

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН

Satbayev University

Институт кибернетики и информационных технологий

Кафедра кибербезопасность, обработка и хранение информации

Цай Алла Андреевна

Разработка информационной системы web-дизайнера на основе современных
технологических решений

ДИПЛОМНАЯ РАБОТА

Специальность 5В070300 – Информационные системы

Алматы, 2020

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН

Satbayev University

Институт кибернетики и информационных технологий

Кафедра кибербезопасность, обработка и хранение информации

ДОПУЩЕН К ЗАЩИТЕ

Заведующий кафедрой

КБОиХИ

канд. техн. наук, доцент

_____ Н. А. Сейлова

« _____ » _____ 20__ г.

ДИПЛОМНАЯ РАБОТА

На тему: Разработка информационной системы web-дизайнера на основе современных технологических решений

Специальность 5В070300 – Информационные системы

Выполнил:

Цай. А. А.

Научный руководитель

к.т.н, доцент

_____ Косников В.А.

« _____ » _____ 20__ г.

Алматы, 2020

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН

Satbayev University

Институт кибернетики и информационных технологий

Кафедра кибербезопасность, обработка и хранение информации

5B070300 – Информационные системы

УТВЕРЖДАЮ

Заведующий кафедрой

КБОиХИ

канд. техн. наук, доцент

_____ Н. А. Сейлова

«_____» _____ 20__ г.

ЗАДАНИЕ

на выполнение дипломной работы

Обучающемуся: Цай Алла Андреевна

Тема: Разработка информационной системы web-дизайнера на основе современных технологических решений

Утверждена приказом Ректора Университета № 762-б от 27.01.2020г.

Срок сдачи законченной работы - 27.05.2020г.

Исходные данные к дипломному проекту: результаты преддипломной практики, результат обзора современного состояния по данной теме, сбор теоретического материала.

Краткое содержание дипломной работы:

- а) Современное состояние вопроса проектирования web-сайтов с использованием современных технологий дизайна;
- б) Структура, диаграммы и алгоритмы функционирования системы
- в) Информационное и программное обеспечение системы;
- г) Внедрение системы;

Рекомендуемая основная литература: *из 10 наименований*

ГРАФИК

подготовки дипломной работы (проекта)

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
Современное состояние вопроса проектирования web-сайтов с использованием современных технологий дизайна	17.02.2020г.	
Структура, диаграммы и алгоритмы функционирования системы	11.04.2020г.	
Информационное и программное обеспечение системы	21.04.2020г.	
Внедрение системы	22.05.2020г.	

Подписи

консультантов и нормоконтролера на законченную дипломную работу (проект) с указанием относящихся к ним разделов работы (проекта)

Наименование разделов	Консультанты, Ф.И.О. (уч.степень, звание)	Дата подписания	Подпись
Разработка информационной системы web-дизайнера на основе современных технологических решений	Косников В.А., кандидат технических наук, доцент		
Нормоконтролер	Бауыржан М.Б., тьютор		
Программная часть	Кабдуллин М.А., ассистент		

Научный руководитель: _____ Косников В.А.

Задание принял к исполнению обучающийся _____ Цай А.А.

Дата " 27 " _____ 01 2020г.

Протокол анализа Отчета подобия Научным руководителем

Заявляю, что я ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

Автор: Цай А.А

Название: Разработка информационной системы web-дизайнера на основе современных технологических решений

Координатор: Вячеслав Косников

Коэффициент подобия 1: 12,5

Коэффициент подобия 2: 6,9

Замена букв: 3

Интервалы: 0

Микропробелы: 0

Белые знаки: 0

После анализа Отчета подобия констатирую следующее:

- обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, признаю работу самостоятельной и допускаю ее к защите;
- обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
- обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, не допускаю работу к защите.

Обоснование:

.....

.....
Дата

.....
Подпись Научного руководителя

Протокол анализа Отчета подобия

заведующего кафедрой / начальника структурного подразделения

Заведующий кафедрой / начальник структурного подразделения заявляет, что ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

Автор: Цай А.А

Название: Разработка информационной системы web-дизайнера на основе современных технологических решений

Координатор: Вячеслав Косников

Коэффициент подобия 1:12,5

Коэффициент подобия 2:6,9

Замена букв:3

Интервалы:0

Микропробелы:0

Белые знаки:0

После анализа отчета подобия заведующий кафедрой / начальник структурного подразделения констатирует следующее:

- обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, работа признается самостоятельной и допускается к защите;
- обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
- обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, работа не допускается к защите.

Обоснование:

.....
.....
.....
.....
.....
.....

Дата

Подпись заведующего кафедрой /

начальника структурного подразделения

Окончательное решение в отношении допуска к защите, включая обоснование:

.....
.....
.....
.....
.....

Дата

.....
Подпись заведующего кафедрой /

начальника структурного подразделения

НАЗВАНИЕ:

Разработка информационной системы web-дизайнера на основе современных технологических решений

АВТОР:

Цай А.А

НАУЧНЫЙ РУКОВОДИТЕЛЬ:

Вячеслав Косников

ПОДРАЗДЕЛЕНИЕ:

ИКИИТ

ДАТА ЗАГРУЗКИ ДОКУМЕНТА:

2020-05-31 12:05:15

ЧИСЛО ПРОВЕРЕК ДОКУМЕНТА: ⓘ

1

ПРОПУЩЕННЫЕ ВЕБ-СТРАНИЦЫ: ⓘ

Уровень заимствований

Обратите внимание! Высокие значения коэффициентов не означают плагиат. Отчет должен быть проанализирован экспертом.



Предупреждение и сигналы тревоги

В этом разделе вы найдете информацию, касающуюся манипуляций в тексте, с целью изменить результаты проверки. Для того, кто оценивает работу на бумажном носителе или в электронном формате, манипуляции могут быть невидимы (может быть также целенаправленное вписывание ошибок). Следует оценить, являются ли изменения преднамеренными или нет.

Замена букв Использование символов из другого алфавита - может указывать на способ обойти систему, поэтому следует установить их использование.	3	показать в тексте
Интервалы Количество увеличенного расстояния между буквами (просим определить является ли расстояние имитацией пробела, так как исходя слова могут быть написаны слитно).	0	показать в тексте
Микропробелы Количество пробелов с нулевым размером - необходимо проверить влияют ли они на неправильное разделение слов в тексте.	0	показать в тексте
Белые знаки Количество символов, выделенных белым цветом, пожалуйста, проверьте не используются ли белые символы вместо пробела, соединяя слова (в отчете подобия система и меняет автоматически цвет букв в черный, чтобы их сделать видимыми).	0	показать в тексте

Заимствования по списку источников

Посмотрите список и проанализируйте, в особенности, те фрагменты, которые превышают КП №2 (выделенные жирным шрифтом). Используйте ссылку «Обозначить фрагмент» и посмотрите, являются ли выделенные фрагменты повторяющимися короткими фразами, разбросанными в документе (совпадающие сходства), многочисленными короткими фразами расположенные рядом друг с другом (парафразирование) или обширными фрагментами без указания источника ("криптоцитаты").

10 самых длинных фраз (7,34 %)

Десять самых длинных фрагментов найденных во всех доступных ресурсах.

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ И АДРЕС ИСТОЧНИКА URL (НАЗВАНИЕ БАЗЫ)	АВТОР	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ
1	https://knowledge.allbest.ru/programming/3c0a65635b2ac79b5d43a89421316c27_0.html		67 1,40 %
2	http://dverkin66.ru/id-8483.html		44 0,92 %
3	https://vk.com/itgorodrus		40 0,83 %
4	http://buxtome.ru/tag/%D0%9F%D1%80%D0%B0%D0%B2%D0%B8%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5/		38 0,79 %
5	http://www.rusarticles.com/veb-dizajn-statya/tendencii-razvitiya-veb-dizajna-sovremennye-tehnologii-sozdaniya-sajtov-311060.html		32 0,67 %
6	https://vk.com/itgorodrus		29 0,60 %
7	https://stfalcon.com/ru/blog/post/10-most-common-web-design-mistakes		29 0,60 %

8	http://www.rusarticles.com/veb-dizajn-statya/tendencii-razvitiya-veb-dizajna-sovremennye-texnologii-sozdaniya-sajtov-311060.html	26	0,54 %
9	https://stfalcon.com/ru/blog/post/10-most-common-web-design-mistakes	26	0,54 %
10	https://stfalcon.com/ru/blog/post/10-most-common-web-design-mistakes	21	0,44 %

из базы данных RefBooks (0,00 %)

Все фрагменты найдены в базе данных RefBooks, которая содержит более 3 миллионов текстов от редакторов и авторов.

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	АВТОР	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (КОЛИЧЕСТВО ФРАГМЕНТОВ)
---------------------	----------	-------	---

ЗАИМСТВОВАНИЙ НЕ НАЙДЕНО

из домашней базы данных (0,63 %)

Все фрагменты найдены в базе данных вашего университета.

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	АВТОР	ДАТА ИНДЕКСАЦИИ	ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)	
1	Диплом на антиплагиат Satbayev University (ИКиИТ)	Супрун Радион	2019-05-06	30 (2)	0,63 %

из программы обмена базами данных (0,27 %)

Все фрагменты найдены в базе данных других университетов.

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ НАЗВАНИЕ БАЗЫ ДАННЫХ	АВТОР	ДАТА ИНДЕКСАЦИИ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (КОЛИЧЕСТВО ФРАГМЕНТОВ)	
1	«Разработка и создание информационно - аналитического web - сайта Развлекательно- новостной портал» TURAN-ASTANA UNIVERSITY (TAU) (.....)	Абулхаир Каукуенов	2020-05-10	13 (1)	0,27 %

из интернета (11,61 %)

Все фрагменты найдены в глобальных интернет-ресурсах открытого доступа.

ПОРЯДКОВЫЙ НОМЕР	ИСТОЧНИК URL	ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)	
1	https://stfalcon.com/ru/blog/post/10-most-common-web-design-mistakes	169 (11)	3,52 %
2	https://vk.com/igorodrus	69 (2)	1,44 %
3	https://knowledge.allbest.ru/programming/3c0a65635b2ac79b5d43a89421316c27_0.html	67 (1)	1,40 %
4	http://www.rusarticles.com/veb-dizajn-statya/tendencii-razvitiya-veb-dizajna-sovremennye-texnologii-sozdaniya-sajtov-311060.html	58 (2)	1,21 %
5	http://buxtome.ru/tag/%D0%9F%D1%80%D0%B0%D0%B2%D0%B8%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5/	55 (2)	1,15 %
6	http://dverkin66.ru/id-8483.html	44 (1)	0,92 %
7	https://www.ps.kz/hosting/shared	25 (2)	0,52 %
8	https://design-orbita.com.ua/polnoe-rukovodstvo-po-sozdaniyu-sajtov/	22 (2)	0,46 %
9	https://toxadab.ru/works/category/development	16 (2)	0,33 %
10	https://www.opiq.kz/kit/16/chapter/821	11 (1)	0,23 %
11	https://irbis.amursu.ru/DigitalLibrary/AmurSU_Edition/8430.pdf	11 (1)	0,23 %
12	http://bseu.by/it/oivt/tema4v2.htm	10 (1)	0,21 %

АНДАТПА

Бұл дипломдық жұмыс веб-сайтты әзірлеуге арналады, оның міндеті клиенттерге сайттарды жасаумен байланысты қызметтерді көрсету, тегін тестілеу мүмкіндігі бар дайын шаблондарды және басқа да жаңа құралдарды ұсыну болып табылады. Осы жұмыста веб-сайттарды жобалау мәселелерінің қазіргі жай-күйіне зерттеу жүргізілді. Жүйенің жұмыс істеу алгоритмдері әзірленді және сипатталған. Веб-сайтты әзірлеу кезінде қолданылған ақпараттық және бағдарламалық қамтамасыз етудің қысқаша сипаттамасы мен негізгі артықшылықтары келтірілген. Сайт PHP 7 бағдарламалау тілін, PhpStorm интеграцияланған даму ортасымен құрылған. Деректер базасымен өзара әрекеттесу үшін MySQL Workbench пайдаланылды.

АННОТАЦИЯ

Данная дипломная работа посвящается разработке веб-сайта, задачами которого является оказание клиентам услуг, связанных непосредственно с созданием сайтов, предоставление готовых шаблонов с возможностью бесплатного тестирования и других новых инструментов. В настоящей работе проведено исследование современного состояния вопроса проектирования веб-сайтов. Разработаны и описаны алгоритмы функционирования системы. Приведены краткое описание и основные преимущества информационного и программного обеспечения, которое было использовано при разработке веб-сайта. Сайт создан с применением языка программирования PHP7, интегрированной средой разработки PhpStorm. Для взаимодействия с базой данных использовался MySQL Workbench.

THE SUMMARY

This thesis is dedicated to the development of a website, whose tasks are to provide customers with services related directly to the creation of sites, providing ready-made templates with the possibility of free testing and other new tools. In this paper, a study of the current state of the issue of website design is conducted. Algorithms of system functioning are developed and described. A brief description and main advantages of the information and software that was used in the development of the website are given. The site was created using the PHP 7 programming language and the PhpStorm integrated development environment. MySQL Workbench was used to interact with the database.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	14
1. Современное состояние вопроса проектирования web-сайтов с использованием современных технологий дизайна	15
1.1 Обзор и анализ современных технологий дизайна	15
1.2 Анализ опыта, ошибок и рисков Web-дизайнера	15
1.3 Постановка задачи	17
2. Структура, диаграммы и алгоритмы функционирования системы	18
2.1 Функциональная структура системы	18
2.2 Диаграмма вариантов использования	19
2.3 Диаграмма состояний	20
2.4 Обобщенный алгоритм функционирования	21
2.5 Алгоритм проектирования сайта	22
2.6 Алгоритм взаимодействия с клиентом	24
3. Информационное и программное обеспечение системы	25
3.1 Описание информационного обеспечения	25
3.2 Структура программного обеспечения	27
3.2.1 Системное программное обеспечение	28
3.2.2 Инструментальное программное обеспечение	28
3.2.3 Прикладное программное обеспечение	30
4. Внедрение системы	36
4.1 Инфраструктура размещения сайта и его продвижение	36
4.2 Перспективы развития	36
ЗАКЛЮЧЕНИЕ	38
СПИСОК АББРЕВИАТУР	39
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	40
Приложение А	41
Приложение Б	43
Приложение В	44

ВВЕДЕНИЕ

В настоящее время, в век информационных технологий человечеству не обойтись без Интернета, современных информационных систем и технологий. С каждым днем Интернет расширяет возможности человека в различных сферах таких как образование, развлечения, и, самое главное, в сфере бизнеса.

Большинство людей желают зарабатывать деньги, не выходя из дома. И такой возможностью может обладать каждый человек благодаря Сети. Существует много вариаций бизнеса в Интернете, такие как: интернет-магазин, разработка и раскрутка сайта, интернет-реклама, РТС и РТW. Главными преимуществами организации бизнеса в Интернете можно назвать непрерывную работу сайта 24 часа в сутки и невысокую стоимость его создания и продвижения, а также с помощью сайта можно предоставлять клиентам необходимую информацию, отслеживать их интерес к товарам и услугам, оперативно принимать заказы. Продвижение сайта в Интернете помогает компаниям быстро реагировать на любые изменения рынка, находить новые способы продажи товаров и услуг, а также увеличивать объемы продаж[1].

Целью данной дипломной работы является создание и развитие веб-сайтов с разработкой информационной системы, построенной на основе современных инструментальных решений, сформировав удобную навигацию по сайту для пользователей, а также предоставить им возможность легко и быстро связываться с компанией через любые удобные для них виды коммуникации (почта, социальные сети, Яндекс.диалоги) при помощи интеграции. Помимо этого на сайте компании пользователи могут посетить форум, где могут задавать вопросы, решать различные вопросы по поводу реализации той или иной задачи, также предлагать свои коды.

Конечно же, немаловажным аспектом при посещении веб-сайта пользователем является адаптивный дизайн, который будет подстраиваться под любые гаджеты, подключенные к интернету, и обеспечивать хорошее восприятие веб-страниц.

В первой главе данной работы будут проводиться обзор и анализ современных технологий дизайна, анализ опыта и ошибок веб-дизайнеров, будут поставлены задачи, решаемые в дипломной работе и решаемые компанией.

Во второй главе будут построены функциональная структура, диаграммы и алгоритмы функционирования системы.

В третьей главе предоставится описание информационного и программного обеспечения, необходимого для разработки веб-сайта.

В четвертой главе будет приведено описание инфраструктуры размещения разработанного сайта, а также рассмотрение дальнейших перспектив его развития.

1 Современное состояние вопроса проектирования web-сайтов с использованием современных технологий дизайна

1.1 Обзор и анализ современных технологий web-дизайна

В современном мире веб-дизайн – это понятие, охватывающее широкий круг значений. Веб-дизайн включает в себя несколько направлений, которые сами по себе являются самостоятельными. Объединение этих направлений и является веб-дизайном. Сюда включаются: графический интерфейс, который мы видим, открыв сайт; проектирование навигационной системы сайта; контент-проект ресурса, то есть информационное содержимое сайта, которое учитывается при разработке.

Веб-дизайн можно назвать процессом создания сайта. Веб-дизайн подразумевает отличное знание современных технологий интернета в комбинации с художественными способностями. Именно это сочетание навыков и позволяет создать первоклассные сайты.

Тенденции в веб-дизайне – один из самых востребованных объектов анализа. Разнообразие подходов к дизайну, наличие общепризнанных ориентиров и достаточно яркая динамика постоянно подбрасывают аналитикам новую пищу для размышлений. Интернет развивается быстро, и тенденции, позволяющие с высокой точностью делать прогнозы, способны стать серьезным конкурентным преимуществом как для предприятий, занимающихся обеспечением бизнес-процессов в Сети, так и для компаний, еще только начинающих искать в интернете своего потребителя.

На сегодняшний день, благодаря развитию информационных технологий, имеются подходы проектирования веб-дизайна на основе данных, такие как Data Driven Design (DDD) и Data Informed Design (DID).

Data Driven Design представляет из себя проектирование продукта на основе количественных данных: исследований, тестов, проверки гипотез и Big Data. С приходом концепции DDD аналитический подход основывается на комплексном анализе Big Data. Он анализирует данные об интересах, стиле жизни и социальном статусе всех пользователей.

Data Informed Design – наиболее гибкий подход, представляет из себя дизайн с учетом данных. В отличие от DDD данная концепция не упирается только в аналитику, она также считается с опытом, креативностью дизайнеров, и ставит в приоритет качественные данные, такие как опросы, наблюдения, наряду с количественными. DID является на сегодня наилучшей парадигмой с равными пропорциями данных и творчества [2].

1.2 Анализ опыта, ошибок и рисков web-дизайнера

Ошибки в веб-дизайне влияют на эффективность веб-сайта, поэтому очень важно, чтобы в продукцию выходили сайты, которые не только

визуально привлекательны, но и удобны в использовании. Самые распространенные ошибки в веб-дизайне связаны с тем, что неопытные дизайнеры отталкиваются не от потребностей клиента, а от визуальной привлекательности сайта. Прежде чем начать работу над дизайном, следует убедиться в получении всей необходимой информации от клиента, хорошо понимать и тщательно уточнять его потребности.

В данной главе приведены ошибки совершаемые веб-дизайнерами и способы как их избежать:

а) Непродуманная навигация. Вся структура страницы должна быть продумана так, чтобы пользователь мог понять, где находится то, что ему нужно. Требуется создать четкий навигационный путь, чтобы посетители могли с легкостью перемещаться в пределах сайта, исключив все ненужные ссылки и использовать выпадающее меню.

б) Непоследовательность в дизайне страниц. Если страницы одного сайта имеют разный дизайн, это может сбивать с толку. Помимо этого, размещение навигации и блока поиска не должно изменяться от страницы к странице, так как общее впечатление от сайта будет крайне негативным. Чтобы дизайн был консистентным необходимо:

- 1) использовать на всех страницах одну и ту же цветовую схему;
- 2) одинаково располагать навигации на разных страницах;
- 3) заголовки оформлять одинакового размера;
- 4) следить за тем, чтобы расстояния по вертикали и горизонтали между элементами макета везде были одинаковыми;
- 5) придерживаться единого оформления ссылок и форм;
- б) использовать иконки, выполненные в одном стиле;

в) Ресурс не оптимизирован под мобильные устройства. Адаптивный дизайн обеспечивает качественное отображение сайта на всех устройствах, что позволяет пользователям различных девайсов без проблем взаимодействовать с ресурсом.

г) Неудачная цветовая схема. Цвета, не подходящие под тематику, повлияют на восприятие сайта. Сливающиеся цвета отталкивают пользователей. В таком случае рекомендуется использовать 3 разных цвета с соотношением 60%, 30% и 10%. Если будут изображения и фотографии, которые требуется использовать, то понадобится выбрать такие цвета, что будут хорошо сочетаться с ними.

д) Некорректное отображение на некоторых разрешениях. Чтобы дизайн выглядел хорошо на всех разрешениях, необходимо самые важные блоки размещать в верхней части страницы, убедиться в удобной читабельности текста в разных колонках, также проверить, все ли элементы страницы находятся на нужных местах и ничего не смещается.

е) Безупречные фото. Изображения формируют у целевой аудитории определенное впечатление о бренде, поэтому рекомендуется использовать фотографии с реальными людьми, а не стоковые изображения. Также нужно оптимизировать размеры изображений для их быстрой загрузки.

ж) Отсутствие свободного пространства. Конечно же, основной функцией сайта является предоставление информации пользователю, но ее необходимо подавать таким образом, чтобы между блоками текста было достаточно свободного места. Это поспособствует:

- 1) Улучшить общее восприятие к предоставляемой информации
- 2) Привлечь внимание к тексту окружив его пустым пространством.
- 3) Отсутствие поиска. Возможно, что нужная пользователю информация находится где-то внутри сайта, соответственно необходимо предоставить ему возможность найти ее самостоятельно. К тому же этот элемент тоже можно сделать довольно привлекательным [3].

1.3 Постановка задачи

Задачи, которые ставятся заказчиком при разработке веб-сайта, представляют собой набор конкретных действий, реализация которых приведет к достижению конечной цели. Список задач планируются в зависимости от многих факторов - бюджета проекта, его важности для компании, имеющихся в наличии ресурсов для его реализации и поддержки, ограничений по времени.

Задачи, решаемые в ходе дипломной работы:

- разработка функциональной структуры системы;
- разработка поведенческих диаграмм;
- разработка обобщенного алгоритма функционирования;
- разработка алгоритма по проектированию сайта;
- разработка алгоритма взаимодействия с клиентом;
- описание информационного обеспечения;
- структура программного обеспечения;
- внедрение результата.

Задачи, решаемые компанией в соответствии с потребностями и пожеланиями клиента:

- сбор информации о целевой аудитории и их целях;
- определение технического задания;
- разработка структурной схемы проекта web – сайта;
- выбор методологии проекта и ее адаптирование;
- разработка оригинального дизайна;
- создание удобных и полезных сервисов для целевой аудитории;
- создание коммуникативных сервисов (комментарии, форумы, эл. почта);
- создание маркетинговых инструментов (обратная связь, опрос);
- обеспечение онлайн-услуг для клиентов.

2 Структура, диаграммы и алгоритмы функционирования системы

2.1 Функциональная структура системы

Функциональная структура системы представляет из себя взаимодействие двух модулей – клиента и веб-дизайнера. Клиент заполняет форму требований, передает веб-дизайнеру. Далее по четко установленным и согласованным требованиям веб-дизайнер начинает разрабатывать сайт.

Функциональная структура системы предоставлена на рисунке 2.1

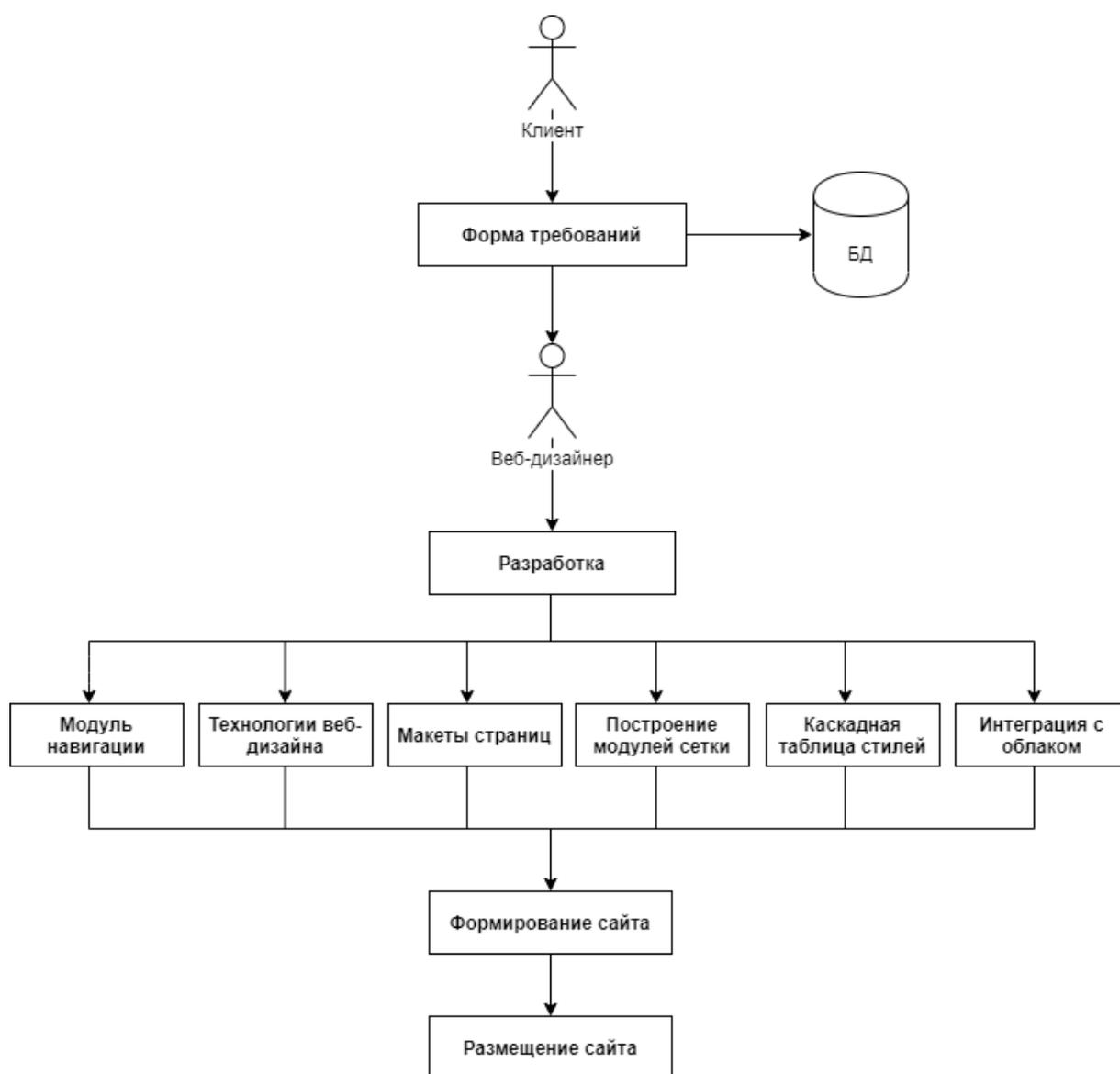


Рисунок 2.1 – Функциональная структура системы

2.2 Диаграмма вариантов использования

Диаграммы вариантов использования моделируют функциональные возможности системы с использованием действующих лиц и вариантов использования.

Варианты использования - это набор действий, сервисов и функций, которые должна выполнять система. «Действующие лица» - это люди или организации, действующие под определенными ролями в системе.

Ниже приведены диаграммы вариантов использования, на которой выделены следующие действующие лица:

- «Пользователь» использует систему для просмотра сайта с целью получения информации об услугах, для возможности обратной связи, отправки заявки, также сможет зарегистрироваться, чтобы иметь доступ к демо-версии шаблонов и быть пользователем форума.

- «Администратор» использует систему для входа в администраторскую панель, имея права на внесение каких-либо изменений, также для обработки заявок.

Диаграмма вариантов использования приведена на рисунке 2.2

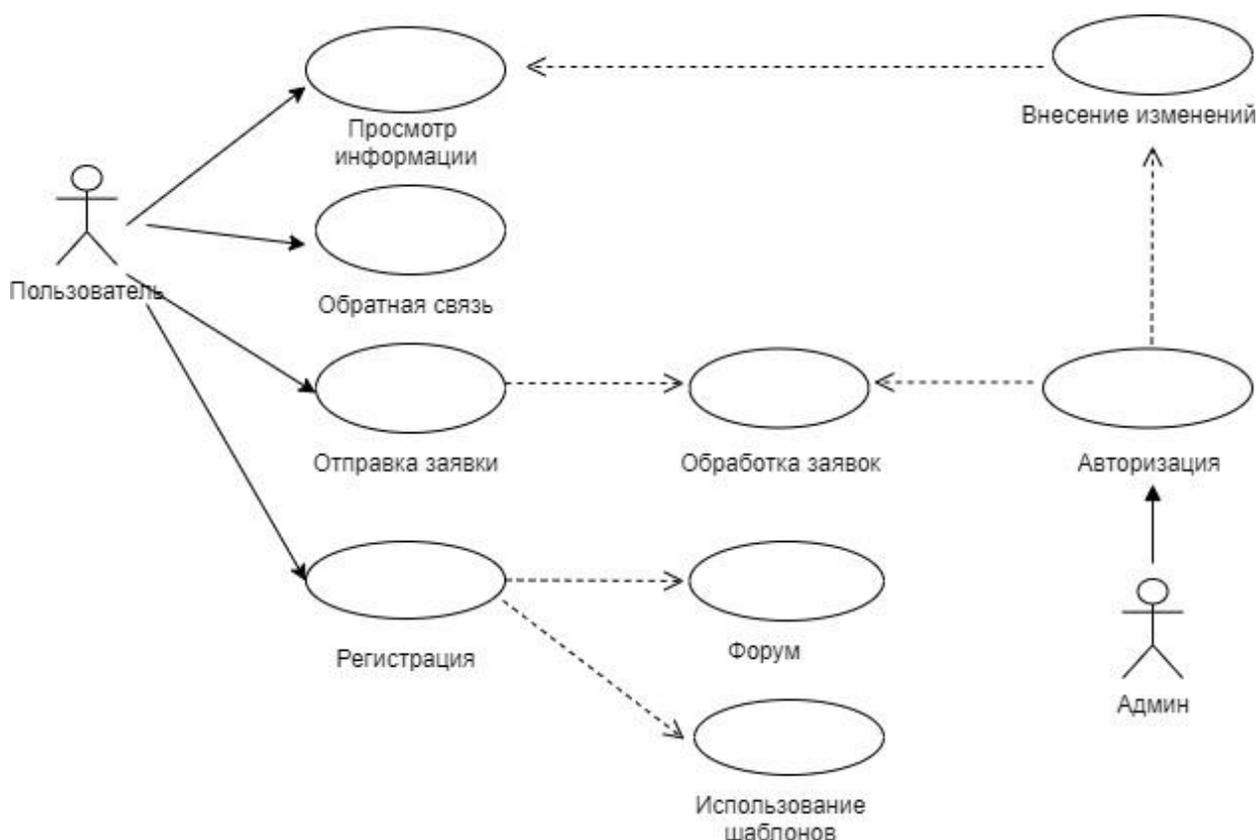


Рисунок 2.2 – Диаграмма вариантов использования

2.3 Диаграмма состояний

Диаграмма состояний описывает поведение отдельного объекта в ответ на серию событий в системе. Предназначением данной диаграммы является описание возможных последовательностей состояний и переходов, которые в целом характеризуют поведение объекта в течение его жизненного цикла [4].

Диаграмма состояний, связанных с действиями администратора, представлена на рисунке 2.3.

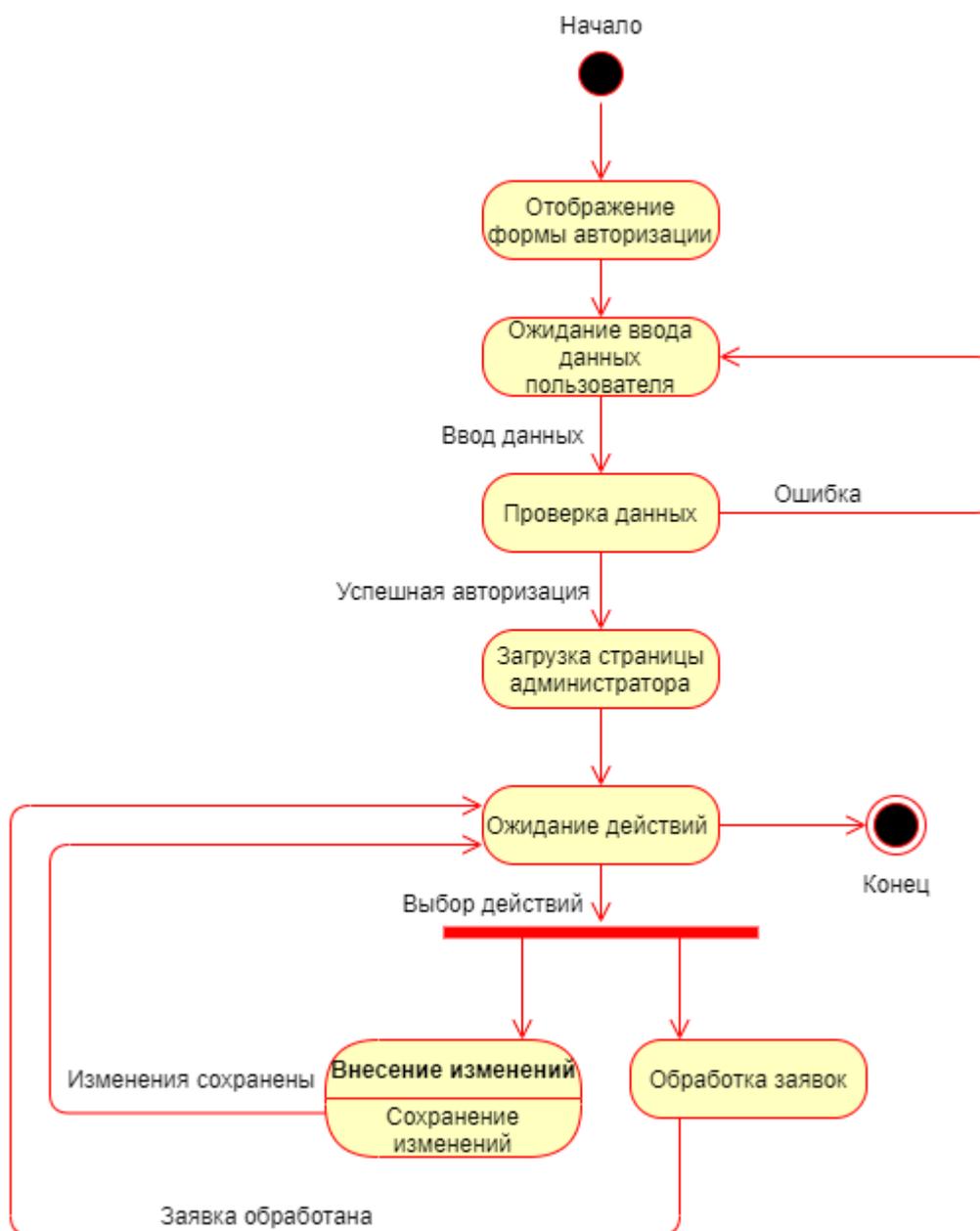


Рисунок 2.3 – диаграмма состояний

Диаграмма состояний, связанных с действиями пользователя представлена на рисунке 2.4.

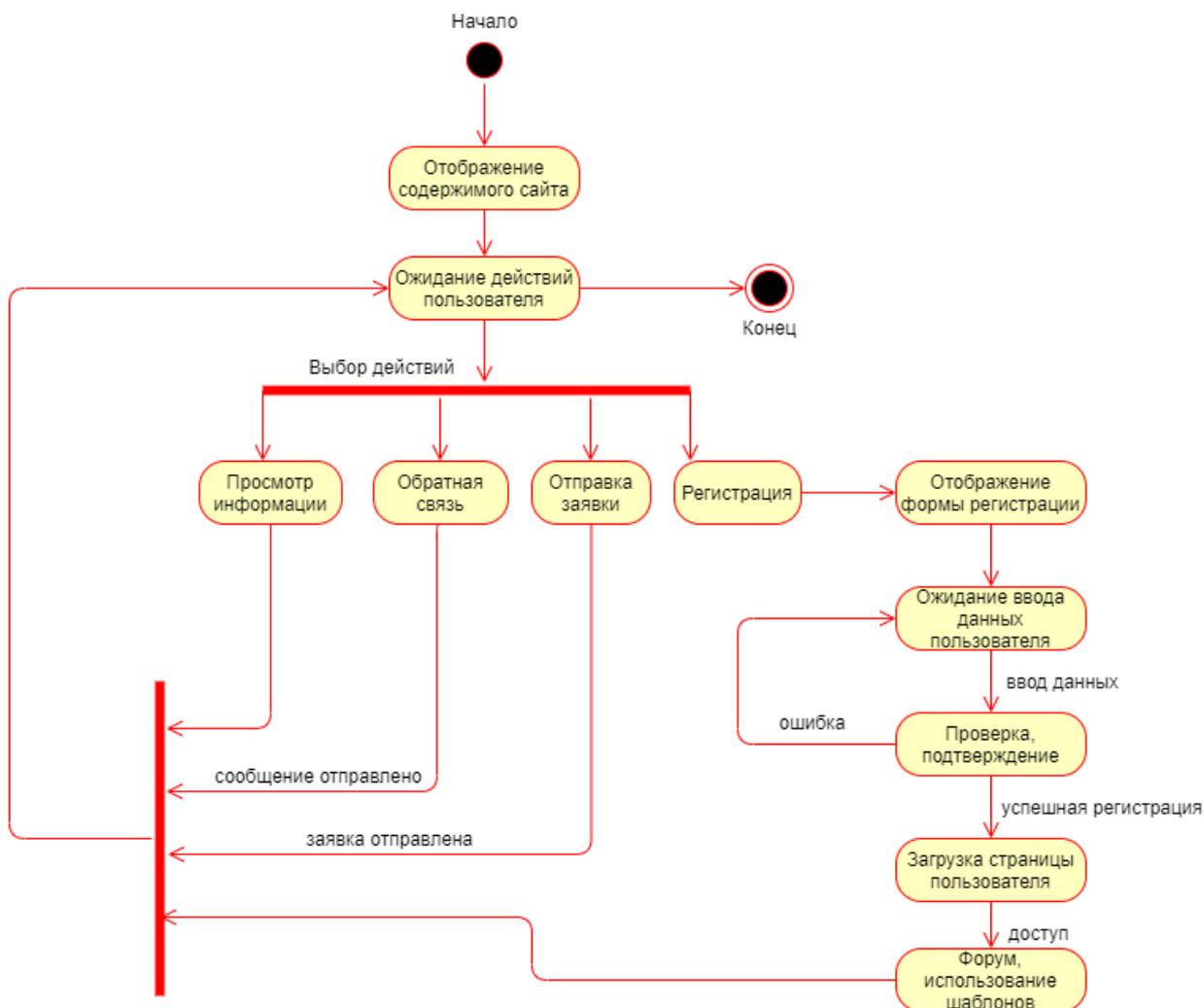


Рисунок 2.4 – диаграмма состояний

2.4 Обобщенный алгоритм функционирования

Посещая сайт, пользователь имеет доступ к информации компании, услугам, обратной связи с компанией, также может отправить заявку на получение какой-либо услуги. На сайте имеются формы авторизации и регистрации. При регистрации пользователь имеет доступ к использованию шаблонов и бесплатного теста, так как компания предоставляет демо-версию любого интересующего шаблона. Также зарегистрированные пользователи могут активно принимать участие в форуме, где можно будет видеть задачи по разным категориям и их решения, предлагать свои методы решения [Приложение Б].

В случае авторизации пользователя определяется роль, исходя из которой осуществляются дальнейшие действия и возможности на данном сайте. Если

авторизацию производит админ, то система дает доступ к странице с возможностями обработки заявок клиентов, внесения изменений на сайте касательно услуг, информации, контактов компании. Если же авторизация происходит со стороны пользователя, то система переводит его на страницу с соответствующими возможностями [Приложение В].

Обобщенный алгоритм функционирования представлен на рисунке 2.5

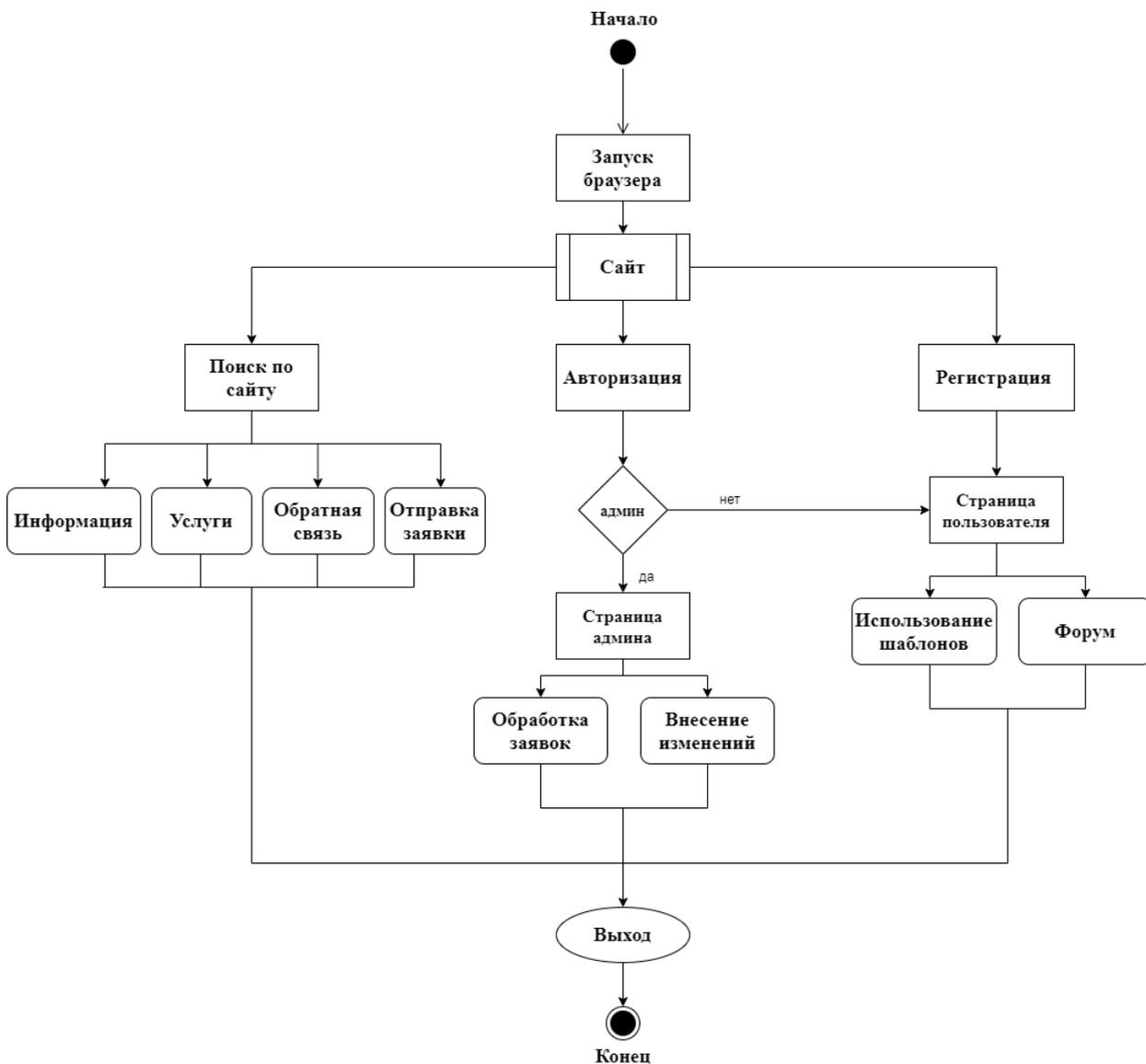


Рисунок 2.5 – обобщенный алгоритм функционирования системы

2.5 Алгоритм проектирования сайта

Несмотря на общепринятые взгляды, основная часть разработки и дизайна веб-сайтов не является необходимой для процесса кодирования. На самом деле, технологии как HTML, CSS и JavaScript придают веб-сайту форму и определяют

способ взаимодействия с информацией. Но то, что остается важнейшей частью жизненного цикла проектирования веб-сайтов, - это этапы предварительного сбора информации, детального планирования и обслуживания после запуска.

В данной главе дипломной работы пошагово описан процесс разработки сайта:

- Первый шаг – проведение исследований, которые определяют, как будут выглядеть последующие шаги. Задачей на данный момент является четкое понимание целей будущего веб-сайта и целевой аудитории [5]. Благодаря тщательно описанному и подробному плану можно избежать затрат дополнительных ресурсов на решение непредвиденных проблем, таких как изменение дизайна или добавление функциональности, которая изначально не планировалась.

- Второй шаг – создание карты сайта. На этом этапе цикла разработки сайта разработчик создает данные, которые позволяют клиенту судить, как будет выглядеть весь сайт. Карта сайта должна описывать отношения между основными областями сайта. Основной причиной создания карты сайта является создание удобного и простого в использовании веб-сайта. На этом шаге необходимо получить от клиента подтверждение, что все выглядит хорошо, и потом переходить к следующему этапу разработки. В этом случае создается каркас или макет. Он описывает элементы, которые будут добавлены на страницу и их расположение.

- На третьем шаге проектирования создается дизайн. Макет сайта является результатом работы дизайнера. Основная функция макета заключается в представлении информационной структуры, визуализации содержимого и демонстрации основных функций. На данном этапе также необходимо получить от клиента подтверждение. Он может просмотреть макет и прислать свой отзыв. Если клиенту не понравятся некоторые аспекты дизайна, нужно будет изменить макет и отправить его обратно ему. Этот цикл следует повторять до тех пор, пока клиент не будет полностью удовлетворен.

- Четвертый шаг – написание контента, которое включает создание заголовков, редактирование текста, написание нового текста [Приложение А].

- Пятый шаг подразумевает создание самого сайта. Обычно сначала создается домашняя страница, а затем добавляются подстраницы в соответствии с иерархией веб-сайта, которая была создана в форме карты сайта. Фреймворки и CMS должны быть реализованы, чтобы сервер мог без проблем справиться с установкой и настройкой.

- На шестом шаге проводится тестирование и запуск. Должны быть проверены каждая ссылка, форма, скрипт. После тщательной проверки необходимо загрузить сайт на сервер.

- Седьмой шаг – техническое обслуживание. Помимо предоставления сайта пользователю, необходимо также убедиться, что все работает нормально, в ином случае быть наготове внести изменения. Система обратной связи, добавленная на сайт, позволит обнаружить возможные проблемы, с которыми

сталкиваются конечные пользователи. Также немаловажной задачей является поддержание актуального состояния сайта.

Контрольный алгоритм проектирования сайта:

Шаг 0. Начало;

Шаг 1. Сбор информации:

- основные цели и целевая аудитория;

Шаг 2. Планирование:

- карта сайта;

- каркас;

Шаг 3. Дизайн:

- макет страниц;

- цикл просмотра;

- утверждение измененного макета; Шаг 4. Написание контента;

Шаг 5. Кодирование;

Шаг 6. Тестирование и запуск;

Шаг 7. Техническое обслуживание:

- система обратной связи;

- устранение проблем;

- поддержание сайта в актуальном состоянии;

Шаг 8. Конец.

2.6 Алгоритм взаимодействия с клиентом

После того как клиент ознакомился с услугами компании, он может подать заявку на получение той или иной услуги: заказать веб-сайт, купить или протестировать готовый шаблон, предоставляемый сайтом. Заполнив нужную форму для совершения заказа, и отправив запрос компании, в течение нескольких секунд на почту клиента придет другая форма, где необходимо будет описать свои требования и пожелания к заказу. Далее проводится уточнение требований клиента, если требования не удовлетворены – производится их корректировка. Если же требования удовлетворены – начинается формирование сайта.

Алгоритм взаимодействия с клиентом:

Шаг 0. Начало;

Шаг 1. Подача заявки;

Шаг 2. Отправка формы на эл. почту клиента; Шаг 3. Получение требований;

Шаг 4. Уточнение требований;

Шаг 5. Условие. Если требования не удовлетворены – возврат на шаг 4;

Шаг 6. Формирование сайта;

Шаг 7. Конец.

3 Информационное и программное обеспечение системы

3.1 Описание информационного обеспечения

Информационное обеспечение включает в себя защиту целостности, доступность, достоверность, безотказность и конфиденциальность пользовательских данных [6].

В информационной базе данных имеются такие таблицы как: «Request», «Clients», «Forum», «Patterns», «Registered users» и «Contacts».

Таблица «Request» хранит информацию о запросах клиента. Таблица «Clients» хранит информацию о клиентах.

Таблица «Forum» содержит информацию о вопросах и ответах пользователей.

Таблица «Patterns» содержит информацию о шаблонах.

Таблица «Registered users» содержит информацию о зарегистрированных пользователях.

Таблица «Contacts» хранит информацию о контактах компании.

Структуры данных таблиц приведены ниже.

Таблица 3.1 - «Request»

поле	описание
Id	Идентификатор запроса
Id_cl	Идентификатор клиента
Email_cl	Почта клиента
Phonenum_cl	Номер клиента
Date	Дата запроса
Status	Статус
description	Описание

Таблица 3.2 - «Clients»

поле	описание
Id	Идентификатор клиента
Name	Имя
Email	Почта
phonenum	Номер

Таблица 3.3 - «Forum»

поле	описание
Id_cl	Идентификатор клиента
Question	Вопрос
Answer	Ответ
Date_ques	Дата добавления вопроса
Date_answ	Дата добавления ответа

Email_reg	Логин пользователя
Role_reg	Роль пользователя

Таблица 3.4 - «Patterns»

поле	описание
Id	Идентификатор шаблона
Id_reg	Идентификатор пользователя
date	Дата добавления

Таблица 3.5 – «Registered users»

поле	описание
Id	Идентификатор пользователя
name	Имя
email	Логин
Password	Пароль
role	Роль

Таблица 3.6 - «Contacts»

поле	описание
email	Почта компании
Phone_number	Номер
address	Адрес

Данные представляют набор взаимосвязанных таблиц, которые организуют физическую модель баз данных представленную на рисунке 3.1

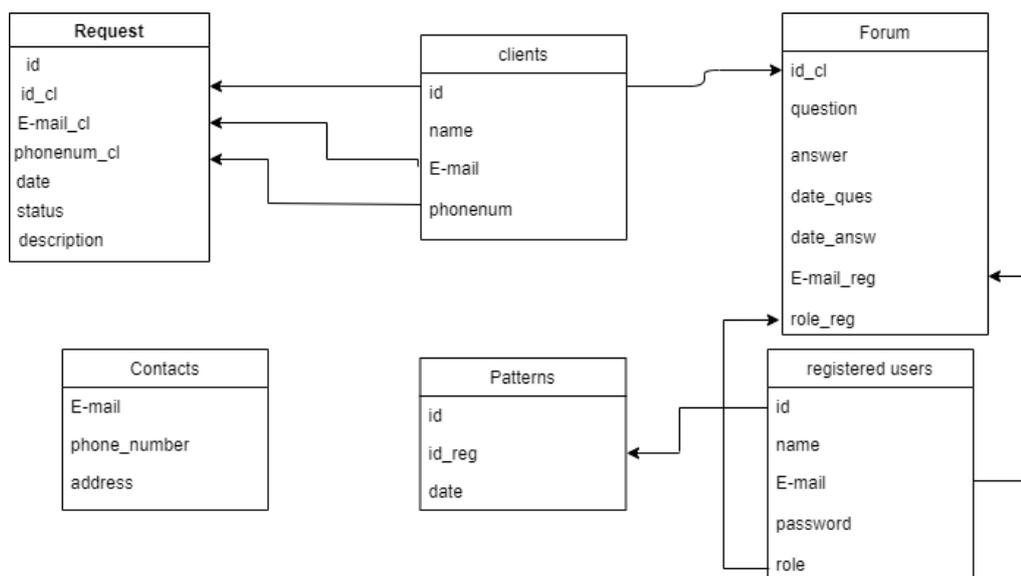


Рисунок 3.1 – физическая модель баз данных

3.2 Структура программного обеспечения системы

Программное обеспечение (ПО) - неотъемлемая часть компьютерной системы, совокупность программ, которые являются логическим продолжением технических средств и управляют работой компьютера и выполняют различные вспомогательные функции.

Структура программного обеспечения подразделяется на три части: системное ПО, инструментальное ПО и прикладное ПО. Структура программного обеспечения представлена на рисунке 3.2.

Системное ПО. В данной дипломной работе для разработки сайта использовалась операционная система Windows 10.

Инструментальное ПО. В качестве инструментального программного обеспечения были использованы язык программирования PHP7, фреймворк Laravel для сборки бэкэнда, фреймворк Vue.js для сборки фронтэнда. Также применялись инструменты laravel mix, sass, axios, bootstrap.

Прикладное ПО. В качестве прикладного программного обеспечения выступают интегрированная среда разработки PhpStorm, распределенная система контроля версий git, MySQL Workbench.

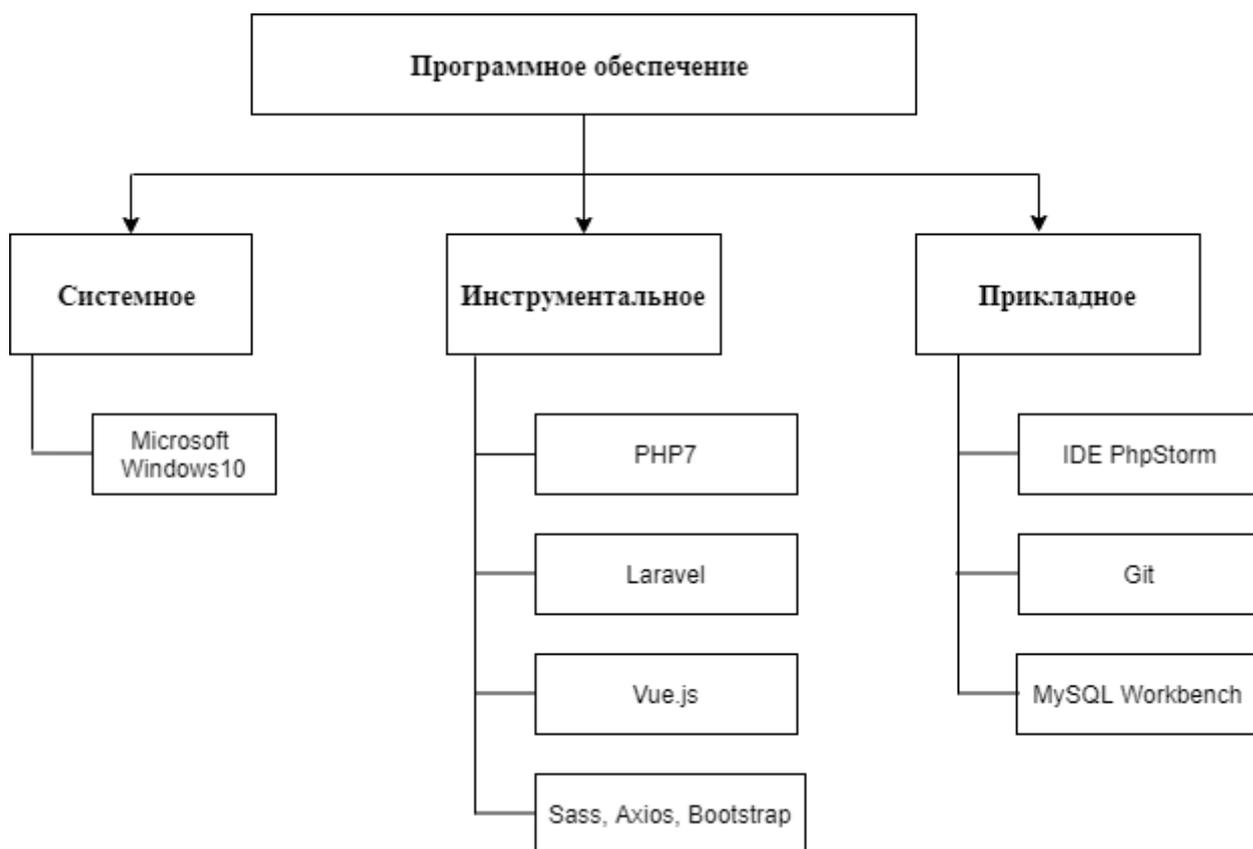


Рисунок 3.2 – Структура программного обеспечения

3.2.1 Системное программное обеспечение

Центральное место среди системных программ занимают операционные системы. ОС выступает в качестве всеобъемлющей структуры, с которой все приложения могут взаимодействовать согласованным образом, значительно упрощая весь процесс разработки. Операционная система представляет собой программное обеспечение, которое позволяет запускать приложения на вычислительном устройстве. ОС управляет всем аппаратным и программным обеспечением на компьютере. Операционные системы включают в себя системные службы, библиотеки и интерфейсы прикладного программирования[7].

При проектировании веб-сайта использовалась операционная система Windows 10 и сервисные программы. Microsoft выпустила Windows 10 в июле 2015 года в качестве дополнения к Windows 8.

Windows 10 обладает встроенными возможностями, которые позволяют корпоративным ИТ-отделам использовать программное обеспечение управления мобильными устройствами для защиты и управления устройствами под управлением операционной системы. Кроме того, организации могут использовать традиционные программы управления настольными системами, такие как Microsoft System Center Configuration Manager. Microsoft Windows 10 интегрировала поддержку технологий многофакторной аутентификации, таких как смарт-карты и токены. Операционная система также включает средства обеспечения безопасности на основе виртуализации, такие как изолированный пользовательский режим, Защитник устройств Windows Defender и Защитник учетных данных Windows Defender. Эти функции Windows 10 хранят данные, процессы и учетные данные пользователя изолированными в попытке ограничить ущерб от любых атак. Windows 10 также расширила поддержку шифрования BitLocker для защиты данных в движении между пользовательскими устройствами, оборудованием хранения, электронной почтой и облачными сервисами. Microsoft объявила о выпуске Windows 10 в сентябре 2014 года, а в следующем месяце представила техническую предварительную версию ОС для избранной группы пользователей под названием Windows Insiders. Microsoft выпустила Windows 10 для широкой публики в июле 2015 года. В целом, пользователи и ИТ-эксперты считают, что Windows 10 гораздо более удобна для предприятий, чем Windows 8 из-за ее более традиционного интерфейса, который повторяет удобную для Windows компоновку Windows 7.

3.2.2 Инструментальное программное обеспечение

Для реализации сайта был выбран язык программирования PHP7, применялись фреймворки laravel и vuejs, и инструмент laravel mix.

PHP является наиболее распространенным и популярным языком программирования на стороне сервера в сети. Это открытый, объектно-ориентированный язык сценариев. Он устанавливается большинством веб-

хостов, имеет тесные связи с базой данных MySQL и коллекцию библиотек для сокращения времени разработки. Разработчики предпочитают PHP из-за его превосходного времени отклика и более низких затрат на хостинг.

Преимущества PHP:

- Представление. PHP-скрипт выполняется намного быстрее, чем те, которые написаны на других языках, таких как JSP и ASP. PHP использует собственную память, поэтому нагрузка на сервер и время загрузки автоматически сокращаются, что приводит к более высокой скорости обработки и повышению производительности.

- Открытый исходный код. Исходный код PHP и программное обеспечение свободно доступны в Интернете. Можно разрабатывать все версии PHP в соответствии с требованиями без каких-либо затрат. Все его компоненты бесплатны для скачивания и использования.

- Синтаксис. PHP имеет легко понятный синтаксис.

- Независимая платформа. PHP доступен для операционных систем Windows, Mac, Linux и Unix. Приложение PHP, разработанное в одной операционной системе, может быть легко выполнено в другой;

- Поддержка базы данных. PHP поддерживает все ведущие базы данных, такие как MySQL, SQLite, ODBC.

- Отчет об ошибках. PHP имеет predefined константы сообщения об ошибках, чтобы генерировать уведомление об ошибке или предупреждение во время выполнения.

- Свободно типизированный язык. PHP позволяет использовать переменную, не объявляя ее тип данных. Он будет взят автоматически во время выполнения в зависимости от типа данных, которые он содержит в своем значении.

- Поддержка веб-серверов. PHP совместим практически со всеми локальными серверами, такими как Apache, Netscape.

- Безопасность. PHP - это безопасный язык для разработки сайта. Он состоит из нескольких уровней безопасности для предотвращения потоков и вредоносных атак.

- Контроль. Для разных языков программирования требуется длинный скрипт или код, тогда как PHP может выполнять одну и ту же работу в несколько строк кода. Он имеет максимальный контроль над веб-сайтами.

Фреймворк - платформа для разработки программных приложений. Фреймворк похож на интерфейс прикладного программирования, хотя технически фреймворк включает в себя API. Платформа служит основой для программирования, в то время как API обеспечивает доступ к элементам, которые поддерживаются платформой. Каркас может также включать библиотеки кода, компилятор и другие программы, используемые в процессе разработки программного обеспечения.

Laravel - самый популярный PHP-фреймворк. Laravel - это полнофункциональная платформа веб-разработки, которая предоставляет возможности для работы с крупномасштабными веб-приложениями.

Благодаря возможности организовывать API и код веб-сайта в хорошо разделенные папки, Laravel помогает лучше организовать код. Laravel предпочтительнее других платформ из-за предложения полного решения MVC, также он поддерживается хранилищем промежуточного программного обеспечения, что очень помогает сделать задачи, вроде авторизации, простыми. Платформы JavaScript (JS) являются одной из наиболее предпочтительных платформ для создания динамичных современных приложений, чатов в реальном времени, электронной коммерции, инвентаризации, обработки и многого другого. Vue.js была выпущена в 2014 году. Данная платформа является самой быстрорастущей платформой, легким прогрессивным JS-фреймворком, в основе которого лежат многие концепции ReactJS и AngularJS. Метод моделирования в Vue – использование тегов стиля в однофайловых компонентах. Однофайловые компоненты дают полный доступ к CSS в том же файле, что и другой код компонента.

Сборка фронтенда осуществлялась с помощью встроенного laravel mix, также на фронтенде использовались такие вспомогательные инструменты как:

- Sass. Sass — язык расширений CSS, предназначен для увеличения уровня абстракции CSS кода и упрощения файлов каскадных таблиц стилей. Sass полностью совместим со всеми версиями CSS.

- Axios. Axios представляет собой HTTP-клиент, основанный на промисах и предназначенный для браузеров[8]. Axios работает во всех современных веб-браузерах. Axios можно использовать для выполнения простого GET запроса. Axios также может быть использован для выполнения простого POST запроса.

- Bootstrap. Bootstrap - это бесплатный интерфейс для быстрой и простой веб-разработки, включающий шаблоны дизайна на основе HTML и CSS для типографики, форм, кнопок, таблиц, навигации, модальных изображений, каруселей изображений и многих других, а также дополнительные плагины JavaScript. Bootstrap также дает вам возможность легко создавать адаптивные дизайны. Преимущества Bootstrap в том, что он прост в использовании, адаптивный CSS Bootstrap подстраивается под любые девайсы (смартфоны, планшеты, настольные компьютеры). Также Bootstrap имеет совместимость со всеми современными браузерами (Chrome, Firefox, Internet Explorer 10+, Safari и Opera).

3.2.3 Прикладное программное обеспечение

Для разработки веб-сайта были использованы прикладные программы как: PHPStorm, Git и MySQL Workbench.

PhpStorm – многофункциональная интегрированная среда разработки (IDE) для PHP и веб-разработчиков, разработанная JetBrains. Он поддерживает версии начиная с PHP 5.3. PhpStorm работает с большинством фреймворков такими как Laravel, WordPress, Joomla, Symfony и другими. Данная IDE предоставляет инструменты и функции поддержки кода для работы с базами данных, соединяется с базой данных, редактирует таблицы и схемы, выполняет

запросы, анализирует схему с помощью UML-диаграммы. PhpStorm предоставляет унифицированный пользовательский интерфейс для работы с системами контроля версий.

Git – бесплатная распределенная система контроля версий с открытым исходным кодом, предназначен для быстрой и эффективной работы с небольшими и крупными проектами. Для работы с базами данных использовался унифицированный визуальный инструмент MySQL Workbench, предоставляющий моделирование данных, разработку SQL и комплексные инструменты администрирования для настройки сервера, администрирования пользователей, резервного копирования и многого другого[9].

Описание контрольного примера

На рисунке 3.3 представлена главная страница разработанного сайта, содержащий вкладки: «Главная», «Форум», «Услуги», «О нас», «Контакты».

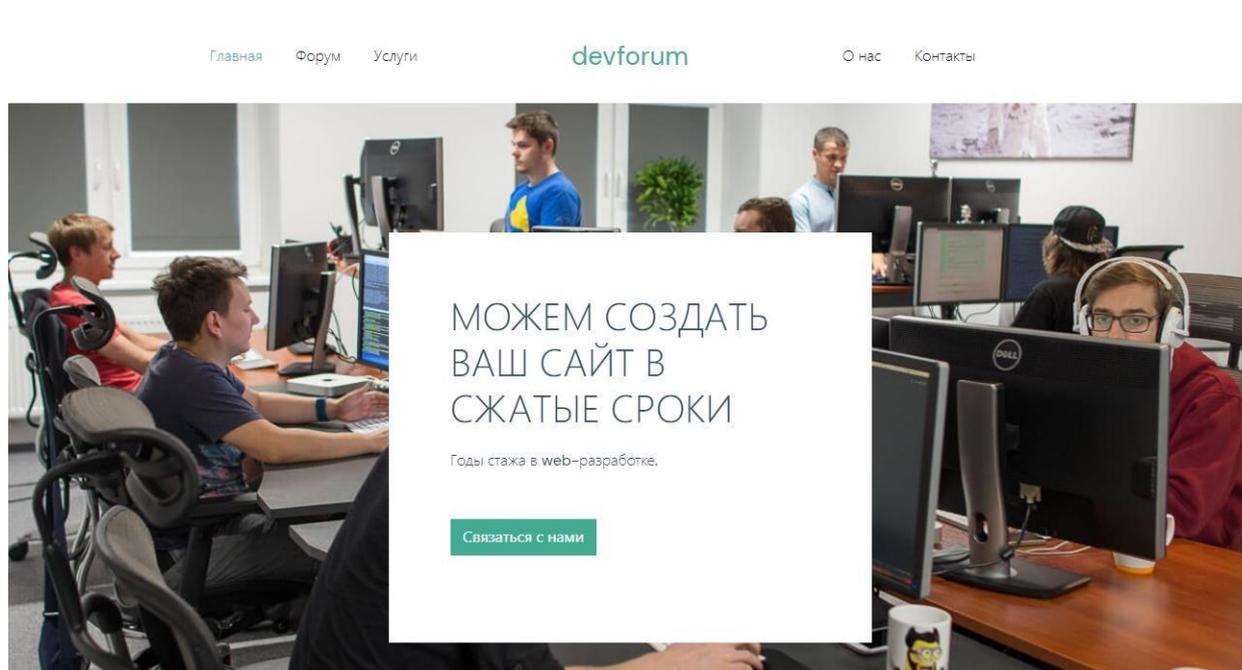


Рисунок 3.3 – Главная страница

Нажав на вкладку «О нас», страница прокрутится до краткой информации о компании, показанной на рисунке 3.4.

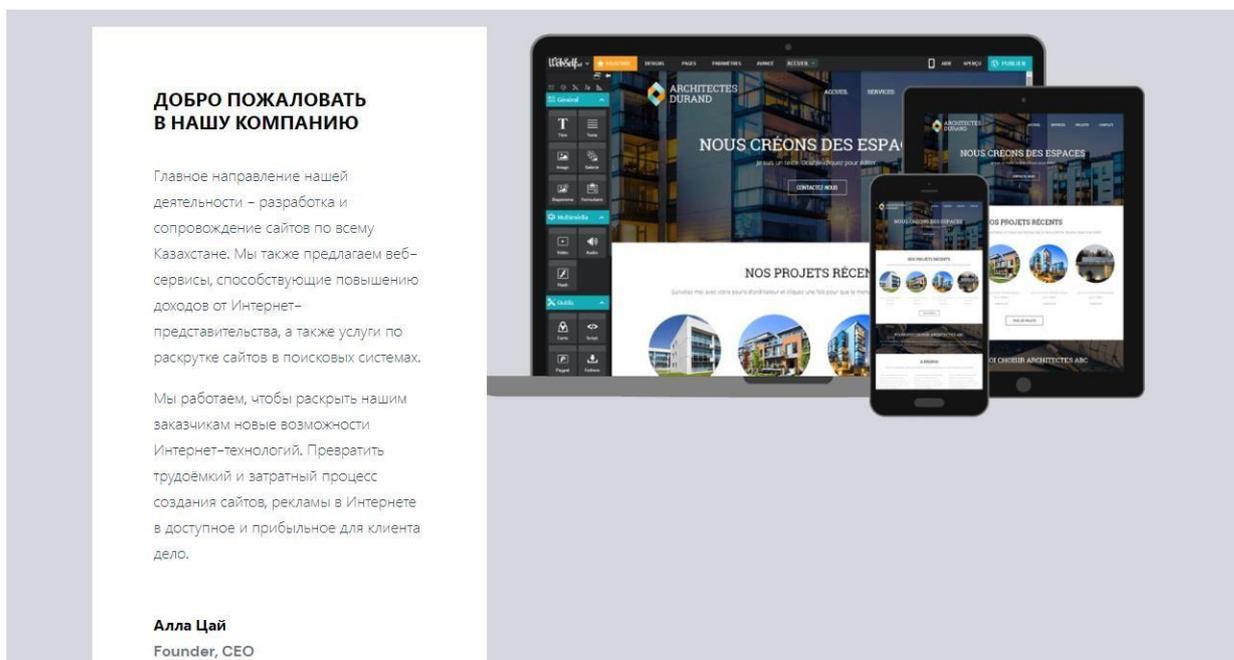


Рисунок 3.4 – Краткая информация о компании

На рисунке 3.5 отображаются услуги компании.

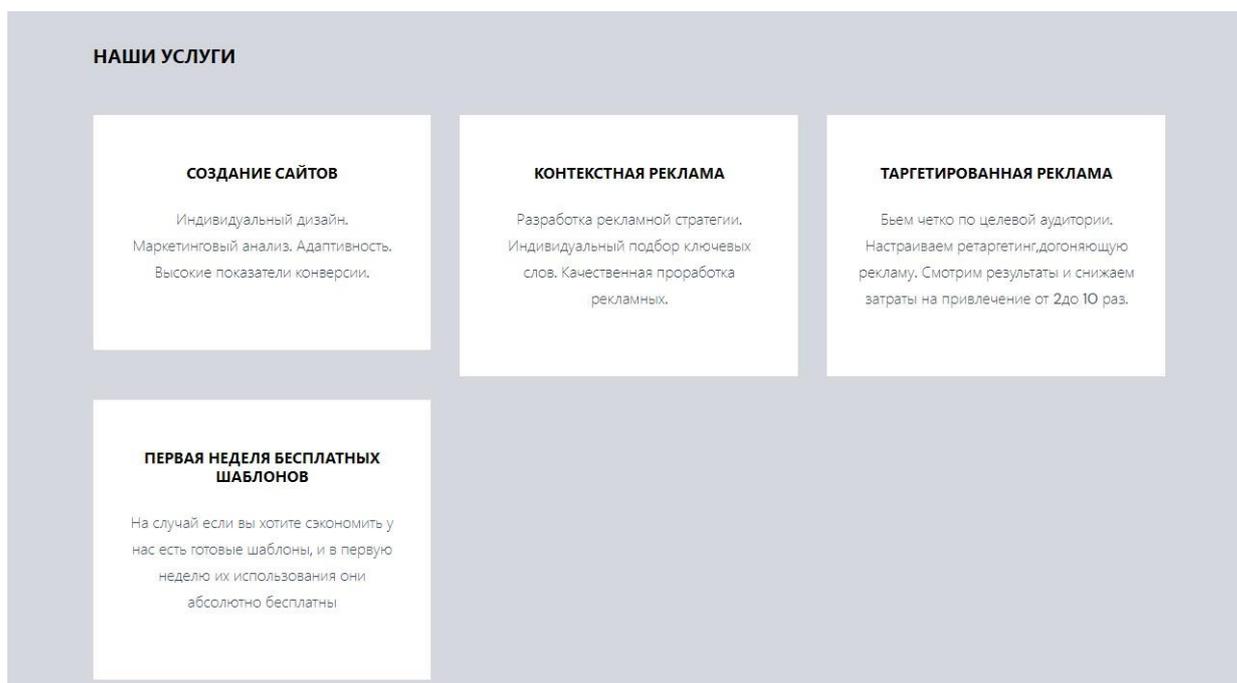


Рисунок 3.5 – Услуги компании

На рисунке 3.6 представляется несколько вопросов, перейдя по которым пользователь попадет на страницу форума.

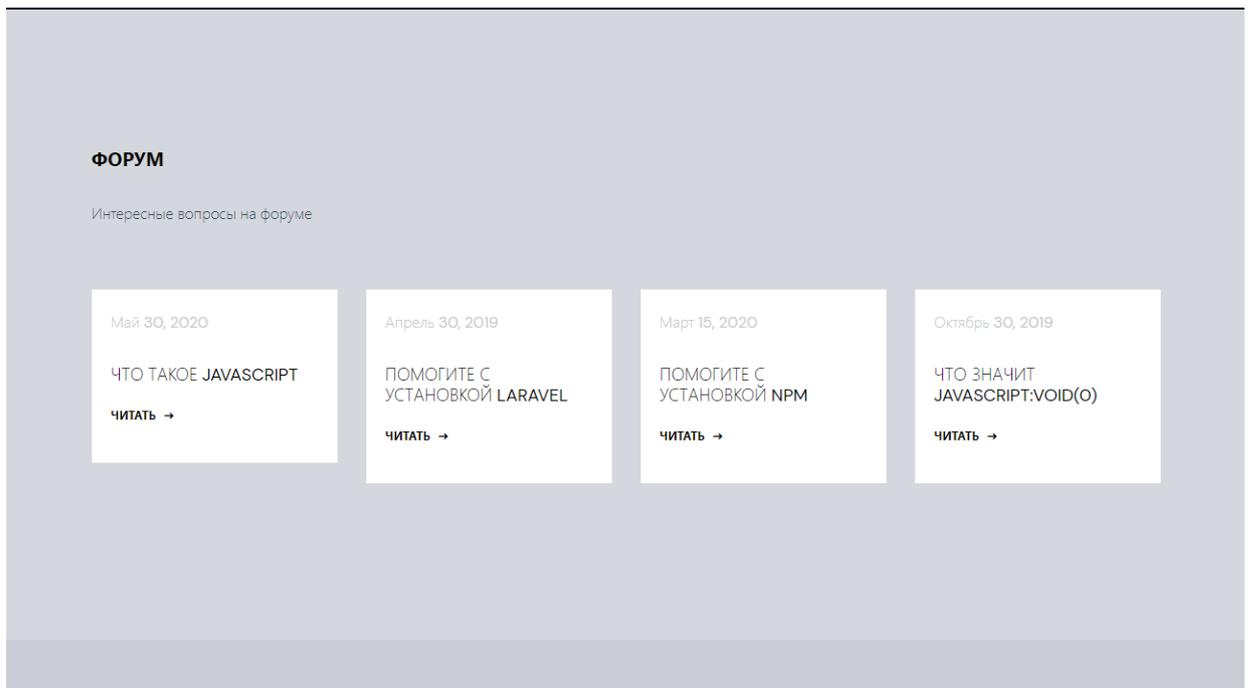


Рисунок 3.6 – Вопросы форума

В конце главной страницы указана форма для обратной связи и социальные сети компании, которые представлены на рисунке 3.7.

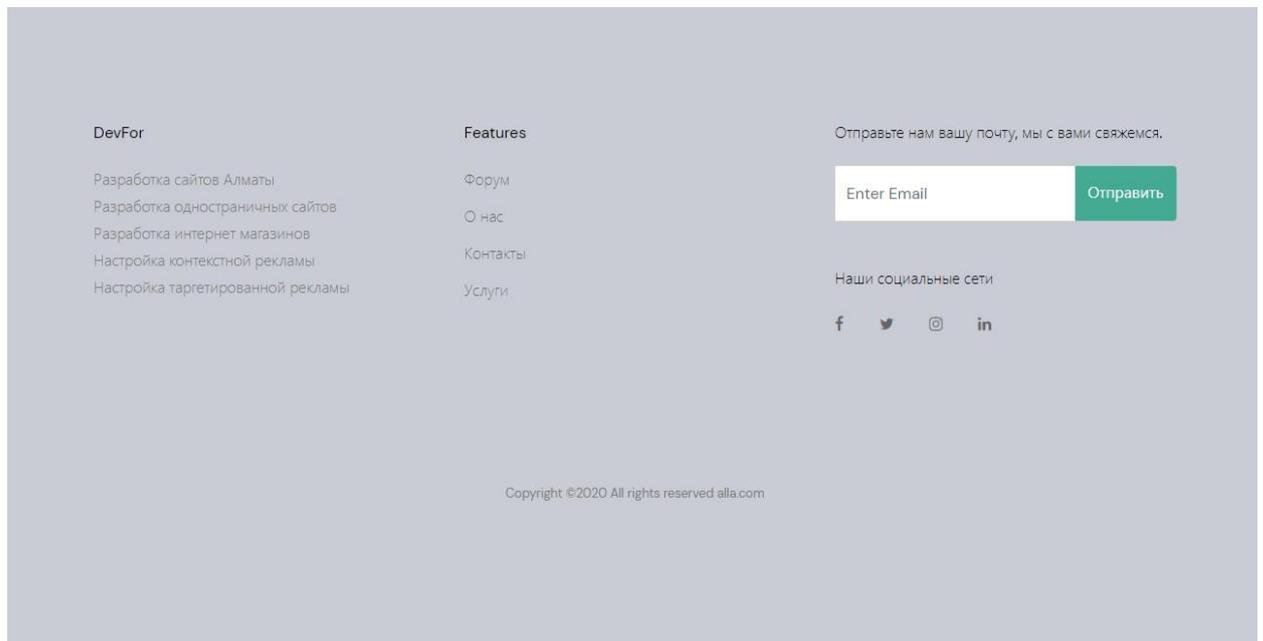


Рисунок 3.7 – Форма обратной связи, социальные сети

На рисунке 3.8 представлена форма регистрации пользователей на форуме.

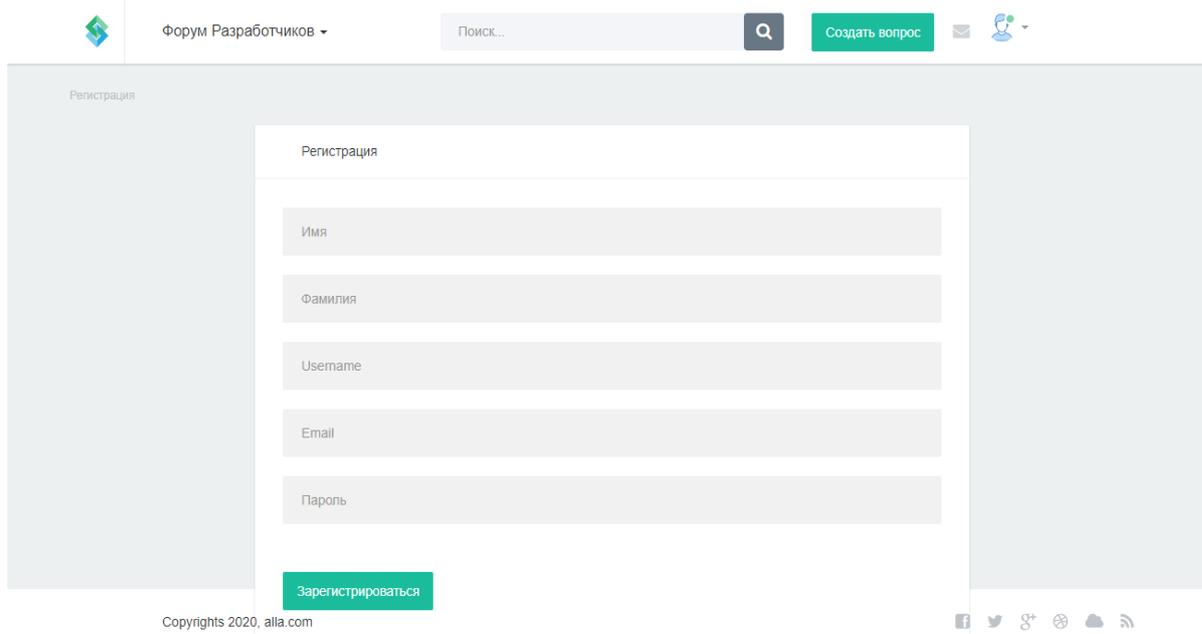


Рисунок 3.8 – форма регистрации пользователей

На рисунке 3.9 представлена форма авторизации пользователя.

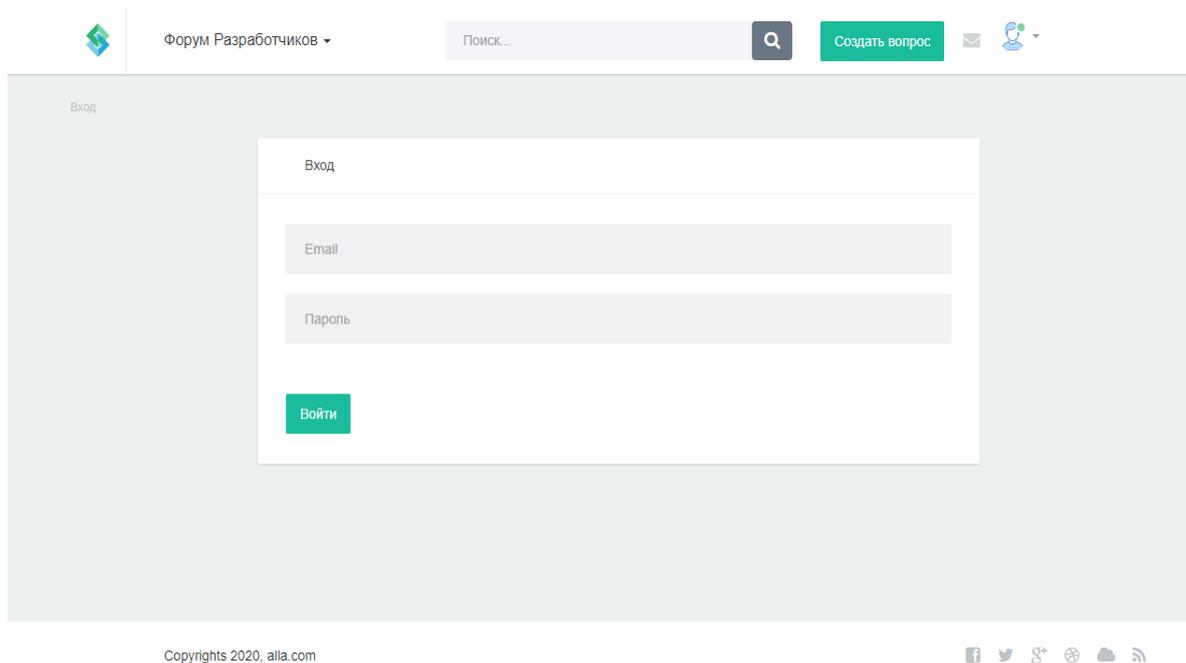


Рисунок 3.9 – Форма авторизации

Пользователь может создать вопрос по любым категориям, представленных справа. Форма для создания вопроса показана на рисунке 4.1.

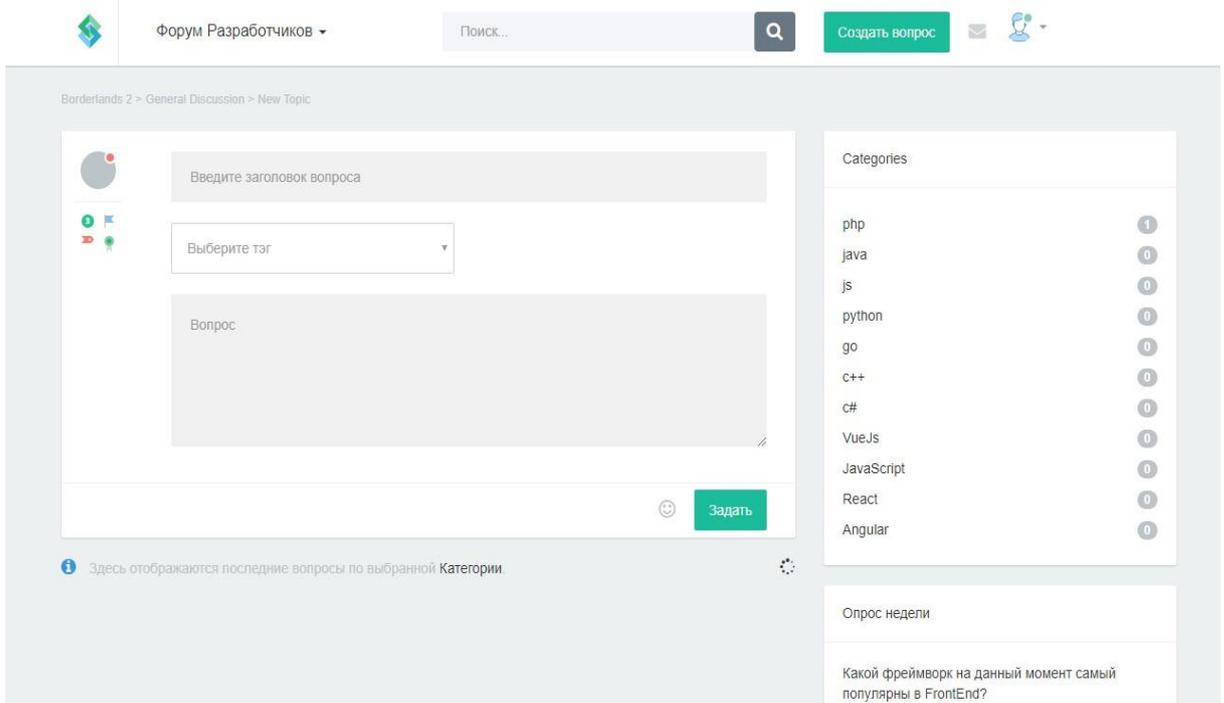


Рисунок 4.1 – Создание вопроса на форуме

На рисунке 4.2 отображается профиль пользователя и его вопросы.

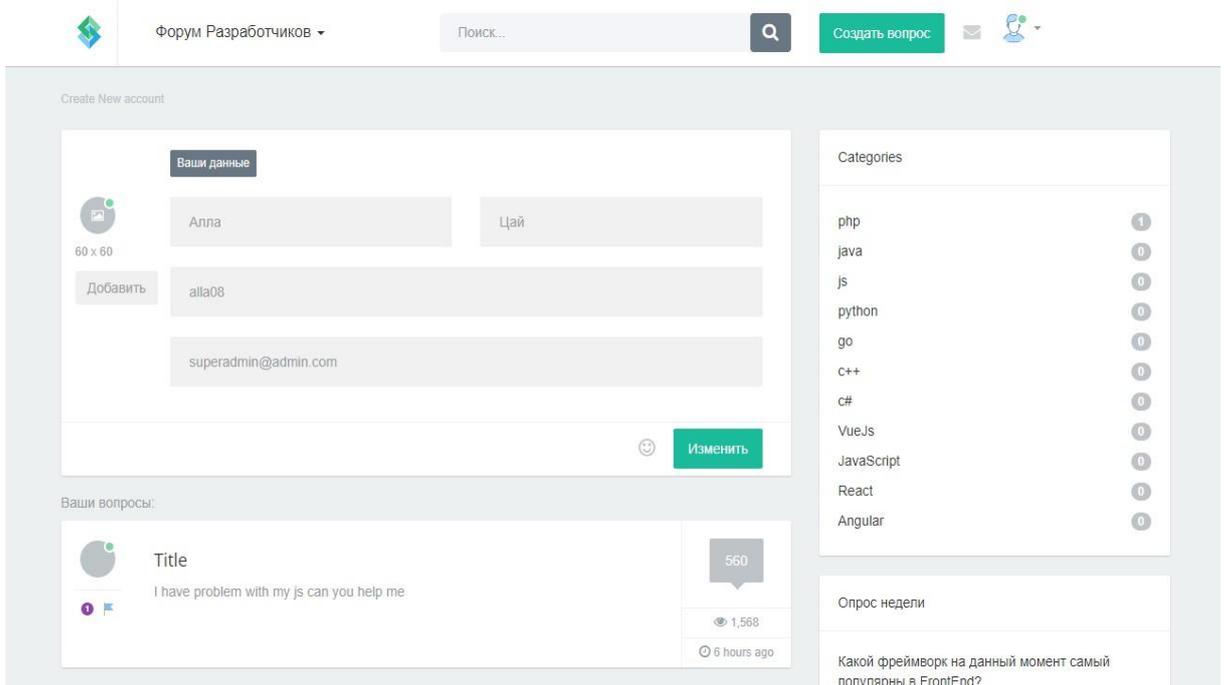


Рисунок 4.2 – Профиль и вопросы пользователя

4 Внедрение системы

4.1 Инфраструктура размещения сайта и его продвижение

Для публикации сайта в сети Интернет необходимо зарегистрировать доменное имя и разместить сайт на хостинге.

В качестве хостинг-провайдера выбран казахстанский провайдер «PS internet company», который также является регистратором доменов с 2006 года.

Сайт будет размещен на хостинге с тарифным планом: 300 тг/мес или 3240тг/год, место на диске – 0.5Гб, платформа – windows, панель управления – Plesk.

Панель управления хостингом Plesk обеспечивает быструю загрузку файлов на хостинг, установку CMS, легкое управление базами данных и резервными копиями, удобное управление почтовыми ящиками.

Сайты на этом хостинге поддерживают протокол IPv6.

После размещения сайта, главной целью и задачей является привлечь как можно больше посетителей. Продвижение сайта будет осуществляться с помощью поисковой оптимизации SEO (Search Engine Optimization).

Различают два вида оптимизации: внутреннюю и внешнюю. Внутренняя оптимизация включает в себя изменения, которые можно внести на сайт, чтобы повысить рейтинг. Например: ориентация на ключевые слова и их внедрение, создание контента, оптимизация скорости страницы. Внешняя оптимизация включает в себя стратегии для улучшения рейтинга, такие как создание ссылок, маркетинг в области СМИ и рекламная рассылка. Конечно же, для достижения наилучшего результата будут использоваться как внутренняя так и внешняя SEO.

4.2 Перспективы развития

В ходе дальнейшего расширения и улучшения по данному проекту будет разрабатываться интеграция сайта компании с социальными сетями для более эффективной поддержки клиентов и коммуникации с ними.

Интеграция с социальными сетями является отличным решением для удобной, а самое главное, быстрой связи клиентов с компанией. Удобство заключается в том, что пользователь будет иметь возможность написать в сообщения группы компании социальной сети, сообщение попадет в раздел Диалоги компании, из этого же раздела пишется ответ и пользователь получает ответ от имени сообщества в той социальной сети, из которой он написал, будь то Вконтакте, Фейсбук или любая другая сеть. Польза данной интеграции в том, что компания никогда не упустит ничего важного, не нужно будет переключаться между вкладками в браузере, все сообщения будут попадать в единое окно диалогов. Пользователи могут выбирать удобный и привычный для себя канал связи. Также сообщения из разных каналов попадают в единую

статистику по диалогам, что помогает правильно оценивать эффективность каждого оператора и коммуникации в целом. Помимо этого клиенты могут настроить интеграцию с Яндекс Диалогами, что позволит им общаться с компанией прямо из результатов поиска, еще до того как они зайдут на сайт.

Следующая перспектива является более масштабной и представляет из себя создание нового инструмента для работы с веб-дизайном на основе существующих, таких как Figma, Sketch и Photoshop.

При выборе приложения для работы важны следующие нюансы:

- Специфика дизайна[10]. Современные сайты в основном содержат блочную структуру, которая удобнее реализуется с помощью простых элементов. В этом случае Sketch и Figma подходят многим дизайнерам. С другой стороны, если макет включает в себя сложную обработку изображений, то без Photoshop не обойтись.

- Технические возможности разработчиков. Часто разработчики работают с форматом .psd и не используют другие программы кроме продуктов Adobe. В таком случае дизайнерам приходится подстраиваться под других специалистов и выбирать нужное приложение, исходя из возможностей разработчиков.

- Удобство пользования. Сложность интерфейса - одна из особенностей работы с продуктами Adobe. Потому как инструменты не предназначены напрямую для проектирования интерфейсов, у дизайнеров возникают сложности при создании макетов, в отличие от специализированных программ Sketch или Figma.

Исходя из этого, предполагается разработать такой инструмент, в котором будет включена как реализация структуры с помощью простых элементов, так и сложная обработка изображений, возможность разработки дизайна без надобности подстраиваться под определенное приложение. Данный продукт разработается для возможности работы на всех видах операционных систем. Также будет предусматриваться работа через браузер и сохранение результатов в облаке. Помимо этого, рассчитывается размещение конечного продукта на сайте компании в качестве демо-версии для пользователей.

Данная стратегия поможет переосмыслить весь подход к созданию адаптивного дизайна и проектированию интерфейсов. Теперь концентрироваться нужно будет не на разработке мобильных сайтов и веб-сайтов, а на разработке гибких систем с набором правил, которые определяют доставку контента и его предоставление.

ЗАКЛЮЧЕНИЕ

В ходе работы был проведен обзор и анализ технологий, разработаны алгоритмы, по которым будут проектироваться сайты, исходя из соответствующих требований клиентов.

В рамках проведенного анализа были выявлены преимущества и недостатки современных технологий используемых веб-дизайнерами при работе с проектированием удобного интерфейса веб-сайта. Требования к самому сайту и дизайну начинаются с отправки формы требований и пожеланий клиента, которые согласовываются и утверждаются вместе с веб-дизайнером, и в дальнейшем реализовываются в течение установленных сроков. Говоря о дизайне веб-сайта, то для оптимального восприятия предпочтительны светлый фон, трехцветное, чистое наполнение, минимализм, если речь идет не о промо-сайтах и сайтах-визитках, а о серьезных корпоративных или информационных ресурсах с частой посещаемостью. Такой тип дизайна мало весит, что значит, быстро грузится и легко адаптируется под мобильную версию.

В настоящей дипломной работе реализованы поставленные задачи, которые являются непосредственно фундаментом для развития сайта, а именно:

- разработка функциональной структуры системы;
- разработка обобщенного алгоритма функционирования;
- разработка алгоритма по проектированию сайта;
- разработка алгоритма взаимодействия с клиентом;
- описание информационного обеспечения;
- структура программного обеспечения.

Также разработан веб-сайт, ориентированный на предоставление различных услуг клиентам с учетом всех пожеланий, используя современные технологические решения.

Подводя итоги, необходимо выделить и вышесказанные перспективы развития, которые включают в себя интеграцию с социальными сетями и разработку новых инструментов для легкой и удобной работы веб-дизайнера.

СПИСОК АББРЕВИАТУР

PTC – Paid To Click PTW – Paid To Work
DDD – Data Driven Design DID – Data Informed Design
Html – Hypertext Markup Language CSS – Cascade Style Sheets
CMS – Content Management System PHP – Hypertext Preprocessor
Sass – Syntactically Awesome Stylesheets Axios – JavaScript библиотека
HTTP – Hypertext Transfer Protocol MVC – Model-View-Controller
CLI – Command Line Interface
API – Application Programming Interface JSP – Java Server Pages
ASP – Active Server Pages

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

- 1 Бизнес в Интернете [Электронный ресурс].URL: <http://ru.wikipedia.org/wiki/>
- 2 Проектирование с использованием данных. Рошель Кинг, Элизабет Черчилль, Кейтлин Тан, 2017 г.
- 3 <https://stfalcon.com/ru/blog/post/10-most-common-web-design-mistakes>
- 4 <https://habr.com/ru/post/74330/>
- 5 Создание сайта в 5 этапов. Рязанов В.И., 2009г.
- 6 https://ru.qwe.wiki/wiki/Information_assurance
- 7 Основы информатики и вычислительной техники (Дистанционное обучение. – Мн.: БГЭУ, 2005). Говядинова Н.Н. [Электронный ресурс] <http://www.bseu.by/it/oivt/tema3v2.htm>
- 8 <https://habr.com/ru/company/ruvds/blog/477286/>
- 9 Git для профессионального программиста. Чакон С., Штрауб Б., Питер, 2017. — 496 с.
- 10 Инструменты и приложения для веб-дизайна [Электронный ресурс] webformymself.com/

Приложение А

Код главной страницы

```
<div data-spy="scroll" data-target=".site-navbar-target" data-offset="300">

<div class="site-wrap" id="home-section">
<div class="site-mobile-menu site-navbar-target">
<div class="site-mobile-menu-header">
<div class="site-mobile-menu-close mt-3">
<span class="icon-close2 js-menu-toggle"></span>
</div>
</div>
<div class="site-mobile-menu-body"></div>
</div>
<header class="site-navbar site-navbar-target bg-white" role="banner">

<div class="container">
<div class="row align-items-center position-relative">

<div class="col-lg-4">
<nav class="site-navigation text-right ml-auto " role="navigation">
<ul class="site-menu main-menu js-clone-nav ml-auto d-none d-lg-block">
<li class="active"><a href="index.html" class="nav-link">Home</a></li>
<li><a href="project.html" class="nav-link">Projects</a></li>
<li><a href="services.html" class="nav-link">Services</a></li>
</ul>
</nav>

</div>
<div class="col-lg-4 text-center">
<div class="site-logo">
<a href="index.html">Marga</a>
</div>

<div class="ml-auto toggle-button d-inline-block d-lg-none"><a href="#"
class="site-menu-toggle py-5 js-menu-toggle text-white"><span class="icon-menu
h3 text-primary"></span></a></div>
</div>
<div class="col-lg-4">
<nav class="site-navigation text-left mr-auto " role="navigation">
```


Приложение Б

Код регистрации

```
<?php
namespace App\Http\Controllers\Auth;

use App\User;
use App\Http\Controllers\Controller; use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Validator;
use Illuminate\Foundation\Auth\RegistersUsers;

class RegisterController extends Controller
{

    use RegistersUsers;
    protected $redirectTo = '/home'; public function __construct()
    {
        $this->middleware('guest');
    }

    protected function validator(array $data)
    {
        return Validator::make($data, [
            'name' => ['required', 'string', 'max:255'],
            'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],
            'password' => ['required', 'string', 'min:6', 'confirmed'],
        ]);
    }

    protected function create(array $data)
    {
        return User::create([ 'name' => $data['name'], 'email' => $data['email'],
            'password' => Hash::make($data['password']),
        ]);
    }
}
```

Приложение В

Код авторизации

```
<?php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller; use App\User;
use Illuminate\Foundation\Auth\AuthenticatesUsers; use
Illuminate\Http\Request;
use Illuminate\Support\Facades\Hash;

class LoginController extends Controller
{

use AuthenticatesUsers; protected $redirectTo = '/home';

public function __construct()
{
$this->middleware('guest')->except('logout');
}

public function login(Request $request)
{
$user = User::where('email', $request['email']->first()); if (empty($user)) {
return response()->json(['message' => 'User doesnt exist'], 400);
}
if (Hash::check($request['password'], $user->password)) {
return response()->json(['api_token' => $user->api_token], 200);
}
return response()->json(['message' => 'Invalid password'], 400);
}
}
```