 0;height: 40px;width: 100%;padding-left: 10px;font-

.widget.wid-meta {}
.widget.wid-meta ul li{ font-size: 16px;}

.widget.wid-related {background-color: #fff;}
.widget.wid-related .wid-content{border-bottom: none;}
.widget.wid-related img{ max-width: 243px; width: 100%;}
.widget.wid-related h4{margin-top: 10px;font-size: 16px;}
.widget.wid-related a{color: #333;}

```

## Рисунок 48 – Описание стилей sidebar

Функцией `margin-top` устанавливается величина отступа от верхнего края элемента в 40 пикселей.

Опцией `background` описывается определенный выделенный блок, где элемент `-moz-selection` задает фону блока цвет `#FFE401`(желтый) и отменяет добавление тени.

Опция `a{color: #000;text-decoration: none;-webkit-transition: 0.3s;` задает цвет `#000` и отменяет все эффекты, в том числе и подчеркивания у ссылок, которое задано по умолчанию. `transition-duration: 0.3s;` задает время в 0,3 секунды для того, чтобы установить длительность анимации перехода до её завершения.

## **4. Техничко-экономическое обоснование**

### **4.1 Описание работы и обоснование необходимости**

Целью данной дипломной работы является создание сайта для стройфирмы «КазЭнергоПромТорг». Основными задачами являются удобное использование сайта, поиск необходимой информации и ознакомление с данной фирмой для дальнейшей работы с ней. В новую эпоху – эпоху четвертой промышленной революции весь мир будет постепенно переходить на различную технику, и моя задача состоит в том, чтобы осуществить малую долю части этого нового мира.

При разработке любого программного продукта необходимо изучить тщательно все его экономические стороны, подсчитать все для экономии и прибыли. Необходимо учесть все факторы: сложность и объем решаемой задачи, стоимость и доступность необходимых ресурсов, необходимый уровень абстракции при разработке и др.

Для начала, необходимо составить список всех этапов и видов работ, точнее говоря – план выполнения работы.

Следует разделить затраты на различные категории: материальные затраты, затраты на оплату труда, амортизация основных фондов, прочие затраты. Прямые материальные затраты являются фактической себестоимостью основных материалов, израсходованных на производство.

В данном разделе приводится рассмотрение экономической составляющей реализации данной проекта, отражающей временные, трудовые финансовые затраты на проект.

### **4.2 Трудоемкость разработки ПП**

Для определения трудоемкости разработки необходимо составить перечень этапов разработки и логически их упорядочить.

Основными задачами планирования работ являются:

- определение объема предстоящих работ;
- взаимная увязка работы и установление рациональной последовательности предстоящих работ;
- установление сроков выполнения работ.

Работы по планированию сводятся к составлению перечня работ, определению их трудоемкости, расчету длительности цикла работ, обоснования сметы затрат на проведение работ.

Перечень работ, разделенный на этапы с указанием трудоемкости приведен в таблице 4.1.

Таблица 4.1 – Распределение работ по этапам и видам и оценка их трудоемкости.

Этапы разработки ПП	Вид работы на данном этапе	Трудоемкость разработки ПП		
		Чел. x час	Кол-во часов в день	Кол-во дней
Анализ требований	Установка цели и задач проекта, выделение базовых сущностей и взаимосвязей между ними	1 x 35	7	5
Проектирование	Получение технических заданий, назначение требований к пользовательскому интерфейсу, оценка и подбор оборудования	1 x 42	7	6
Реализация	Экспериментирование и анализ, строение прототипов, как целой системы, так и ее частей	1 x 42	7	6
Тестирование продукта	Тестирование и дополнение в системе	1 x 35	7	5
Внедрение и поддержка	Установка программного обеспечения, обучение пользователей, исправление выявленных ошибок.	1 x 49	7	7
Итоговая трудоемкость выполнения дипломной работы		1 x 203	7 x 29	

### 4.3 Расчет затрат на разработку ПП

Для определения затрат на разработку ПП нужно составить смету, которая включает следующие статьи:

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

#### 4.4 Материальные затраты

К материальным затратам в данной дипломной работе относятся затраты на основные и вспомогательные материалы, а также затраты на электроэнергию. В таблице 4.2 приведен расчет затрат на материальные ресурсы. В таблице 4.3 произведен расчет затрат на программные обеспечения

Таблица 4.2 – Расчет затрат на материальные ресурсы

№	Наименование	Описание	Цена за единицу, тг	Сумма, т
1	Ноутбук	Asus X541U	200000	200000
2	Оптический сетевой терминал	Tenda E100	17000	17000
	Итого			217000

Таблица 4.3 – Расчет затрат на программные обеспечения

№	Наименование	Название продукта	Сумма
1	Операционная система	Microsoft Windows Basic 7 64 bit	50000
2	Антивирус	ESET NOD32 Cyber Security Pro	12000
3	Adobe Photoshop cs 3	Adobe Photoshop cs 3	14000
4	Denwer	Denwer	Бесплатно
5	Notepad++	Notepad++	Бесплатно
	Итого		76000

Общая сумма затрат на материальные ресурсы ( $Z_m$ ) определяется по формуле (1).

$$Z_m = \sum_{i=1}^n P_i * C_i. \quad (1)$$

#### 4.5 Затраты на электроэнергию

Так как для создания сайта мы используем технику и электричество, также необходимо рассчитать затраты на потребляемое электричество.

Затраты на электроэнергию приведены в Таблице 4.4. Общая сумма затрат рассчитывается по формуле (2).



$$Z_э = \sum_{i=1}^n M_i * K_i * T_i * Ц. \quad (2)$$

С 1 марта 2018 года цена на электроэнергию по тарифу ТОО «АлматыЭнергоСбыт» составляет 21,99 тг/кВт\*ч с учетом НДС.

Таблица 4.4 – 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
Оптический сетевой терминал	0,9	0,9	203	21,99	3615,8
Ноутбук	0,3	0,7	203	21,99	937,4
Итого затраты на электроэнергию					4553,2

#### 4.6 Затраты на оплату труда

Эта статья затрат учитывает выплаты по заработной плате за выполненную работу, исчисленные на основании тарифных ставок и должностных окладов в соответствии с принятой в организации – разработчике системой оплаты труда. Затраты на оплату труда рассчитывают по форме, приведенной в таблице 4.5.

Таблица 4.5 – Расчет затрат на оплату труда

Категория работника	Трудоемкость разработки ПП, чел. х ч	Часовая ставка, тг/ч	Сумма, тг
Начинающий Инженер-программист	1 х 203	326,08	67000
ИТОГО затрат на оплату труда			67000

Общая сумма затрат на оплату труда  $Z_{тр}$  рассчитывается по формуле (3).

$$Z_{тр} = \sum_{i=1}^n ЧС_i * T_i. \quad (3)$$

Часовая ставка работника, рассчитанная по формуле, равняется - 326,08 тг/час.

#### 4.7 Социальный налог

В статью социальный налог включается 9.5% от затрат на оплату труда. При этом необходимо учесть, что пенсионные отчисления (10% от заработной платы) не облагаются налогом. Таким образом чтобы рассчитать социальный налог нам необходимо отнять 10% от  $Z_{тр}$  и умножить на 9.5%:

Для начала рассчитаем сумму пенсионных отчислений по формуле (4):

$$Z_{по} = \frac{Z_{тр} * 10\%}{100\%} = \frac{67000 * 10\%}{100\%} = 6700 \text{ тг.} \quad (4)$$

Затем рассчитаем социальные отчисления по формуле (5)

$$Z_{со} = \frac{(Z_{тр} - Z_{по}) * 9.5\%}{100\%} = \frac{(67000 - 6700) * 9.5\%}{100\%} = 5728,5 \text{ тг.}$$

#### 4.8 Амортизация основных фондов

Амортизационные отчисления рассчитываются по формуле (5).

$$Z_{AM} = \sum_{i=1}^n \frac{F_i * N_{ai} * T_{нpi}}{100 * T_{эф}} \quad (5)$$

Необходимо учитывать, что в стоимость ОФ также входят затраты на такие вещи как: доставка, монтаж, установка программного обеспечения и оборудования. Норма амортизации для линейного способа начисления вычисляется по формуле (6).

$$N_{ai} = \frac{100}{T_{ни}} \quad (6)$$

Использование ОФ варьируется от 3 до 10 лет. Ноутбук используется в течении 7 лет, оптический сетевой терминал – 10 лет, программное обеспечение -2 года. Используя формулу (6), заполним Таблицу 4.6 для отображения амортизации основных фондов.

$$N_{A1} = 100/7 = 14,29 \%,$$

$$N_{A2} = 100/10 = 10 \%,$$

$$H_{Az} = 100/2 = 50 \%$$

Таблица 4.6 – Амортизация основных фондов

Наименование оборудования и ПО	Стоимость оборудования и ПО, тг	Годовая норма амортизации, %	Эфф-ый фонд врем. работы оборудования и ПО, ч/год	Время работы оборудования и ПО для разработки ПП, ч	Сумма, тг
Ноутбук	200000	14,29	2016	203	2877,8
Оптический сетевой терминал	17000	10	2016	203	171,2
Microsoft Windows Basic 7 64 bit	50000	40	2016	203	2013,9
Антивирус	12000	40	2016	203	483,3
Adobe Photoshop cs 3	14000	40	2016	203	563,9
ИТОГО амортизация основных фондов					6110,1

#### 4.9 Прочие затраты

Прочие затраты включают в себя затраты на арендную плату с коммунальными услугами, расходы на рекламу расходы на лицензирование и сертификацию, а также канцелярские и прочие хозяйственные затраты.

Учитывая, что в месяце 25 семичасовых рабочих дней то мы получим:

$$T_p = \frac{203}{7*25} = 1,16 \text{ (мес)}. \quad (7)$$

В среднем арендная плата в месте с коммунальными услугами составляет 150000 тенге. Следовательно, арендная плата составит:

$$Z_{Ap} = 150000 * 1,16 = 174000 \text{ тг}. \quad (8)$$

Таблица 4.7 – Перечень затрат на прочие расходы

Наименование	Сумма, тг
Арендная плата	174000

Хостинг и домен	6000
Беспроводная мышь	2000
Итого	182000

#### 4.10 Смета затрат на разработку ПП

Рассчитав все возможные затраты, связанные с созданием программного обеспечения, опираясь на расчеты, составила смету общих затрат и отразила ее в Таблице 4.8.

Таблица 4.8 – Смета общих затрат

Наименование затрат	Сумма, тг
1. Материальные затраты – электроэнергия	4553,2
2. Затраты на оплату труда	67000
3. Отчисления на социальные нужды	5728,5
4. Амортизация основных фондов	6110,1
5. Прочие затраты	182000
ИТОГО по смете	266157,1

#### 4.11 Определение возможной (договорной) цены ПП

Величина возможной (договорной) цены ПП устанавливается на основе эффективности, качества и сроков её выполнения на уровне, отвечающим экономическим интересам заказчика (потребителя) и исполнителя.

$$Ц_{д} = З_{\text{нир}} \left( 1 + \frac{P}{100} \right), \quad (9)$$

P – средний уровень рентабельности ПП. % (принимается в размере 20 %)

$$Ц(д) = 266157,1 * (1+0,2) = 319388,52 \text{ (тенге).}$$

Далее определяется цена реализации с учетом налога на добавленную стоимость (НДС), ставка (НДС) устанавливается законодательно. Налоговым Кодексом РК. На 2018 год ставка НДС установлена в размере 12%.

Цена реализации с учетом НДС рассчитывается по формуле:

$$Ц_{р} = Ц_{д} + Ц_{д} * \text{НДС}, \quad (10)$$

$$Ц_{р} = 319388,52 + 319388,52 * 12\% = 357715,14 \text{ тенге.}$$

Рассчитанную возможную цену ПП можно округлить 357715 тенге.

#### 4.12 Оценка социально-экономических результатов функционирования ПП

В случае если оценивать экономический эффект с учетом всех деталей, то капитальные затраты на проектирование и внедрение рассчитываются с учетом длительности работ на этом этапе. Итак, рассмотрим подробнее расчет капитальных затрат на проектирование и внедрение системы автоматизации. Под проектированием понимается совокупность работ, которые необходимо выполнить, чтобы спроектировать систему, часть системы или поставленную задачу. Под внедрением понимается комплекс работ по вводу в промышленную эксплуатацию системы с возможными ее доработками.

В компании работает сотрудник, обязанностью которого является консультация клиентов по всевозможным вопросам касательно компании.

Зарботная плата этого работника составляет 100000 тг/месяц. Это 12000000 тг/год.

Так же необходимо учесть пенсионные отчисления и социальный налог, которые рассчитаем по формуле (11)

$$Z_{\text{по}} = \frac{Z_{\text{гот}} * 10\%}{100\%} = \frac{12000000 * 10\%}{100\%} = 1200000 \text{ тг.} \quad (10)$$

Затем рассчитаем социальные отчисления:

$$Z_{\text{со}} = \frac{(Z_{\text{тр}} - Z_{\text{по}}) * 9.5\%}{100\%} = \frac{(12000000 - 1200000) * 9.5\%}{100\%} = 1026000 \text{ тг,}$$

$$Z_{\text{тр1}} = Z_{\text{гот}} + Z_{\text{со}} = 12000000 + 1026000 = 13026000 \text{ тг.}$$

Таким образом мы видим, что оплата заработной платы данному сотруднику составляет 13026000 тенге. Но если использовать разработанный веб-сайт в качестве источника информации, то можно обойтись без данного сотрудника. То есть можно не тратить денег в принципе на консультацию.

Экономия затрат от внедрения программного продукта определяется по формуле (11):

$$\mathcal{E} = Z_{\text{тр}} - Z_{\text{ом}}, \quad (11)$$

где  $Z_{тр}$  – затраты до внедрения системы;  
 $Z_{ом}'$  – затраты после внедрения системы.  
Подставив значения получим следующее:

$$\mathcal{E} = 13026000 - 0 = 13026000 \text{ тг.}$$

Срок окупаемости программного продукта определяется по формуле (12):

$$T_{ок} = C/\mathcal{E},$$

где  $C$  – затраты на разработку и внедрение системы, тенге;  
 $\mathcal{E}$  – экономия затрат от внедрения системы, тенге/год.  
Подставив значения, получим:

$$T_{ок} = \frac{357715}{13026000} = 0,027 \approx 0,324 \text{ месяца} \approx \text{пол месяца.}$$

По результатам расчета экономической эффективности проектирования и внедрения средства автоматизации сразу можно сказать, что это выгодно. Хотя выгода и косвенная, но, как правило, заметная в средне и долгосрочной перспективе.

В процессе расчета экономической эффективности необходимо учитывать одно свойство автоматизации. Заключается оно в следующем: чем больше средств и времени потрачено на автоматизацию, тем выше экономический эффект от внедрения. Объясняется это довольно просто: если качественно подойти к выбору программного продукта, качественно проработать все процессы на этапе проектирования и внедрения, все описать и отладить, то в последующем будет потрачено гораздо меньше средств на эксплуатацию программы.

По расчетам на внедрение ПП было затрачено 266157,1 тенге, в том числе на оплату труда, оборудование, электроэнергию и прочие затраты.

В будущем программный продукт не требует каких-либо значительных затрат на комплектующие при эксплуатации.

## 5 Безопасность жизнедеятельности

Разработка веб приложения достаточно трудоемкий процесс, который требует оптимальных условий таких как комфортное рабочее место, достаточное количество освещения которое так же не будет мешать, а также приемлемый микроклимат. Данная дипломная работа под названием «Разработка и управление web-ориентированной информационной системы для предприятия «КазЭнергоПромТорг»» осуществлялась в техническом отделе строительной компании «КазЭнергоПромТорг», которая находится по адресу г. Алматы, ул. Суюнбая 20. Отдел находится на 2 этаже в кабинете №20. Рабочий день начинался с 10.00 утра до 18.00 вечера в будние дни с перерывами на обед с 12.00 до 13.00.

### 5.1 Анализ условий труда

#### 5.1.1 Описание рабочего места

Согласно ГОСТу 12.2.032-78 конструкция рабочего места и расположение всех его элементов равны антропометрическим, физическим и психологическим требованиям. Выполнение трудовых операций было обеспечено в пределах зоны досягаемости и оптимальной зоны моторного поля.

В помещении имеются 2 рабочих стола, оснащенные персональными компьютерами, принтерами, сканерами, сетевым оборудованием, лицензионным программным обеспечением и т.д. В отделе ежедневно кроме меня находится еще 1 человек, это другой сотрудник технического отдела.

Также помещение оснащено аптечкой первой медицинской помощи.

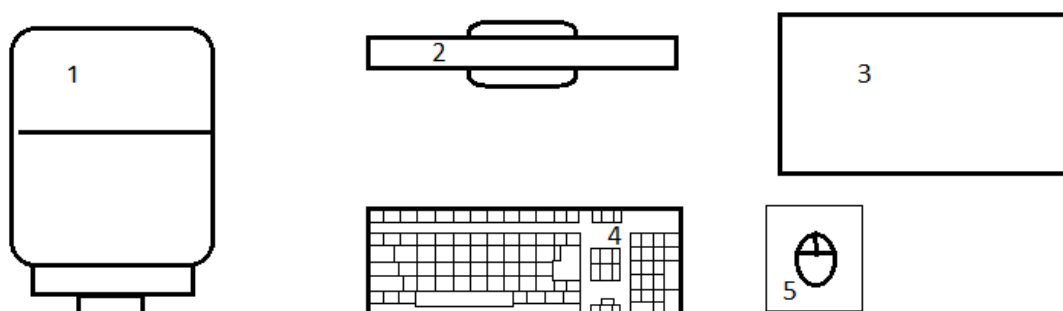


Рисунок 49 – Размещение основных и периферийных составляющих ПК на рабочем столе

- 1 – Принтер, сканер;
- 2 – монитор;

- 3 – стойка для документов;
- 4 – клавиатура;
- 5 – мышь.

На рабочем столе не имеется ничего постороннего, кроме элементов нужных для комфортной работы. Так как минимальное расстояние от монитора до глаз 50 см, монитор относительно меня был расположен на 60 см.

Создание благоприятных условий труда имеет большое значение как и для удобства работников так и для быстрого и эффективного выполнения работы.

Размеры поверхности рабочей зоны каждого сотрудника имеют следующие значения:

- ширина – 1200 мм;
- глубина – 800 мм;
- высота – 725 мм.

Столы с мониторами расположены друг от друга на расстоянии 2 м.

Для комфорта сотрудника рабочий стол обладает следующим свойствами:

- высота стола выбирается таким образом, что а ним можно комфортно и свободно сидеть продолжительное время;
- поверхность стола обладает свойствами, исключающими появление бликов в поле зрения программиста;
- в конструкции стола предусмотрены выдвижные ящики для хранения документов и канцелярии.

Таким образом, помещение отвечает требованиям санитарии и гигиены труда, согласно установленным нормам.

### **5.1.2 Освещение рабочего места**

Правильно спроектированная и выполняемая светотехническая продукция снижает усталость и повышает производительность, имеющая хорошие характеристики для окружающей и средств производства, а также обеспечивая положительный психологический эффект, повышает безопасность и снижает фактор получения травм.

Плохое освещение может вызвать зрительное напряжение, что приводит к преждевременной усталости. Также плохую роль может сыграть и яркое освещение, это может быть чревато раздражением и жжением в глазах. Все эти факторы могут привести к болезни человека, поэтому правильный расчет освещения играет очень важную роль в работе.

Так как я работала в помещении, в котором установлены компьютеры и где выполнялись зрительные работы высокой точности, то использовалось искусственное освещение. В этом помещении установлены люминесцентные лампы, так как они по сравнению с обычными лампами накаливания имеют множество преимуществ. Общая освещенность помещения составляла 300 лк\.



Так же помещение освещено достаточно равномерно что не перезагружает во время работы. Кабинет, котором я работала не имеет никаких нарушений в плане освещения и полностью соответствует всем нормам, указанным в Санитарно-эпидемиологических требованиях к освещению в помещениях для эксплуатации компьютеров и видеотерминалов от 21 января 2015 года №38.

### **5.1.3 Нормирование шума**

На рабочем месте в помещении, где работа производится на ПК, или производится научная или образовательная деятельность уровень шума допускается не более 50 децибел акустических в соответствии с настоящими Санитарно-эпидемиологическими требованиями к условиям работы при воздействии физических факторов от компьютеров и видеотерминалов.

Уровень шума и вибрации в данном помещении соответствуют вышеуказанным нормативным требованиям.

### **5.1.4 Пожарная безопасность**

В помещениях с ПК и ВТ следует крайне осторожно соблюдать пожарную безопасность, так как причиной возгорания может послужить целый ряд причин: короткие замыкания, перегрузки, большие переходные сопротивления, искрение и электрические дуги, статическое электричество; применение электрооборудования, не соответствующего категориям помещений по пожарной безопасности; отсутствие в радиоэлектронных устройствах устройств защиты от перегрузок по току и напряжению, а также тепловой защиты элементов.

В офисе, в котором осуществлялась моя дипломная работа, установлены углекислые огнетушители, так как они являются наиболее эффективными при возгорании. Также в кабинете висит план эвакуации здания, который все работники подробно изучили.

### **5.1.5 Электромагнитное и ионизирующее излучения**

Допустимые значения уровней неионизирующих электромагнитных излучений на рабочих местах у ВТ и ПК указаны в санитарных правилах «Санитарно-эпидемиологические требования к условиям работы при воздействии физических факторов от компьютеров и видеотерминалов» от 21 января 2015 года №38. В таблице 5.2 показаны значения уровней неионизирующих электромагнитных излучений помещения, где разрабатывался проект.

Таблица 5.1 – Значения уровней неионизирующих электромагнитных излучений помещения

Наименование параметра	Часть ПК	Расстояние	Значение
Напряженность электрического поля вокруг ПК, ВТ:	Монитор	На уровне головы	25 Вольт на метр
в диапазоне частот 5 – 2000 Герц в диапазоне частот 2 – 400 кГц:	Монитор		
Плотность магнитного потока вокруг ПК, ВТ: в диапазоне частот 5 – 2000 Гц: в диапазоне частот 2 – 400 кГц:	Монитор Монитор	На уровне головы	250 нанно Тесла

Значения уровней неионизирующих электромагнитных излучений в данном помещении соответствуют санитарным нормам.

## 5.2 Параметры микроклимата

В санитарных нормах «Санитарно-эпидемиологические требования к условиям работы с источниками физических факторов (компьютеры и видеотерминалы), оказывающих воздействие на человека» установлены величины параметров микроклимата, создающие комфортные условия. Эти нормы устанавливаются в зависимости от времени года, характера трудового процесса и характера производственного помещения (значительные или незначительные тепловыделения).

В помещении, в котором осуществлялась моя дипломная работа установлен кассетный кондиционер Gree-18 R410A, который вентилирует помещение с забором воздуха с улицы, фильтруя его.

Функции кассетного кондиционера Gree-18 R410A:

- Производительность по холоду – 5000 Вт
- Потребляемая мощность в режиме охлаждения – 2450 Вт
- Рекомендуемая площадь охлаждения/обогрева – 45/50 м<sup>2</sup>

- Потребляемая мощность в режиме обогрева – 2550 Вт
- Потребляемый ток в режиме охлаждения – 9,55 А
- Потребляемый ток в режиме обогрева – 2,9 А;
- Расход воздуха внутренним блоком – 550 м3/час.



Рисунок 50 – Кассетный кондиционер Gree-18 R410A

В таблице 5.2 приведены параметры микроклимата помещения отдела информационных технологий.

Таблица 5.2 – Значения параметров микроклимата для данного помещения.

Период года	Температура воздуха, °С	Температура поверхностей, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	22-24	21-25	60-40	0,1
Теплый	23-25	22-26	60-40	0,1

### 5.2.1 Расчет вентиляции

Так как кассетный кондиционер проводит вентиляцию помещения, необходимо вычислить расчет вентиляции и сравнить результаты с характеристикой кассетного кондиционера Gree-18 R410A. Расчет вентиляции будет производиться для кабинета технического отдела. Общая площадь помещения составляет 24 квадратных метров, высота 3 м.

Расчет воздухообмена по кратности рассчитывается по формуле (13).

$$L = n * S * H \text{ (м3/час)} . \quad (13)$$

где  $L$  – требуемая производительность приточной вентиляции, м<sup>3</sup>/ч;  
 $n$  – нормируемая кратность воздухообмена, час;  
 $S$  – площадь помещения, м<sup>2</sup>;  
 $H$  – высота помещения, м.

Нормируемая кратность воздухообмена есть отношение объема воздуха поступающего в помещение в течение часа к объему помещения. Согласно СНиП РК 3.02-25-2004 нормируемая кратность воздухообмена для кабинета образовательного учреждения  $n=3$ .

Таблица 5.3 – Нормируемая кратность воздухообмена

Помещение	Кратность воздухообмена
Кабинет	3

Подставив все полученные данные в формулу (13) вычислим требуемую производительность приточной вентиляции:

$$L_1 = 3 * 24 * 3 = 216 \text{ м}^3/\text{час} .$$

Расчет воздухообмена по количеству людей осуществляется по формуле (14).

$$L_2 = N * L_{\text{норм}},$$

(14)

где  $L_2$  – требуемая производительность приточной вентиляции, м<sup>3</sup>/ч;  
 $N$  – количество людей;  
 $L_{\text{норм}}$  – норма расхода воздуха на одного человека.

В кабинете ежедневно находится 3 человека, соответственно  $N=3$ . Согласно СНиП РК 3.02-25-2004 в кабинетах, лабораториях норма расхода воздуха на одного человека составляет - 40 м<sup>3</sup>/час. Подставив данные в формулу (14) получим следующее значение:

$$L_2 = 3 * 40 = 120 \frac{\text{м}^3}{\text{час}}.$$

В жилых и общественных помещениях постоянным вредным выделением является выдыхаемая людьми углекислота (СО<sub>2</sub>). Определение потребного воздухообмена производится по количеству углекислоты, выделяемой человеком и по допустимой концентрации её.

Потребный воздухообмен определяется по формуле (15).

$$(15) \quad L_3 = \frac{N * G}{X_n - X_b} \text{ (м}^3\text{/час),}$$

где  $L_3$  – потребный воздухообмен, м<sup>3</sup>/час;  
 $G$  – количество вредных веществ, выделяющихся в воздух помещения, г/ч;  
 $N$  – количество людей;  
 $X_n$  – предельно допустимая концентрация вредности в воздухе рабочей зоны помещения, согласно ГОСТ 12.1.014-84;  
 $X_b$  – максимально возможная концентрация той же вредности в воздухе населенных мест.

Количество углекислоты, выделяемой одним человеком при легкой работе составляет  $G = 23$  л/ч. Согласно ГОСТ 12.1.014-84 предельно допустимая концентрация углекислоты в воздухе рабочей зоны помещения равна  $X_n = 1,25$  л/м<sup>3</sup>, а для больших городов (свыше 300 тыс. жителей)  $X_b = 0,5$  л/м<sup>3</sup>. Подставляем значения в формулу (15) и определяем потребный воздухообмен:

$$L_3 = \frac{3 * 23}{1.25 - 0.5} = 92 \text{ м}^3\text{/час.}$$

Рассчитав производительность по трем критериям, выбираем наибольший из них, то есть по кратности воздухообмена, где  $L_1 = 216$  м<sup>3</sup>/час. Исходя из полученных данных, видно, что кассетный кондиционер Gree-18 R410A подходит данному помещению.

## Заключение

При выполнении дипломной работы выяснилось, что создание сайта весьма трудоемкий процесс, требующий серьезного подхода со всех касающихся его сторон. основополагающими задачами работы были привлечение новых клиентов, создание удобного просмотра информации для посетителей сайта, простая форма обратной связи.

Одна из задуманных и достигнутых целей была возможность показать, что в нашем современном мире программирование является одной из важных частей во всех отраслях деятельности. Была выполнена большая работа, при которой был разработан полностью функционирующий сайт, с интересным и современным дизайном, включая функциональный и доступный каждому интерфейс. Разбитые по блокам категории дают возможность сконцентрировать внимание на интересующей вас информации. При выполнении работы были раскрыты многие возможности написания сайта на языке HTML и JavaScript. Данные языки позволяют сочетать себе простоту и функциональность исполнения.

В результате проделанной работы были сделаны выводы о совершенной работе. Весь проект послужил отличной платформой для тестирования эффективности технологий создания сайтов и представляет собой современное техническое решение проблемы недоступности реального офиса в реальном времени. Подобное решение позволяет потенциальным клиентам получить доступ к интересующей его продукции. При этом имеется возможность не только просмотреть, но и обсудить детали вероятного заказа с обладателем данного ресурса. Нынешние тенденции развития сайтов говорят лишь о скором продвижении технологий создания выполненного в данной работе продукта.

В разделе безопасности и жизнедеятельности была проведен анализ условий труда, описание и характеристика помещений, схема размещения рабочих мест. Проведена оценка микроклимата рабочей территории. А также рассчитаны и выбраны источники искусственного и естественного освещения помещения.

В разделе технико-экономического обоснования были проведены различные расчеты, составлена смета затрат, а также была выведена оценка социально-экономических результатов функционирования ПП

## Список литературы

- 1 Борисов, Е. Ф. Основы экономики: Учебное пособие / Е. Ф. Борисов. –М.: Юрайт – Издат, 2009. – 316 с.
- 2 Экономическая теория: Учебник/ под ред. В.Д. Камаева, Е.И.Лобачевой. – М.: Юрайт-Издат, 2010. – 557с.
- 3 Экономика и анализ деятельности предприятий / Т.П. Елисеева, М.Д.Молев, Н.Г. Трегулова. - Ростов-на-Дону: Феникс, 2011. - 480 с.
- 4 Стивен Хольцнер. PHP в примерах. / Стивен Хольцнер. М.: «Бином-Пресс», 2011 г. Пер. с англ. 352 с
- 5 Гутманс Э., Баккен С, Ретанс Д. PHP 5. Профессиональное программирование. / Пер. с англ. СПб: Символ- Плюс, 2009. 704 с., ил.
- 6 Веллинг Л. PHP и MySQL Web-разработчикам.- М.: SAMS, 2012. - 893с.
- 7 Томсон Лаура. Разработка Web-приложений на PHP и MySQL: Пер. сангл. /Лаура Томсон, Люк Веллинг. – 2-е изд., испр. – СПб: ООО «ДиаСофтЮП», 2013. – 672 с.
- 8 Дубаков М.А. Создание Web-страниц: искусство верстки. – Мн.:Новое знание, 2014. – 287 с.
- 9 Ларри Ульман. Ульман Л. Основы программирования на PHP:/ЛарриУльман. Пер. с англ. -М.: ДМК Пресс, 2011. -288 с.: ил. (Самоучитель).
- 10 Нидерст Б. Web-мастеринг для профессионалов. «Настольный справочник ». - М.: БИНОМ, 2013. - 313 с.
- 11 Аманбаев У.А. Экономика предприятия. – А.: Бастау, 2012. – 100 с.
- 12 Налоговый Кодекс Республики Казахстан <http://www.salyk.kz>
- 13 Создание рабочего места на предприятии, Кремлев А. П. 2010.
- 14 СНиП РК 2.09.04 Общие строительные нормы и правила устройства систем вентиляции и кондиционирования воздуха.
- 15 СНиП РК 2.04.-05.2012 Естественное и искусственное освещение. Общие требования.
- 16 Дженнифер Нидерст Роббинс "HTML5, CSS3 и JavaScript. Исчерпывающее руководство", 2014 год.
- 17 Лоусон, Шарп. Изучаем HTML 5.- М.:Наука 2000. – 432 с.
- 18 Анаб Бабаев «Создание сайтов», 2014 год.