

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего
образования
«Тольяттинский государственный университет»

Кафедра

«Прикладная математика и информатика»

(наименование)

09.03.03 Прикладная информатика

(код и наименование направления подготовки / специальности)

Разработка программного обеспечения

(направленность (профиль)/специализация)

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(БАКАЛАВРСКАЯ РАБОТА)**

на тему

«Разработка программного обеспечения веб-представительства кадрового
агентства»

Обучающийся

Т. А. Страмцов

(Инициалы Фамилия)

(личная подпись)

Руководитель

д.т.н., доцент, С.В. Мкртычев

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2024

Аннотация

Тема выпускной квалификационной работы: «Разработка программного обеспечения веб-представительства кадрового агентства».

Предмет исследования в данной работе – это программное обеспечение веб-представительства кадрового агентства.

Объектом исследования является веб-представительство кадрового агентства.

Цель работы состоит в том, чтобы создать функциональное веб-приложение для кадрового агентства, которое соответствует требованиям и тенденциям отрасли.

Данная работа состоит из:

- введения;
- трех глав;
- заключения;
- списка используемой литературы.

В первой главе речь идет об обоснование разработки сайта, а также задач, решаемых в процессе разработки.

Во второй главе рассматривается проектирование веб-приложения компании. Разрабатывается логическая модель, а также внешние и внутренние структурные схемы сайта.

В третьей главе представлен процесс разработки веб-представительства, выбор инструментов и их описание, а также описание страниц и программных модулей веб-представительства, тестирование и публикация сайта в Интернете.

Бакалаврская работа представлена на 59 страницах и включает в себя 35 рисунков и 3 таблицы.

Список используемой литературы и используемых источников содержит 25 источников.

Оглавление

Введение.....	4
Глава 1 Постановка задачи на разработку программного обеспечения информационной системы.....	6
1.1 Функциональные и архитектурные требования веб-представительств организаций сферы услуг.....	6
1.2 Разработка требований к веб-представительству кадрового агентства.....	7
1.3 Анализ аналогов веб-представительства кадровых агентств.....	9
Глава 2 Проектирование программного обеспечения веб-представительства кадрового агентства.....	20
2.1 Логическое моделирование веб-представительства.....	20
2.2 Логическое моделирование данных веб-представительства.....	24
2.3 Разработка внутренней структуры веб-представительства.....	26
2.4 Разработка внешней структуры веб-представительства.....	28
2.5 Разработка графического дизайна веб-представительства.....	30
Глава 3 Реализация и тестирование.....	32
3.1 Разработка графического дизайна веб-представительства.....	32
3.2 Выбор технологии реализации проекта веб-представительства ..	36
3.3 Выбор и описание средств работы с базой данных.....	39
3.4 Реализация веб-представительства.....	40
3.5 Тестирование разработанного веб-представительства.....	48
Заключение.....	57
Список используемой литературы и используемых источников.....	58

Введение

В наши дни стремительное развитие информационных технологий неизбежно влияет на многие области жизни людей. Одной из таких областей является поиск работы и сотрудников.

Присутствие кадровых агентств в Интернете становится жизненно важным, поскольку оно не только предоставляет информацию о текущих возможностях в плане трудоустройства, но и упрощает сам процесс поиска работы для специалистов и организует постоянную занятость для населения, что является ключевым для передовых стран [3].

Объектом исследования является веб-представительство кадрового агентства.

Предметом исследования является программное обеспечение веб-представительства кадрового агентства.

Цель работы состоит в том, чтобы создать функциональный веб-сайт для кадрового агентства, который соответствует требованиям и тенденциям отрасли.

Для достижения поставленной цели необходимо решить следующие задачи:

- выполнить постановку задачи на разработку программного обеспечения веб-представительства кадрового агентства;
- спроектировать программное обеспечение веб-представительства кадрового агентства;
- реализовать функциональность и выполнить тестирование программного обеспечения веб-представительства кадрового агентства [11].

Методы исследования: методы и технологии проектирования программного обеспечения [11].

Практическая значимость работы заключается в том, что результаты могут быть использованы кадровым агентством для создания собственного веб-приложения.

Данная работа состоит из введения, трех глав, заключения и списка используемой литературы и используемых источников.

Во введении подтверждена актуальность темы работы, представлены объект, предмет, цель и задачи исследования.

В первой главе представлены обоснование разработки сайта, а также задачи, решаемые в процессе разработки.

Во второй главе представлено проектирование веб-приложения. Разработаны логическая модель, внешние и внутренние структурные схемы.

В третьей главе представлена разработка веб-представительства, выбор и описание программных средств, описание страниц и модулей веб-представительства, а также тестирование и публикация сайта в Интернете.

В заключении описываются результаты выполнения выпускной квалификационной работы.

Бакалаврская работа представлена на 59 страницах и включает в себя 35 рисунков и 3 таблицы.

Список используемой литературы и используемых источников содержит 25 источников.

Глава 1 Постановка задачи на разработку программного обеспечения веб-представительства кадрового агентства

1.1 Функциональные и архитектурные особенности веб-представительств кадровых агентств

Веб-приложения представляют собой архитектуру, которая требует тщательного планирования функциональных потребностей. Следует основательно анализировать каждый компонент, который гарантирует их успешное функционирование.

Пользовательский опыт (UX), одна из фундаментальных особенностей при разработке веб-приложений. Веб-представительство для кадрового агентства должно строго следовать дизайнерским принципам, которые разрабатываются с учетом потребностей конечного пользователя. Это подразумевает создание интуитивно понятной навигации и обеспечение очевидности последовательности пользовательских действий.

Внедрение интерактивных элементов. Всплывающие сообщения или компактные формы на главной странице, значительно упрощают доступ к корпоративным услугам, таким как публикация вакансий и подача резюме, и одновременно повышают активность пользователей.

Предоставление полной информации, которая характеризуется своей краткостью и ясностью, а также сопровождается заголовком, отражающим суть контента. Подобный подход расширяет возможности для обоснованных решений со стороны других пользователей приложения.

Визуальный дизайн. Согласованность в стиле на протяжении всего веб-представительства подчеркивает профессионализм и повышает доверие к компании [1]

Адаптивность веб-приложения для обеспечения доступности на множестве гаджетов, от мониторов персональных компьютеров до мобильных телефонов и планшетов, реализация адаптивного дизайна является ключевым

фактором. Этот подход позволяет веб-приложению без труда настраиваться на разнообразные платформы, предоставляя пользователям беспрепятственный доступ и ознакомление с контентом [10].

Видимость в поисковых сервисах считается ключевым элементом для притягивания естественных пользовательских потоков к интернет-приложениям. Для доминирования в поисковой выдаче и обхода конкурентных организаций, агентства по подбору персонала должны разработать и реализовать передовые процедуры поисковой оптимизации, например применение стратегически подобранных поисковых запросов и составление мета-описания [10].

Прямая коммуникация пользователей со специалистами поддержки клиентов, это напрямую способствует укреплению доверия и решению проблем. Отсюда такие инструменты как интерактивные онлайн-чаты, формы обратной связи с электронной почтой и информационные разделы вроде «Часто задаваемых вопросов» играют важную роль в развитии компаний.

Усовершенствование производительности и безопасности в своих веб-приложениях, путем внедрения аналитических инструментов при разработке, например Sonar, EsLint и husky.

1.2 Разработка требований к веб-представительству кадрового агентства

При организации разработки веб-приложения кадрового агентства, первостепенная задача заключается в формулировании чётких директив, отражающих бизнес-задачи и предпочтения целевого пользователя. Для этого мы используем методологию FURPS+, разделяющую требования к цифровым продуктам на пять основных категорий.

Функциональные требования (Functionality), которые излагают функции системы, задачи, лежащие в её основании, и критически важные для удовлетворения нужд как пользователей, так и компании.

Удобство использования (Usability), описывающее функционирование продукта глазами пользователя и ставящее требования к визуальным аспектам, логике структуры и пользовательскому интерфейсу, оказывающему прямое воздействие на пользовательское восприятие и удовлетворение от взаимодействия с веб-приложением.

Надёжность (Reliability) – критерий, оценивающий способность системы противостоять ошибкам и величину готовности к восстановлению после падений. Этот принцип особенно значим для платформ, где неполадки могут повлечь серьёзные финансовые потери или недовольство клиентов.

Производительность (Performance), характеризующая скорость ответа системы и общую работоспособность.

Поддерживаемость (Supportability) представляет собой лёгкость в адаптации, обслуживании, расширении и настройке программного продукта, что, несомненно, имеет значение для его долгосрочной стабильности и эффективности.

Согласно методологии FURPS+, веб-представительство кадрового агентства должно разрабатываться в соответствии с обозначенными требованиями, которые отражены в таблице 1.

Таблица 1 – Требования к веб-представительству кадрового агентства в методологии FURPS+

Требование	Статус	Полезность	Риск	Стабильность
Функциональные требования				
Наличие личного кабинета	Обязательное	Критическая	Средний	Высокая
Главная страница является входной точкой в веб-приложение	Обязательное	Критическая	Низкий	Высокая
Формирование списка предложений	Обязательное	Критическая	Средний	Высокая
Управление вакансиями или резюме	Обязательное	Критическая	Средний	Высокая

Продолжение таблицы 1

Требование	Статус	Полезность	Риск	Стабильность
Удобство использования				
Интуитивная навигация и понятный интерфейс	Обязательное	Критическая	Средний	Средняя
Основной контент расположен в контейнере по середине	Обязательное	Важная	Средний	Средняя
Надежность				
Доступ пользователю 24/7/365	Обязательное	Критическая	Средний	Высокая
Среднее время сбоев: до 1 минуты 1 в день	Обязательное	Критическая	Средний	Средняя
Производительность				
Время реакции системы на запросы должно быть не более 2 секунд	Обязательное	Критическая	Средний	Высокая
Поддерживаемость				
Время устранения возникших проблем 48 часов (В рабочее время)	Обязательное	Критическая	Средний	Высокая

Соблюдение данных требований является фундаментом надежного веб-приложения кадрового агентства, что соответствует миссии фирмы предоставлять качественные услуги своим клиентам.

1.3 Анализ аналогов веб-представительства кадровых агентств

В растущей атмосфере конкуренции, компаниям по предоставлению кадровых услуг необходимо постоянно совершенствоваться. Исследуя поведение потребителей, агентства по трудоустройству стремятся организовать сервис наилучшего качества.

Для достижения лидирующих позиций, необходимо предварительно собрать информацию о конкурентах. Данный этап предваряет создание

веб-представительства, чтобы выделить существенные особенности, повышающие конкурентоспособность продукта.

Как пример можно привести агентство «КадрФФ», которое охватывает весь спектр услуг по поиску и привлечению персонала для бизнес-структур [16]. Это агентство имеет собственный лэндинг – одностраничный веб-сайт, излагающую основные аспекты их методологии работы, включая механизм подачи заявок и средства коммуникации с клиентами, что подтверждает скриншот их основного раздела сайта, представленный на рисунке 1.

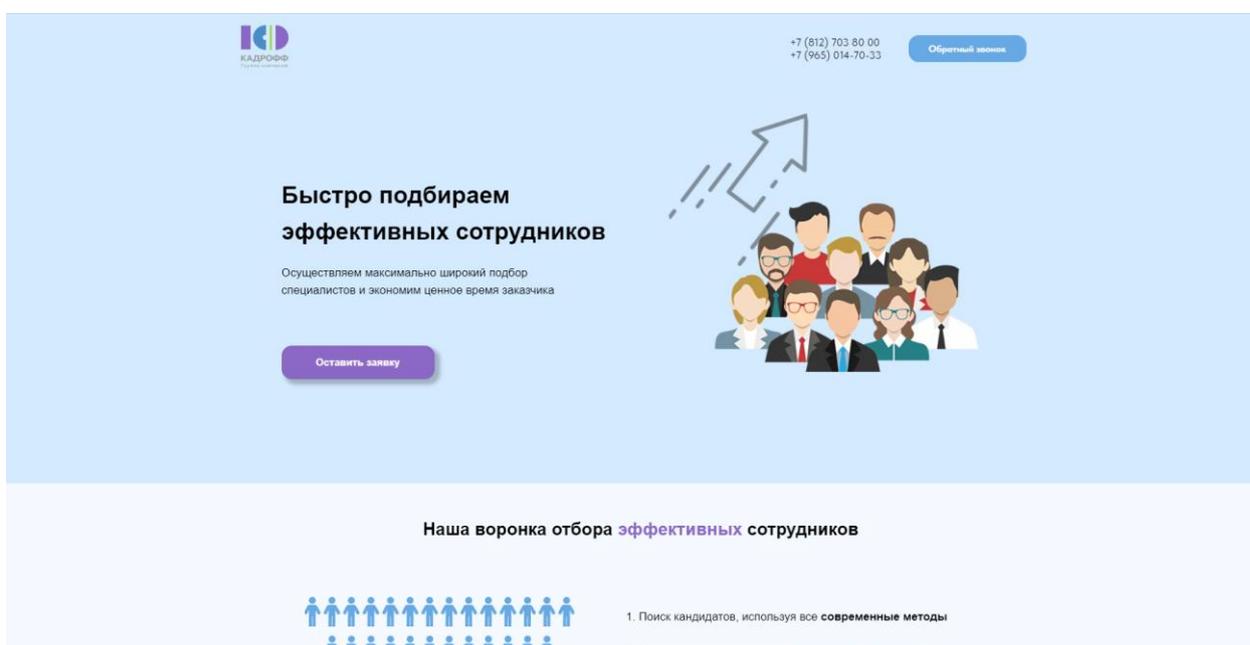


Рисунок 1 – Скриншот главной страницы

Особенность данного сервиса в разработке через конструктор сайтов «Tilda», такое решение с одной стороны имеет главное преимущество, это ускорение разработки, с другой не позволяет разрабатывать сложный и масштабируемый функционал, что может повлиять на дальнейшее восприятие пользователей. Указанный инструмент предлагает интуитивно понятные шаблоны и редакторские инструменты, облегчая тем самым процесс разработки сайта, без обращения к знаниям программирования.

Преимущества, которыми отличаются конструкторы веб-сайтов подобного рода:

- неподготовленные пользователи обретают возможность самостоятельной разработки собственного веб-сайта;
- экономическая выгода и временная эффективность в связи с возможностью проведения функциональных и дизайнерских внедрений самостоятельно;
- широкий диапазон предлагаемых компонентов, обеспечивающих простор для внесения изменений и включения контента, включая мультимедийные файлы и интерактивные формы для обратной связи.

Вместе с тем, существуют недостатки, характерные для данного подхода:

- не всегда присутствует возможность добавления сложной функциональности или интеграции сторонних сервисов;
- ограниченный выбор шаблонов и дизайна, что может привести к появлению неоригинальных сайтов;
- нарушения в области SEO-оптимизации и управлении ресурсами из-за ограниченного доступа к исходному коду;
- могут быть ограничения по хостингу и хранению данных.

Дополнительно замечено, что применение стандартизированных решений влечёт за собой проблемы брендинга на веб-сайте, что представлено на рисунке 2, где фиксируются логотип «Tilda» в заголовке веб-страницы и упоминания принадлежности сайта к конструктору.

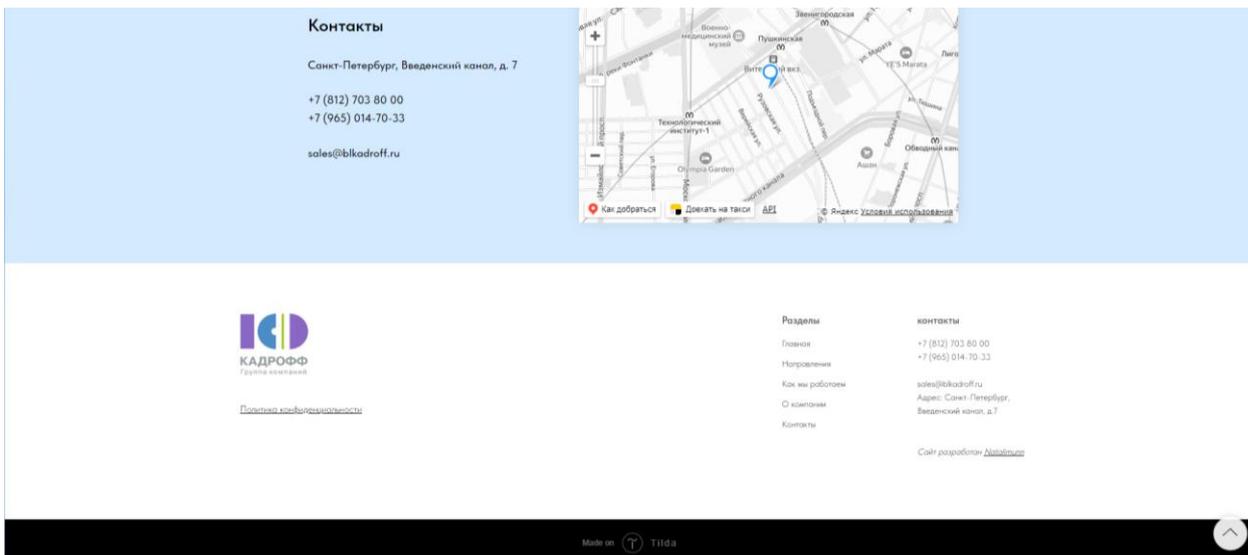


Рисунок 2 – Футер главной страницы сайта «Кадровфф»

Рассмотри другого конкурента «Аспект», функциональность которого превосходит веб-представительство «Кадровфф» [17]. В данном примере пользователи встречаются с таким расширением функционала, как навигация по разделам, списком горящих вакансий и ссылки на соцсети.

Визуализация главной страницы агентства «Аспект» представлена на рисунке 3.

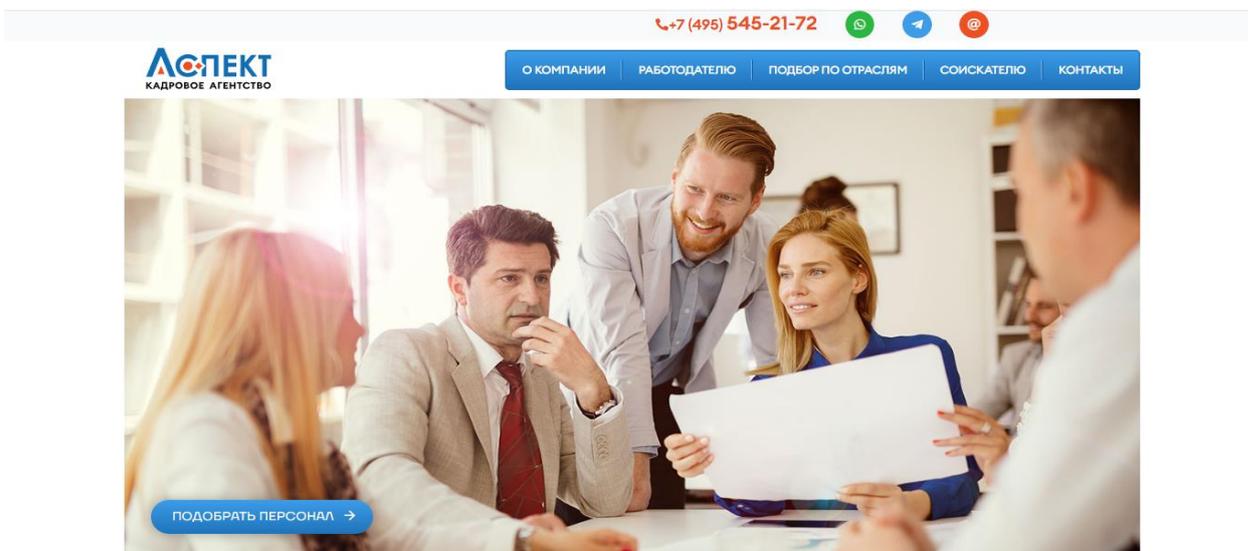
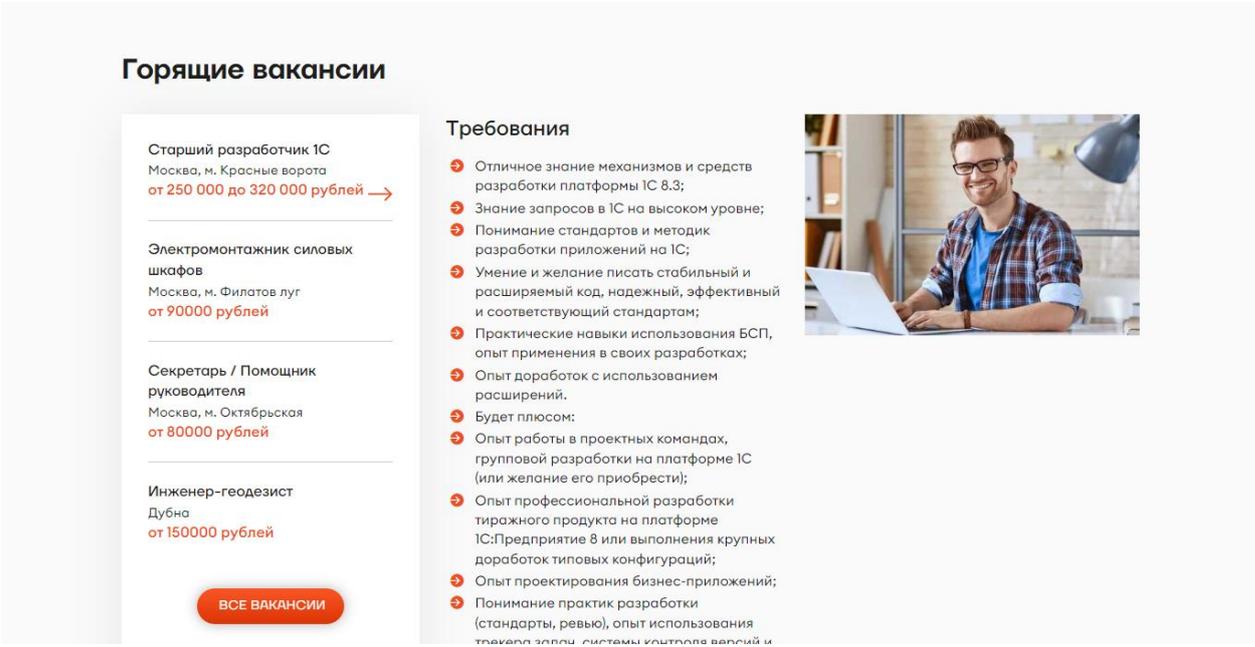


Рисунок 3 – Главная страница веб-представительства кадрового агентства «Аспект»

Данный сайт, выделяется тремя основными элементами, форма для запроса обратного звонка, где клиенты вносят стандартные данные вроде имени и номера телефона; модулем для подачи заявки на отбор персонала, что дублирует указанные выше поля; и секцией с контактной информацией и указанием местоположения офиса на картографическом сервисе.

Недостаточность предложений, адресованных сотрудникам, и отсутствие резюме специалистов ставят организацию в зависимость от внешних веб-приложений для набора персонала. Помимо того, присутствует потребность агентства в расширенном штате операторов контактного центра и менеджеров для обеспечения высокого уровня обслуживания. Принимая во внимание указанные факторы, рекомендуется доработка платформы для исключения утечки потенциальных клиентов, и повышение функциональности.

Раздел, отображающий список доступных вакансий, чья реализация способствует повышению конкурентоспособности и привлечению трафика соискателей, изображено на рисунке 4.



Горящие вакансии

Старший разработчик 1С
Москва, м. Красные ворота
от 250 000 до 320 000 рублей →

Электромонтажник силовых шкафов
Москва, м. Филатов луг
от 90000 рублей

Секретарь / Помощник руководителя
Москва, м. Октябрьская
от 80000 рублей

Инженер-геодезист
Дубна
от 150000 рублей

ВСЕ ВАКАНСИИ

Требования

- Отличное знание механизмов и средств разработки платформы 1С 8.3;
- Знание запросов в 1С на высоком уровне;
- Понимание стандартов и методик разработки приложений на 1С;
- Умение и желание писать стабильный и расширяемый код, надежный, эффективный и соответствующий стандартам;
- Практические навыки использования БСП, опыт применения в своих разработках;
- Опыт доработок с использованием расширений.
- Будет плюсом:
- Опыт работы в проектных командах, групповой разработки на платформе 1С (или желание его приобрести);
- Опыт профессиональной разработки тиражного продукта на платформе 1С:Предприятие 8 или выполнения крупных доработок типовых конфигураций;
- Опыт проектирования бизнес-приложений;
- Понимание практик разработки (стандарты, ревью), опыт использования трекера задач, системы контроля версий и



Рисунок 4 – Раздел с «горящими» вакансиями

Устаревание интерфейса и legacy-код составляют ключевую проблематику для интернет-ресурса, влияя на их взаимодействие с платформой. Актуальность внедренной в прошлом библиотеки JQuery, которая ныне считается не отвечающей современным требованиям, поскольку была вытеснена более новыми фреймворками и библиотеками, маркирует технологическое отставание проекта [22]. Перечень запросов к упомянутой библиотеке продемонстрирован на рисунке 5.

www.jobaspect.ru	200	document	Other	17.8 kB	111 ms
bootstrap.min.css	200	stylesheet	index:38	(disk cache)	15 ms
animate.css	200	stylesheet	index:39	(disk cache)	14 ms
owl.carousel.min.css	200	stylesheet	index:40	(disk cache)	14 ms
owl.theme.default.min.css	200	stylesheet	index:41	(disk cache)	14 ms
style.css	200	stylesheet	index:42	(disk cache)	15 ms
style.css	200	stylesheet	index:48	(disk cache)	14 ms
logo.svg	200	svg+xml	index:122	(disk cache)	15 ms
kadrovoe_agentsvto_aspekt.jpg	200	jpeg	index:168	(disk cache)	16 ms
blob:https://www.jobaspect.ru/274f62cc-2b25-4e8a-83b5-1f214d...	200	script	VM17:41	0 B	23 ms
ostorozhno_moshenniki.jpg	200	jpeg	index:259	(disk cache)	148 ms
na_rynke_hr-uslug_s_2008_goda.svg	200	svg+xml	index:289	(disk cache)	148 ms
kachestvo_uslug.svg	200	svg+xml	index:294	(disk cache)	150 ms
jquery-3.3.1.min.js	200	script	index:1465	(disk cache)	154 ms
bootstrap.min.js	200	script	index:1466	(disk cache)	152 ms
jquery.mask.js	200	script	index:1467	(disk cache)	153 ms
jquery.mousewheel.min.js	200	script	index:1468	(disk cache)	163 ms
wow.min.js	200	script	index:1469	(disk cache)	166 ms
parallax.min.js	200	script	index:1470	(disk cache)	167 ms
owl.carousel.min.js	200	script	index:1471	(disk cache)	169 ms
metisMenu.min.js	200	script	index:1472	(disk cache)	171 ms
lightzoom.js	200	script	index:1474	(disk cache)	171 ms
main.js	200	script	index:1476	(disk cache)	173 ms
modals.js	200	script	index:1477	(disk cache)	173 ms
captcha.js	200	script	index:65	36.0 kB	230 ms
kvalificirovannye_konsultanty.svg	200	svg+xml	index:299	(disk cache)	174 ms
sovremennye_tehnologii.svg	200	svg+xml	index:304	(disk cache)	174 ms
aktualnaya_baza_rezume.svg	200	svg+xml	index:309	(disk cache)	174 ms
dolgosrochnoe_sotrudnichestvo.svg	200	svg+xml	index:314	(disk cache)	174 ms
spetsializaciya_po_otraslyam.svg	200	svg+xml	index:319	(disk cache)	174 ms
konfidentialnost.svo	200	svg+xml	index:324	(disk cache)	174 ms

369 requests | 190 kB transferred | 14.3 MB resources | Finish: 14.1 min | DOMContentLoaded: 1.19 s | Load: 6.51 s

Рисунок 5 – Список запросов на получение скриптов

Веб-приложение «SuperJob» выделяется на рынке интернет-ресурсов своими конкурентными преимуществами и является местом встречи для поиска взаимовыгодного сотрудничества между работодателями и соискателями [18]. Функциональный спектр данного сервиса организован таким образом, что обеспечивает удобную навигацию и интуитивно понятное взаимодействие с системой через различные модули.

Индивидуальный доступ к системе через личный кабинет и процедуру авторизации способствует уменьшению нагрузки на call-центры и менеджеров, при этом авторизованным пользователям предоставляется возможность самостоятельного управления своими вакансиями и резюме – важный аспект в оптимизации процесса трудоустройства.

Предложенный сервисом список вакансий – это инструмент, который становится доступным после выполнения входа в систему сотрудниками, позволяя им ознакомиться с персонализированными предложениями о работе.

Возможности редактирования собственных предложений о работе – это функционал, направленный на повышение самостоятельности пользователей и минимизацию их потребности в ассистировании со стороны службы поддержки, давая больше контроля над контентом публикуемых объявлений.

Основная страница приложения, как представлена на рисунке 6.

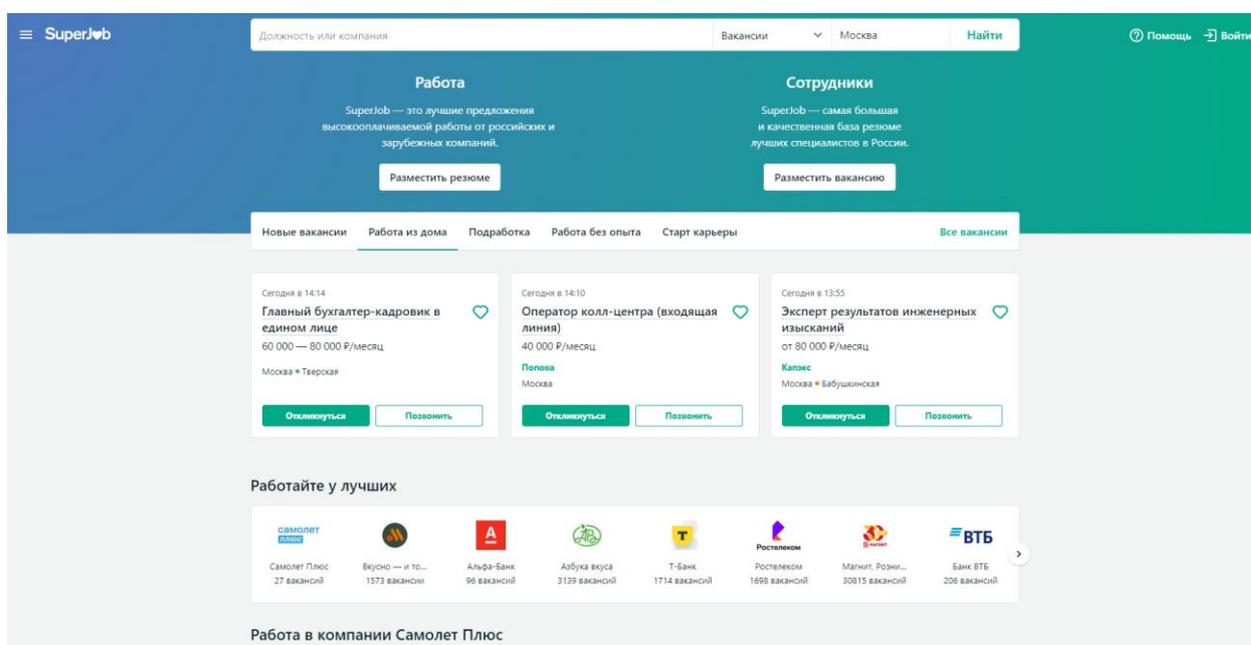


Рисунок 6 – Главная страница портала «SuperJob»

Функциональные аспекты веб-интерфейса открываются перед пользователем мгновенно, повышая тем самым скорость в решении воспользоваться онлайн-платформой. Система навигации ресурса реализована

интуитивно понятно, предлагая быстрый переход между разделами. Тем не менее, дефицит записей на домашней странице может вызвать задержку в оценке предложения пользователями. Можно выделить основные замечания и предложения:

- обогащение страницы дополнительным списком, возможно с добавлением пагинации или опцией скользящего просмотра;
- рекомендуется внедрение элементов с плавными анимациями, добавление теней и округлостей к интерфейсным деталям, чтобы оживить визуальное восприятие.

Карточка вакансии требует полного представления информации об условиях труда и описания должности. Чтобы удовлетворить данный запрос, следует реорганизовать контент так, чтобы данные о вакансии оказались визуально приоритетными и предшествовали названию и символике фирмы. Страница изображена на рисунке 7.

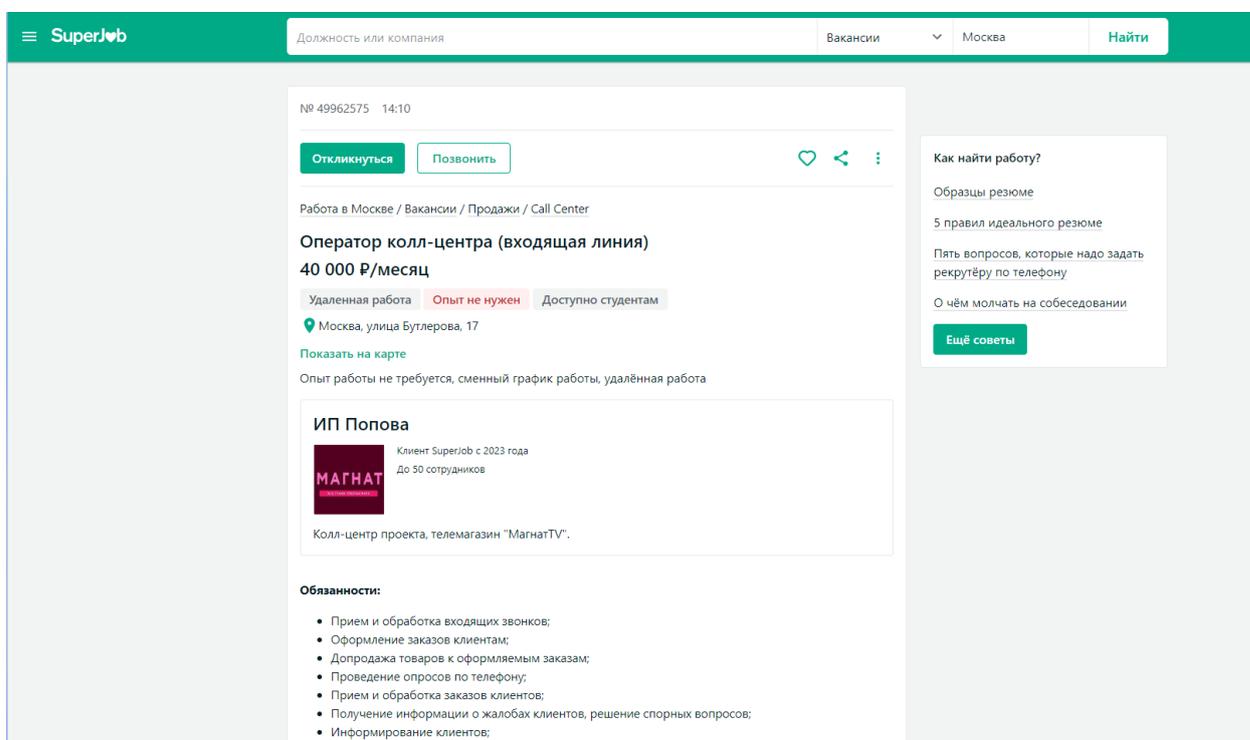


Рисунок 7 – Детальная информация карточки вакансии «Оператор Call-центра» портала «SuperJob»

Таблица 2 – Сравнительный анализ веб-представительств кадровых агентств

Характеристика	Кадрофф	Аспект	SuperJob
Наличие личного кабинета	0	0	3
Главная страница является входной точкой в веб-приложение	2	2	3
Формирование списка предложений	0	1	3
Управление вакансиями или резюме	1	1	3
Интуитивная навигация и понятный интерфейс	1	1	2
Основной контент расположен в контейнере по середине	3	3	3

Применяем численную оценку согласно нижеуказанным категориям для анализа степени совмещения тремя веб-ресурсами кадровых агентств с заранее определенными нормами функционирования:

- полная эквивалентность указанным нормам получает оценку «3»;
- минимальное отклонение от норм отмечается как «2»;
- крупное отклонение наблюдается с оценкой «1»;
- при отсутствии соответствия нормам ставится «0».

Кадрофф, Аспект и SuperJob. Оценочные баллы за показатели, следующие:

- возможность входа пользователем в индивидуализированный раздел сайта – Агентство Кадрофф и Аспект не предоставляют эту функцию, показатель «0», тогда как SuperJob идеально справляется с этой задачей – показатель «3»;

- главная страница сайта как портал взаимодействия с веб-приложением – Кадровфф и Аспект имеют оценку «2», SuperJob полностью выполняет функцию, оценка «3»;
- создание списка актуальных предложений работы или подбора персонала – Кадровфф не реализует данный критерий, что влечет показатель «0»; Аспект частично соответствует требованию, оценка «1»; SuperJob полностью удовлетворяет критерию, получая «3»;
- организация управления размещаемыми вакансиями и резюме – Кадровфф и Аспект одинаково не полностью соответствуют, оценка «1»; в то время как SuperJob достигает максимального результата с оценкой «3»;
- присутствие интуитивной навигации в сочетании с понятным интерфейсом – Кадровфф и Аспект удостоиваются оценки «1», SuperJob отвечает критериям лучше, но не идеально, оценка «2».

Страница «Кадровфф» несмотря на то, что была создана с помощью популярного веб-конструктора, сталкивается с барьерами расширения и интеграции передовых бизнес-процессов из-за ограниченных возможностей платформы «Tilda». А отсутствие функции контроля через личный кабинет является существенным препятствием к современному веб-представительству.

«Аспект», обладая развитой онлайн-платформой, все же не может похвастаться современностью: устаревший интерфейс в сочетании с унаследованной кодовой структурой снижают ее эффективность для пользователей.

Для компании «SuperJob», хотя и замечено наличие достойного веб-приложения, но стоит увеличить количество интерактивных элементов, а также необходимо ускорить процессы взаимодействия клиента с платформой.

Учитывая вышеизложенное, конечные рекомендации для веб-представительства нашего кадрового агентства предполагают необходимость превзойти конкурентные платформы в функциональности, внешней привлекательности, а также юзабилити. Веб-приложение должно включать

раздел с информацией о предложениях, личный кабинет с возможностью редактирования данных, а также интуитивно понятную форму для создания вакансий и резюме.

Выводы по главе 1

В 1 главе выпускной квалификационной работы был рассмотрен анализ функциональных и архитектурных характеристик веб-представительств для кадровых агентств, было проведено исследование с целью сравнения существующих аналогов и выработки критериев для нашего собственного веб-представительства. Изучив аналогичные сервисы, мы пришли к выводу, что наше веб-представительство кадрового агентства должно отвечать следующим пунктам:

- сайт должен иметь современный и актуальный дизайн интерфейса, соответствующий настоящим трендам веб-разработки;
- передвижение по веб-приложению должно быть интуитивно понятным, чтобы упростить пользователям исследование сайта;
- вся информация на страницах должна быть выполнена в простом и понятном виде;
- на сайте должны быть представлены список вакансий и резюме, личный кабинет пользователя, система авторизации и список откликов;
- на сайте должна быть возможность онлайн редактирования и создание вакансий или резюме;
- должна отображаться краткая информация о вакансии или резюме в списке.

Эти требования способствуют созданию веб-представительства кадрового агентства, предлагая не только эстетически привлекательную и емкую пользовательскую среду, но и мощный инструмент для достижения целей компании.

Глава 2 Проектирование программного обеспечения веб-представительства кадрового агентства

2.1 Логическое моделирование веб-представительства

Веб-представительство, акцентирует внимание на синхронизации взаимодействия трёх ключевых субъектов:

- работодатель,
- работник,
- специалист технической поддержки.

С учётом функционала, упрощающего создание вакансий и резюме, мы обозначили специфические сценарии взаимодействия данных акторов с системой.

Для моделирования диаграммы используется язык моделирования UML, который обеспечивает единый визуальный язык для проектирования и внедрения программных средств [2].

Разработанная диаграмма представлена на рисунке 8.

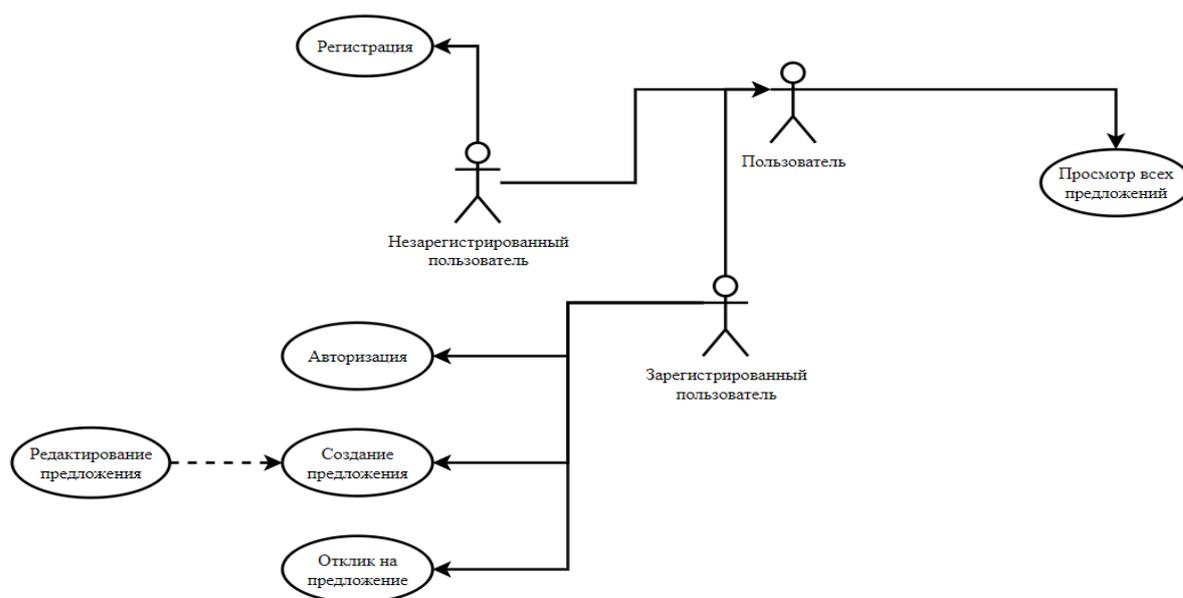


Рисунок 8 – Диаграмма прецедентов веб-представительства

Веб-ресурс предназначен для взаимодействия сотрудников и работодателей, где они знакомятся с предложениями друг для друга. К тому же, предусмотрена опция регистрации индивидуальных предложений. Подав ответ, пользователи могут активно принимать участие в подборе персонала или поиске работы.

Техническая поддержка, состоящая из квалифицированных экспертов, обеспечивает бесперебойную работу сайта. Они не только реагируют на запросы пользователей, но и заботятся о качестве контента, а также учитывают отзывы о практичности интерфейса, способствуя его оптимизации для более эффективного использования.

На рисунке 9 представлен прецедент под наименованием «Принять предложение» [24].

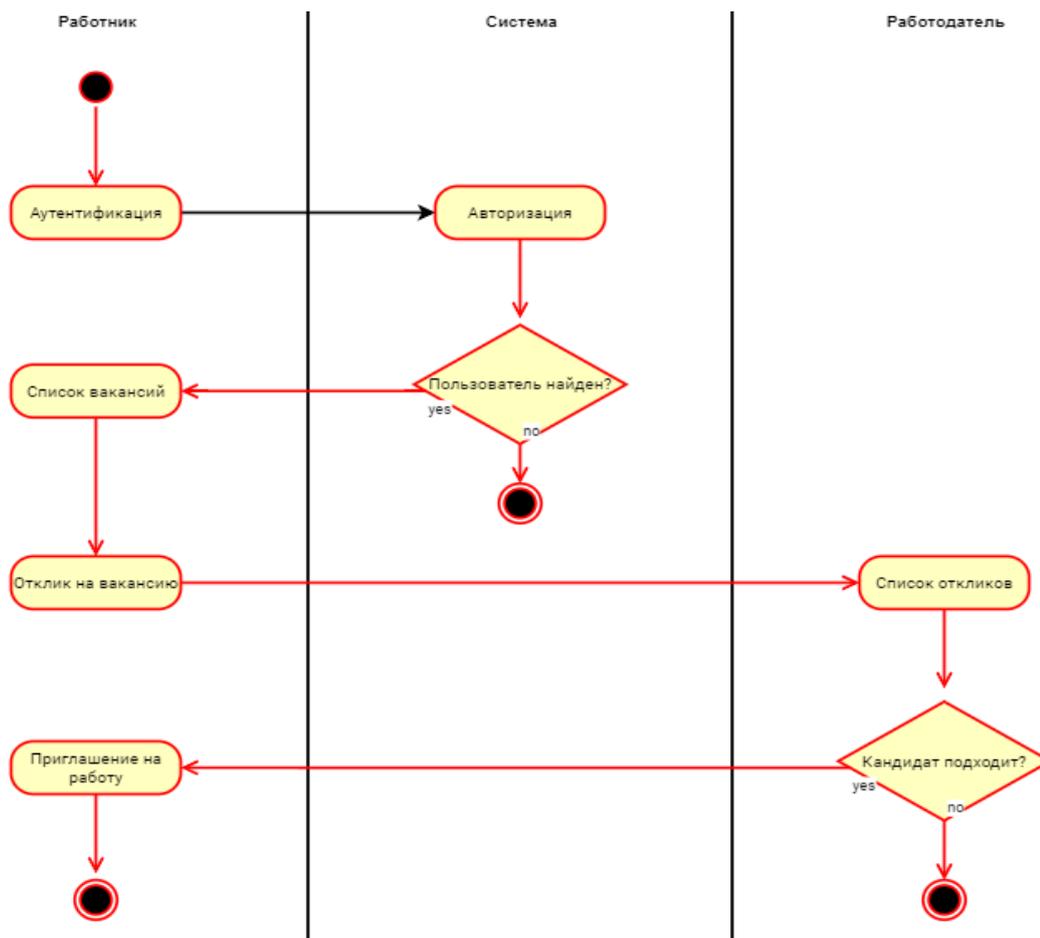


Рисунок 9 – Диаграмма деятельности процесса принятия предложения

В центре процесса стоит работник, проходящий аутентификацию в сервисе как начальный этап, что демонстрирует диаграмма. Его успешная авторизация открывает доступ к вакансиям. После рассмотрения различных вакансий и выбора отвечающей критериям, работник направляет отклик, который предстоит рассмотреть работодателю.

Последующий этап подразумевает оценку соответствия работника предложенной вакансии, что приводит к возможности отклика работодателем. В случае положительной оценки потенциальному сотруднику будет направлено уведомление о приглашении на работу, что завершает эту часть процедуры взаимодействия в системе.

Визуализируя структуру системы, диаграмма классов в языке UML (Unified Modeling Language) играет ключевую роль [9]. Системные классы со своими атрибутами и методами, а также связи между объектами являются её составной частью. Эти диаграммы представляют собой основу не только для осознания архитектуры системы разработчиками, но и для дальнейшей реализации программного кода, поддерживая ясность представления о структуре системы.

Из анализа функциональных заданий и бизнес-требований сетевого представительства рекрутинговой организации, мы выявили ключевые сущности и их взаимосвязи, отражённые на схеме под номером 10 [23].

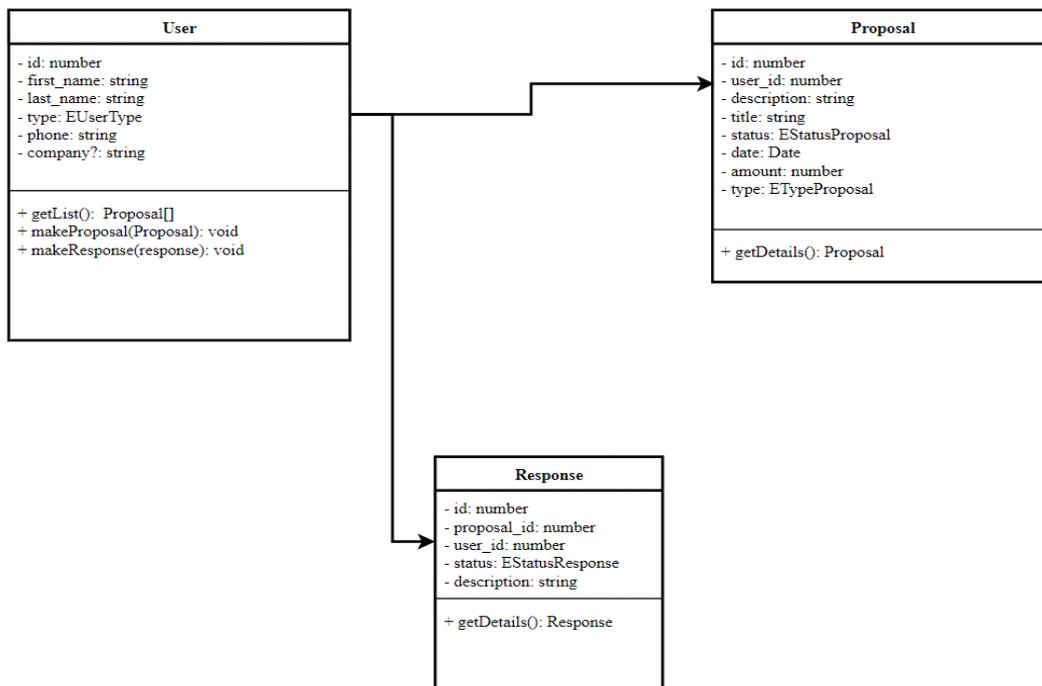


Рисунок 10 – Диаграмма классов веб-представительства

В таблице 3 демонстрируется спецификация классов веб-представительства.

Таблица 3 – Спецификация диаграммы классов веб-представительства

Класс	Описание
User	Класс объектов, моделирующих на логическом уровне пользователей
Proposal	Класс объектов, моделирующих на логическом уровне предложения
Response	Класс объектов, моделирующих на логическом уровне ответы на предложения

В контексте создания сетевого представительства для кадровых агентств, спецификация в таблице 3 выступает в качестве неотъемлемого элемента. Она детализирует классы для веб-представительства, их характеристики и

взаимодействия. Класс «User» инкапсулирует данные и функциональность, относящиеся к пользователям, в то время как классы «Proposal» и «Response» отражают категории предложений и отклики на них.

Инструментарий, представленный диаграммой классов, оказывается незаменимым при организации веб-ресурсов. Он не только способствует структурированию процесса разработки, делая его интерпретируемым и прозрачным, но и упрощает последующее написание кода. Эта диаграмма служит основой, на которую мы будем опираться при разработке. Так, мы получаем возможность осмысленно налаживать взаимосвязи между составляющими системы, что в итоге приводит к созданию функционального и качественного веб-представительства.

2.2 Логическое моделирование данных веб-представительства

В сфере управления информацией преобладают два типа систем хранения данных: SQL и NoSQL, каждая из которых обладает уникальным набором характеристик и диапазоном использования.

SQL (Structured Query Language) – это язык для работы с реляционными базами данных, которые хранят информацию в структурированном виде, как в таблице с четко определенными столбцами. SQL идеально подходит для управления данными, которые имеют определенную структуру [12].

NoSQL (Not Only SQL) – это группа нереляционных баз данных, которые используют различные структуры для хранения информации, например, документы или пары ключ-значение. NoSQL лучше подходит для работы с данными, которые не имеют четкой структуры [13].

Наш выбор системы управления базами данных для проекта пал на NoSQL, с учетом следующих факторов:

- модифицируемость: NoSQL демонстрирует выдающуюся способность адаптироваться к изменяющимся требованиям

структуры данных благодаря отсутствию ограничений на объем колонок;

- быстрота развертывания: В среде NoSQL отсутствует предварительное условие детального оформления схем базы данных, что значительно ускоряет процесс внедрения;
- минимализм в извлечении данных: NoSQL оснащен синтаксически упрощенным языком запросов, который резонирует с запросами нашего проекта. [20]

В сценариях, требующих выверенной точности и контроля в обработке данных и транзакциях, например, в банковской сфере, предпочтение отдают структурам SQL [15]. Однако NoSQL зарекомендовал себя как многогранная технология, подходящая для манипуляций с разнообразными источниками информации и ее архитектурными композициями.

В базах данных NoSQL схемы хранения данных дифференцируются на четыре основных архетипа.

Объекты данных, имеющие простую ключевую структуру, где уникальный ключ связан со значением, формируют категорию хранилищ «ключ-значение». Они находят широкое применение для операций быстрого доступа и эффективного кэширования.

Документо-ориентированные системы удерживают информацию в документных коллекциях, применяя для их структурирования формы JSON, XML или BSON. Группирование схожих документов усиливает функциональность по управлению и индексации контента.

Столбцовые хранилища, предпочтительны в контексте аналитических процессов. Они организованы по столбцам, где каждый столбец представляет собой логический массив значений [19].

Графовые хранилища, последний тип в этой классификации, используют узлы (изолированные документы с произвольным содержимым) и ребра, которые связывают эти узлы. Этот тип отлично подходит для задач, где важно

создавать взаимосвязи между данными, например, в системах рекомендаций, оптимизации маршрутов и т.д. [19].

Специализированный веб-ресурс, разрабатываемый нами, оперирует на платформе документно-ориентированных баз данных. Используемая база данных организована в трехмерное пространство коллекций:

- proposals,
- responses,
- users.

Каждый документ в коллекции proposals содержит уникальный идентификатор id, наряду с подробным описанием предложения, ожидаемой суммой вознаграждения, датой подачи и текущим статусом обращения. Коллекция responses и users структурированы аналогичным образом: в responses каждый ответ ассоциируется с уникальным идентификатором клиента user_id, ссылкой на соответствующее предложение proposal_id, статусом и комментарием, тогда как users содержит личные данные пользователя, тип роли в трудовой сфере и наименование соответствующей компании.

Таким образом, благодаря структурированному хранению информации, возможно высокоэффективное управление данными предложений, откликов и профилей пользователей, что позволяет создавать динамичное и легко расширяемое веб-приложение.

2.3 Разработка внутренней структуры веб-представительства

Определение адекватной структуры стоит в начале пути создания цифрового информационного продукта. Она служит костяком будущего ресурса и логической основой, устанавливающей взаимное расположение разделов и страниц.

Структура цифрового ресурса требует вариабельности, изменяясь с учетом цели данного веб-представительства. Сложность и многоуровневость

структуры разительно отличается между, например интернет-магазинами и сайтами-визитками.

Ключевыми архитектурами структур ресурсов выступают:

- линейная архитектура: страницы последовательно соединены между собой. Простота такой последовательности делает её предпочтительной для менее сложных веб-проектов, таких как сайты-портфолио или визитки;
- древовидная архитектура: представляет собой иерархию от общего к частному во взаимосвязи страниц, часто встречающейся в более крупных веб-ресурсах, с обширным функционалом [14].

Оценив все положительные стороны, мы пришли к выводу о целесообразности разработки веб-представительства нашего кадрового агентства именно в таком иерархическом ключе, как показано на рисунке 11.

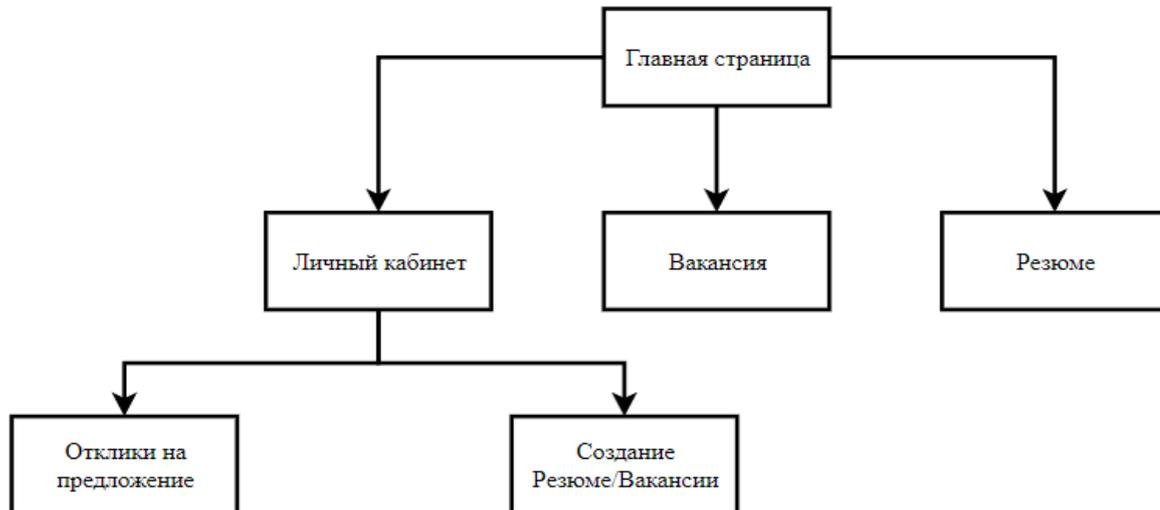


Рисунок 11 – Структурная схема сайта кадрового агентства

Когда пользователь переходит на сайт, он сразу попадает на главную страницу. Навигация по сайту организована с помощью меню, расположенного в верхнем блоке сайта.

На главной странице размещается основная информация, которая необходима для привлечения внимания пользователя, например, список доступных предложений.

В разделе «Вакансия» содержится информация о вакансии. На ней можно ознакомиться с предложением и при желании отправить отклик.

В разделе «Резюме» содержится информация о кандидате. На ней можно ознакомиться с навыками кандидата и при желании отправить отклик.

В разделе «Личный кабинет» можно посмотреть информацию о пользователе, внести контактные данные и перейти в раздел «Отклики» или создание резюме/вакансии.

Раздел «Отклики» включает все ответы пользователя на предложения, здесь можно ознакомиться с решением со стороны работодателя или сотрудника.

Раздел «Создание резюме» содержит форму создания и список уже созданных предложений.

Интуитивная структура портала содействует быстрому освоению интерфейса пользователем, вносит вклад в клиентское удовлетворение, что способствует удержанию посетителя и стимулирует использование предлагаемых функций.

2.4 Разработка внешней структуры веб-представительства

Четырёхкомпонентная архитектура страницы – это универсальный канон в дизайне веб-порталов:

- шапка (Header), часто с логотипом, основным навигационным контролем, а также элементами аутентификации пользователя;
- боковое пространство (Sidebar), предоставляющее вспомогательные навигационные средства, информационные сегменты и рекламные материалы;

- центральное поле (Body), предназначенное для размещения первостепенной информации, целью которой является задерживание интереса аудитории;
- подвал страницы (Footer), где традиционно находятся данные контактов, ссылки на социальные сети и прочие вспомогательные ссылки.

Визуальный концепт, заложенный в экспозиции внешней композиции нашего сайта, зафиксирован в рисунке 12.

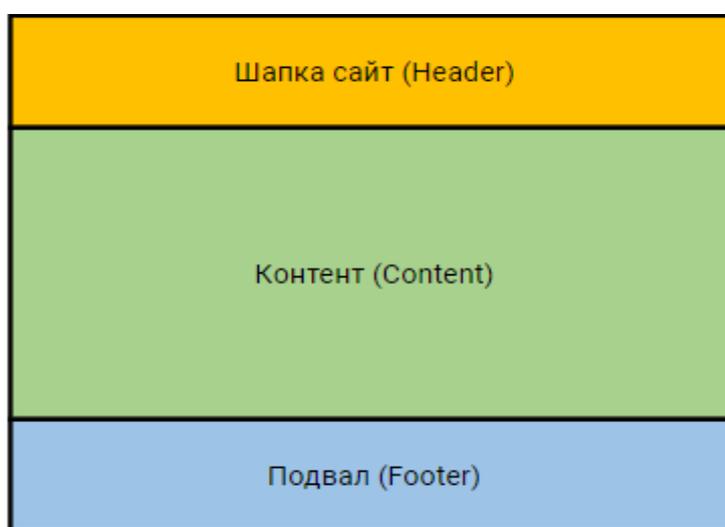


Рисунок 12 – Внешняя структурная схема сайта кадрового агентства

Три фундаментальных структурных компонента образуют архитектуру нашего кадрового агентства:

- шапка, где логотип нашей организации соседствует с доступом к личному кабинету;
- центральная часть предназначена для презентации первостепенного контента;
- подвал, включает в себя элементы навигации, фундаментальные сведения о компании, данные для связи, а также гиперссылки, ведущие в социальные сети.

Такой рациональный и стандартный подход в разработке структурных элементов обеспечивает пользователю простоту и интуитивную очевидность во взаимодействии с ресурсом. Также, обеспечивает его опыт пользования сервисом комфортным, последовательным и логично устроенным.

2.5 Разработка графического дизайна веб-представительства

Визуальное воплощение сайта, выраженное его стилистическим оформлением, занимает важную позицию. Совокупность таких атрибутов, как цветовое решение, графические элементы, выбор и применение шрифтов, а также использование иконок, формируют первостепенное впечатление от ресурса.

Колористические предпочтения находятся в прямой зависимости от бренда, избегая чрезмерной яркости, они нацелены на прочтение гармонии между фоном и текстом. Примером может служить применение различных тонов фирменного синего вместе с контрастным белым или желтым для улучшения восприятия текста.

Следование принципам высокого качества и согласованности в графических элементах, оказывают воздействие на скорость загрузки и внимание пользователей.

Шрифты, обязательно сходные с целостным стилем сайта, предпочтение легко читаемым и не утомляющим взгляд вариантам типа Arial или Verdana усиливаются потенциалом выразительных шрифтов для заголовков, в числе которых Times New Roman и Georgia.

Использование соответствующих стилю сайта иконок, которые обеспечивают простоту и интуитивность, способствуют облегчению навигации и украшают ресурс.

Коммерческие веб-страницы, обязанные привлекать клиентов, требуют специальной тщательности в соблюдении указанных дизайнерских принципов.

Выводы по главе 2

В этой главе мы подробно рассмотрели процесс проектирования веб-представительства кадрового агентства, пройдя все необходимые этапы. Работа началась с логического моделирования, включая разработку диаграмм прецедентов, активностей, классов и логическую структуризацию данных. Этот подход дал ясное понимание функций сайта, структуры данных и взаимодействий между ними.

Провели логическое моделирование и определили внутреннюю структуру веб-представительства, руководствуясь иерархическим принципом организации. Этот выбор дает возможность четко представить связи между страницами, что отражено в структурной схеме.

Анализ внешней организации веб-ресурса был следующим этапом, и результаты этого анализа вылились в определение стандартных разделов сайта. Помимо этого, представлена структурная схема, демонстрирующая расположение указанных блоков.

В конце мы сконцентрировались на графическом дизайне, подчеркивая значимость четырех основных элементов: цветовой палитры, графических элементов, шрифтов и иконок. Включены рекомендации по выбору этих элементов с учетом специфики, целевой аудитории и фирменного стиля агентства.

После завершения описанных процедур мы установили полное представление о внешнем и внутреннем виде нашего веб-представительства кадрового агентства.

Глава 3 Реализация и тестирование

3.1 Разработка графического дизайна веб-представительства

Информационные системы эффективно создаются при использовании соответствующих инструментов разработки, которые определяют конечные характеристики продукта, включая производительность, функциональность и надежность. Ассортимент доступных технологических инструментов и платформ в настоящее время представляется обширным, при этом каждый отдельный инструмент демонстрирует уникальные преимущества и характеристики.

Необходим анализ для выбора наилучшего набора инструментов, принимая во внимание конкретные задачи, требования клиента и способности разработчиков. Этот подход обеспечивает оптимальное решение из множества доступных при разработке сайтов.

Веб-редакторы, предлагают два основных вида интерфейсов:

- текстовые,
- визуальные (WYSIWYG).

Текстовые редакторы, требующие ручного кодирования, предоставляют возможность тщательной оптимизации и предлагают разработчику высокую степень свободы. Визуальные редакторы, напротив, предлагают более простую и интуитивно понятную среду для разработки, составляя код автоматически из стандартных элементов и позволяя видеть результат немедленно. Они подходят для тех, кто не стремится к глубокому погружению в языки кодирования вроде HTML и CSS.

Visual Studio Code, текстовый редактор, обладающий функциональностью интегрированной среды разработки (IDE), демонстрируется как образец этой категории инструментов на рисунке 13. Он включает механизмы, как автодополнение, многочисленные расширения, а также интеграцию с контролем версий через Git и инкорпорированный

отладчик для языков JavaScript, TypeScript, Node.js, предлагая всё это бесплатно.

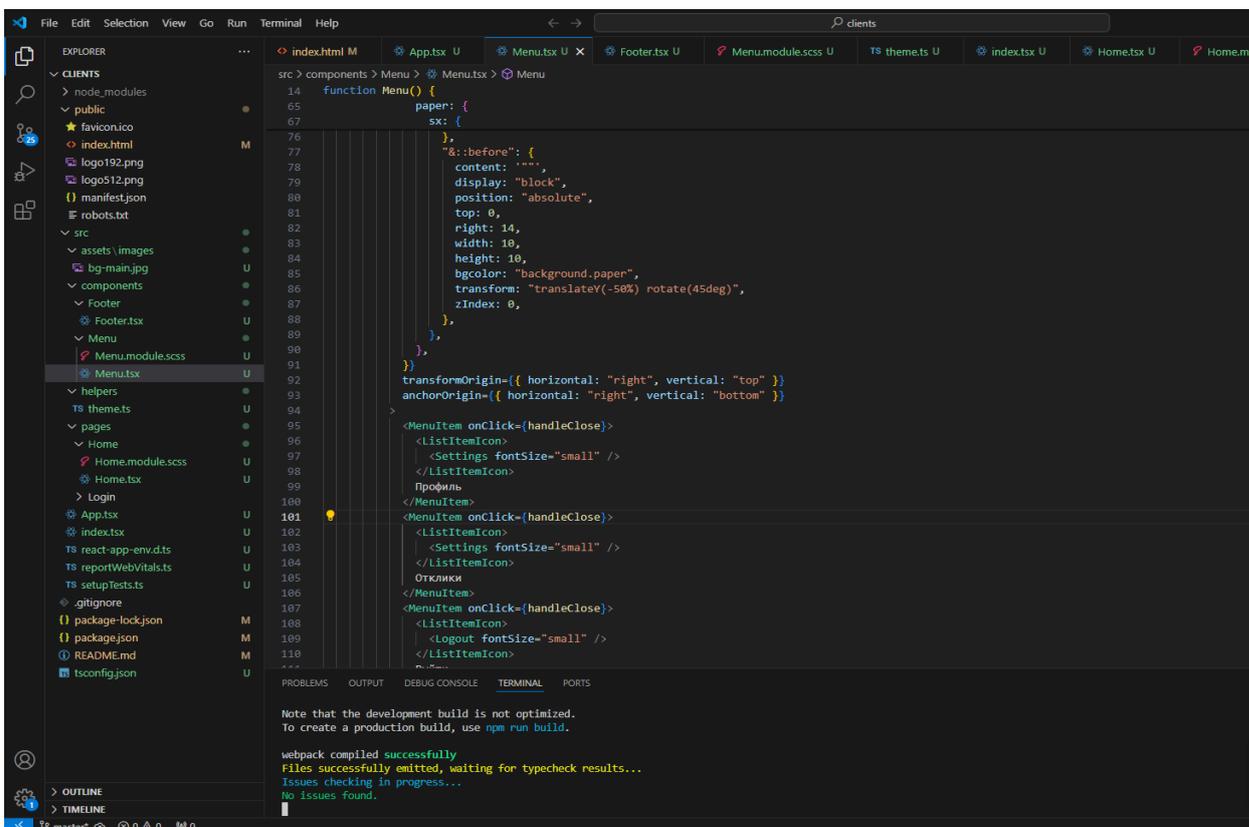


Рисунок 13 – Среда Visual Studio Code

На рисунке 14 представлен Sublime Text – этот редактор, прежде завоевавший популярность среди разработчиков веб-страниц на HTML, предлагается для использования в бесплатной версии с некоторыми ограничениями. Полный набор функций доступен после покупки лицензии. Кроме того, пользователи имеют возможность как интеграции готовых плагинов, так и разработки персонализированных дополнений.

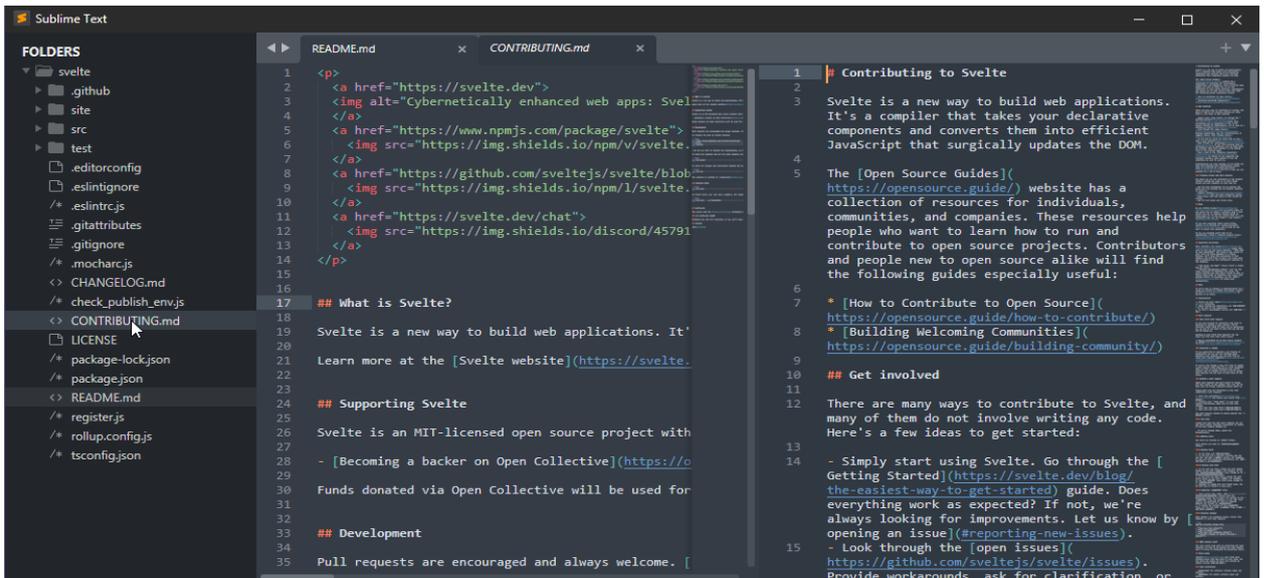


Рисунок 14 – Sublime Text

Также, специалисты часто предпочитают WebStorm от JetBrains, демонстрируемым на рисунке 15. Эта интегрированная среда предлагает разнообразный функционал, от реализации автоматизированного дополнения кода до модулей для проверки кода на ошибки, средства для его оптимизации и отладки, возможности для синхронизации с репозиториями на GitHub, а также множество доступных плагинов.

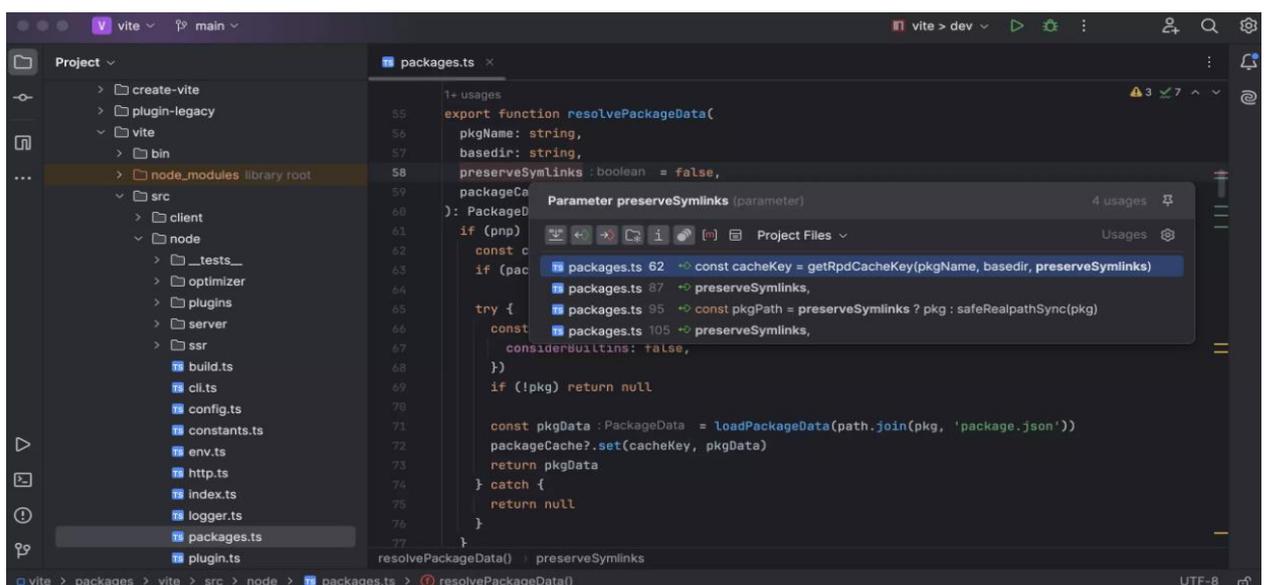


Рисунок 15 – WebStorm

Значимым инструментом в арсенале разработчиков веб-сайтов является Adobe Dreamweaver CC, демонстрируемый на рисунке 16.

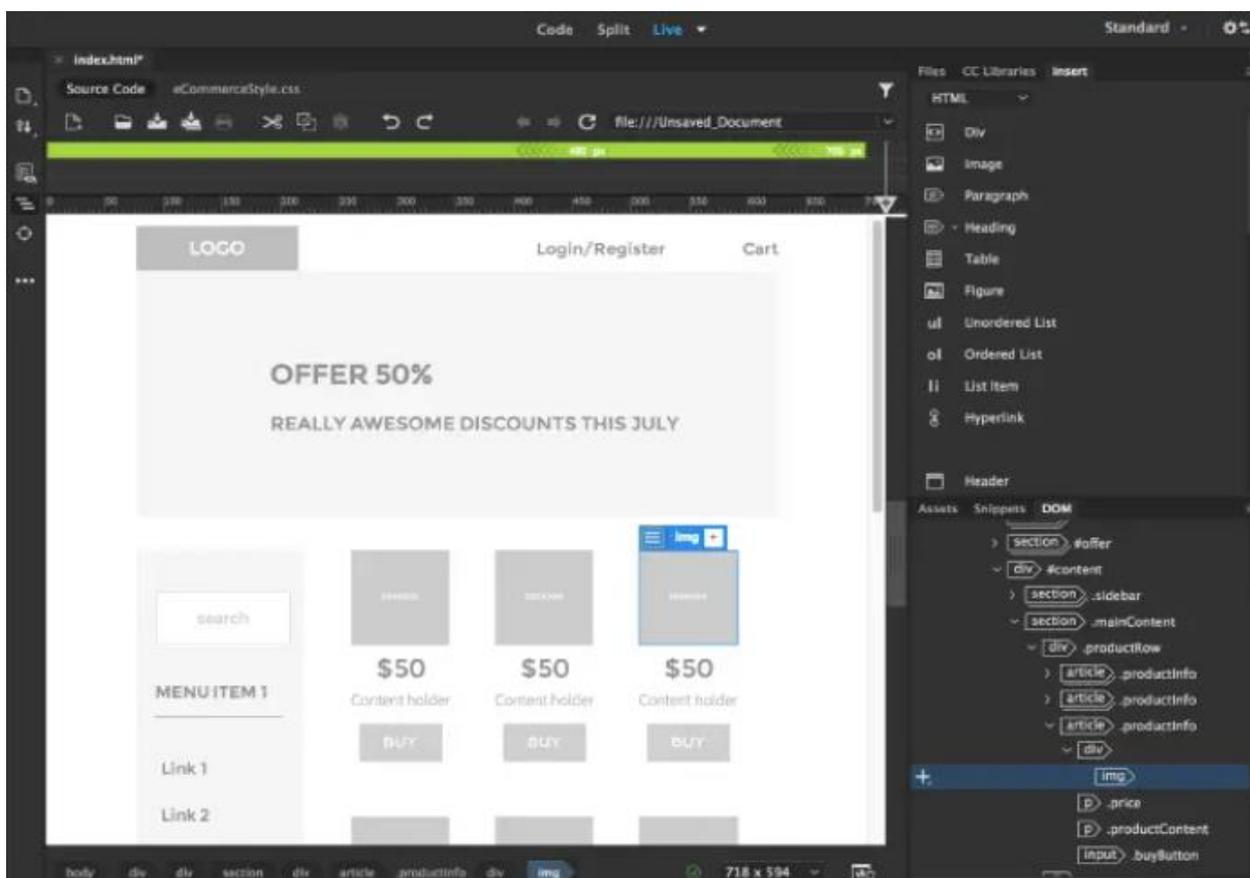


Рисунок 16 – Adobe Dreamweaver CC

Этот веб-редактор, созданный компанией Adobe Inc., обладает гибкими возможностями для креативного веб-дизайна, комбинируя текстовый и графический интерфейсы редактирования. Поддержка расширенного спектра языков программирования и предоставление функции предпросмотра в реальном времени значительно облегчают процесс веб-разработки, позволяя мгновенно увидеть итоговый результат труда разработчика. Несмотря на это, использование Dreamweaver обусловлено наличием подписки, стоимость которой весьма существенна, что представляет собой значимый минус данного продукта.

Для разработки веб-представительства кадрового агентства, мы отдали предпочтение использованию Visual Studio Code в силу его простоты,

бесплатного использования и наличия функционала, необходимого для скорейшего и эффективного процесса разработки.

3.2 Выбор технологии реализации проекта веб-представительства

Среди применяемых нами технологий – HTML, служащий каркасом для веб-страниц и выполняющий загрузку структуры и информационного наполнения через браузер при обращении пользователя. CSS мы задействовали для придания стилистики и эстетической презентабельности страниц, подняв общее визуальное восприятие сайта на более высокий уровень [21].

Динамичность и интерактивность интерфейса достигается посредством TypeScript, язык, который расширяет стандартные возможности JavaScript благодаря типизации. Также для создания интерфейса веб-приложения используется React, представляющего собой JavaScript фреймворк, который облегчает процесс разработки путем использования уже готовых функций и методов.

С целью оптимизации разработки и развертывания проекта интегрирован Webpack – инструментарий, способствующий ускоренной подготовке и запуску приложений.

Платформа Firebase была выбрана как инструментальная система, предоставляющая целый арсенал сервисов от баз данных и аутентификации до хостинговых решений, что обуславливает сетевое взаимодействия элементов проекта.

Серверная логика была реализована на языке Go, который предназначен для написания надежных и эффективных программ.

Синтаксическая простота HTML, выраженная через набор инструкций, известных как теги, упрощает демонстрацию содержимого веб-страницы в браузерах. Чтобы повысить эстетическую привлекательность и улучшить визуальное восприятие ресурса, применяется CSS, предоставляющий

инструментарий для стилизации и оптимизации пользовательского интерфейса [4].

Несмотря на многообразие возможностей по созданию интернет-проектов различной сложности с использованием HTML и CSS, ключевой недостаток заключается в их неспособности выполнять программные алгоритмы: интерактивность, работа с сервером, вычисления на стороне клиента и реакция на действия посетителей. Для решения этих задач используется язык программирования JavaScript. JavaScript, самый популярный язык для веб-разработки, который обеспечивает интерактивность для страниц [25].

JavaScript – это интерпретируемый язык, что означает, что код не нуждается в компиляции и может быть выполнен непосредственно браузером. Современные браузеры встроили в себя интерпретатор JavaScript, поэтому код может запускаться на любом компьютере [25].

React использует компонентный подход к разработке. Основной принцип заключается в том, что любой пользовательский интерфейс разбивается на множество независимых компонентов, каждый из которых отвечает за определенную часть интерфейса. Компоненты можно легко комбинировать и повторно использовать, что делает разработку более модульной и масштабируемой [7].

Для ускорения проектирования существует библиотека Material-UI. Этот инструмент – библиотека компонентов React с открытым исходным кодом, которая реализует Material Design от Google. Она является всеобъемлющей и может быть использована в рабочей среде из коробки. Содержит множество уже готовых компонентов для React, которые можно модернизировать под свой проект [6].

Пример компонента изображен на рисунке 17

```
src > components > Menu > Menu.jsx ...
11 import styles from './Menu.module.scss';
12
13
14 function Menu() {
15   const [anchorEl, setAnchorEl] = React.useState<null | HTMLElement>(null);
16   const open = Boolean(anchorEl);
17   const handleClick = (event: React.MouseEvent<HTMLElement>) => {
18     setAnchorEl(event.currentTarget);
19   };
20   const handleClose = () => {
21     setAnchorEl(null);
22   };
23
24   return (
25     <AppBar position="fixed">
26       <Toolbar className={styles["Menu"]} >
27         <div className={styles["Menu-Navigate"]} >...
28       </div>
29     </div>
30     <div>
31       <Box
32         sx={{ display: "flex", alignItems: "center", textAlign: "center" }}
33       >
34         <Tooltip title="Account settings">...
35       </Tooltip>
36     </Box>
37     <MUIMenu
38       anchorEl={anchorEl}
39       id="account-menu"
40       open={open}
41       onClose={handleClose}
42       onClick={handleClose}
43       slotProps={...
44     >
45       <MenuItem onClick={handleClose}>
46         <ListItemIcon>
47           <Settings fontSize="small" />
48         </ListItemIcon>
49         Профиль
50       </MenuItem>
51       <MenuItem onClick={handleClose}>
52         <ListItemIcon>
53           <Settings fontSize="small" />
54         </ListItemIcon>
55         Отклики
56       </MenuItem>
57     </>
58   );
59 }
60
```

Рисунок 17 – Пример компонента «Меню»

Основой фреймворка React является концепция состояния (state), представляющая собой изменяемые данные внутри компонентов. Непосредственно реакция на модификации данных заключается в авто-перерисовке компонентов, включая их дочерние элементы, для отображения нового состояния, такой процесс называется re-rendering.

Эффективность перерисовки достигается благодаря алгоритму виртуального DOM, сопоставляющего текущее и предшествующее состояния, оптимизируя обновления лишь актуализированных частей реального DOM [7].

В процессе работы, React сначала формирует виртуальный DOM-экземпляр для каждого компонента. При любых изменениях фреймворк анализирует виртуальный DOM с учетом новых данных, сопоставляет его с предыдущим состоянием и осуществляет обновление исключительно тех

областей реального DOM, которые претерпели изменения. Данная оптимизация гарантирует непрерывно быстрое и эффективное функционирование приложения, даже при частых изменениях данных.

Расширению возможностей React способствует его экосистема, состоящая из многообразных библиотек и инструментов. Примерами могут служить Router для навигации, Zustand для управления состоянием, Server-renderer для серверного рендеринга, Devtools для отладки в браузере, а также механизмы для интеграции HTML-кода.

Для работы с фреймворком требуется Node.js – среда выполнения JavaScript, обеспечивающая ряд функций, от запуска JavaScript-кода на сервере, управления зависимостями через npm, до сборки и упаковки ресурсов с использованием инструментов вроде Webpack и Vite. Node.js также необходим для поддержания работы локального сервера и динамичной пересборки проекта в ответ на дополнения или изменения в коде.

Проекты, как наш, часто внедряют платформу Firebase от OpenAI. Firebase предлагает богатый арсенал инструментов, начиная от хранилищ файлов и статистики сайта, до аналитики сбоев, механизмов баз данных, хостинга и аутентификации пользователей. Бесплатный план Firebase предоставляет все нужные инструменты для нашего проекта, с возможностью перехода на расширенные платные функции в будущем [8].

3.3 Выбор и описание средств работы с базой данных

В мире документно-ориентированных баз данных стоит выделить три фаворита, завоевавших доверие среди программистов.

MongoDB, благодаря её универсальности, высокой производительности и простоте в эксплуатации, стала лидером среди CouchDB и Firestore. Широкое применение и активное сообщество пользователей делают её привлекательным выбором для различных приложений.

Firestore, являющаяся составной частью Firebase, зарекомендовала себя как решение для интерактивных веб-систем, где критично мгновенное отображение свежих данных благодаря своим мощным real-time возможностям и прочности связи данных между устройствами пользователей и серверами [5].

CouchDB, продемонстрировала свою эффективность для сценариев, требующих высокопроизводительного доступа и расширяемости систем: её модель, основанная на дистрибуции и динамике структуры данных, позволяет оперативно и гибко управлять множеством документов в формате JSON.

Целесообразность использования каждой из упомянутых БД лежит в их специфических атрибутах, применимых к различным задачам и приложениям. Для агентства по подбору персонала, потребность в быстром обновлении информации и лёгкости в использовании указывает на Firestore как предпочтительное решение.

3.4 Реализация веб-представительства

Достигнув понимания о ключевых инструментах, можно переходить непосредственно к процессу проектирования.

Сразу после инициализации, для которой применима команда `prx create-react-app`, система React автоматически формирует начальную архитектуру, что демонстрируется в рисунке 18.

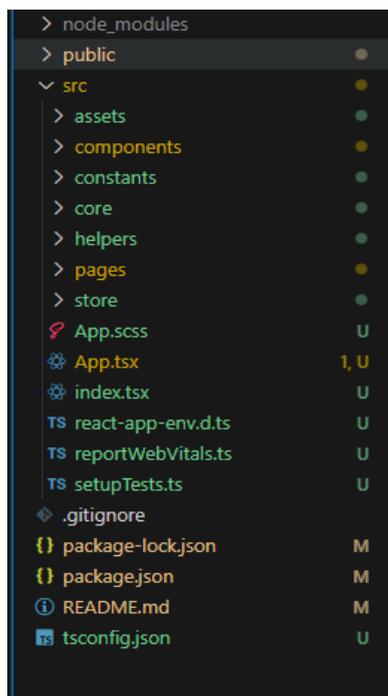


Рисунок 18 – Структура проекта

Проектная архитектура разделяется на две ключевые директории, `assets` и `components`. Каталог `assets` является хранилищем статических данных, содержа в себе CSS-код, шрифты, а также графические материалы. Директория `components` собирает в себе модульные блоки – основные строительные элементы веб-страниц, упрощая и оптимизируя процесс разработки за счёт следующих преимуществ:

- возможность повторного применения компонентов для различных страниц, что эффективно сокращает дублирование кода и делает структуру проекта менее запутанной;
- модулярный подход, где каждая секция веб-сайта обусловлена отдельным блоком, способствующим более ясной организации кода;
- удобство внесения изменений: модификация одного компонента происходит без необходимости обновления множества файлов, обеспечивая экономию времени, а также предотвращение в программном коде возникновения дефектов.

Специализированный каталог `core` занимается диспозицией настроек программы, где `stores` ответственен за конфигурацию и данные, с `Zustand` в качестве управляющей системы хранения данных. Директория `pages` исполняет роль контейнера для кода веб-страниц.

Центральный элемент программы, файл `App.tsx`, интегрирует в себе основные компоненты и логическую обработку.

Точка входа, или `index.js`, предназначен для конфигурирования и запуска программы [22]. Файл `tsconfig.json` задействован для тонкой настройки рабочего окружения TypeScript, включая маршрутизацию к модулям и определение базового каталога проекта. Упомянутый `package.json` набран метаданными проекта, перечнем зависимостей и скриптами для среды Node.js.

Под директорией `pages` расположены подкаталоги с файлами формата `*.tsx`, предназначенными для базовых страниц сайта, демонстрируемых на рисунке 19.

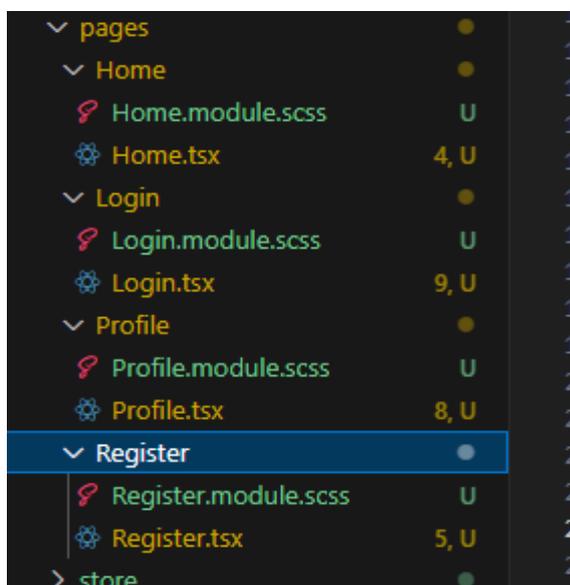


Рисунок 19 – Файлы страниц сайта

Структура главной страницы сайта, код которой представлен на рисунке 20, мы организовали в два блока:

- основная секция, интегрирующая механизм авторизации на сайте и функционал поиска по различным предложениям;
- секция контента, где пользователю представлен перечень доступных предложений вместе с ключевой информацией о каждом из них.

```
}, []);  
  
return (  
  <div className={styles["Home"]} >  
    <div className={styles["Home-Main"]} >  
      <Container className={styles["Home-Main-Container"]} >...  
    </Container >  
  </div >  
  
  <Box className={styles["Home-Content"]} >  
    <Container > ...  
  </Container >  
</Box >  
  
  <div id="detail" >  
    <Outlet />  
  </div >  
</div >  
)  
};  
}  
  
export default Home;
```

Рисунок 20 – Код главной страницы

Объекты, соответствующие вакансиям или резюме, извлекаются из базы данных и группируются в массив для удобства обработки. Наследованное поведение данных подразумевает их передачу от старших компонентов к дочерним через атрибут `props`. Однако проблема избыточного кодирования может быть решена применением средств управления состоянием.

В основе лежит функция `create`, предложенная пакетом `Zustand`. Эта функция упрощает процесс формирования единого хранилища данных, доступного во всём приложении. Это позволяет избежать трудностей, связанных с пропс-드릴лингом – излишней передачей свойств через компоненты. Основной запрос изображен на рисунке 21.

```

export const useProposalStore = create<IUseProposalStore>((set, get) => ({
  proposals: [],
  search: "",
  isLoading: false,

  setSearch: (value: string) => set({ search: value }),

  getProposals: async () => {
    try {
      set({ isLoading: true });

      const data = await axios
        .get<IProposals[]>(BASE_URL + "/proposals")
        .then((response) => response.data);

      if (data?.length) set({ proposals: data });
    } catch {
    } finally {
      set({ isLoading: false });
    }
  },
});

```

Рисунок 21 – Код получения списков предложений

Чтобы облегчить процесс навигации на веб-странице, в серверном окружении React используется встроенный пакет React Router. Этот инструмент полезен и способствует эффективному управлению переходами между различными разделами приложения. Избежание сложности ручного кодирования навигационной структуры достигается за счет предоставления готовых к применению решений, специфичных для многочисленных операций маршрутизации.

Задача создания навигационной системы включает несколько шагов. Подготовка файла роутера, представленного на рисунке 22, импортирование необходимых компонентов и создание объекта роутера.

```

const router = createBrowserRouter([
  {
    path: "/",
    element: <Home />,
    errorElement: <>404 Not found</>,
    children: [
      {
        path: "login",
        element: <Login />,
      },
      {
        path: "register",
        element: <Register />,
      },
      {
        path: "profile",
        element: <Profile />,
      },
    ],
  },
], {});

```

Рисунок 22 – Код навигации по приложению

Этот процесс не требует значительных временных затрат и может быть осуществлен даже разработчиками, лишь начинающими знакомство с React Router, благодаря наличию понятной документации.

Интегрировав компонент Dialog из материал-пакета material-ui, мы обеспечили создание полноэкранных всплывающих окон для профильных страниц, что позволяет избежать дублирования исходного кода. Благодаря этому, скорость и эффективность разработки интерактивных элементов значительно возросли. Как видно на примере, представленном на рисунке 23, встроенные стили и функциональные возможности компонента оказались достаточными для данной задачи.

```

return (
  <Dialog
    fullscreen
    keepMounted
    open={true}
    onClose={handleClose}
    TransitionComponent={Transition}
  >
    <AppBar sx={{ position: "relative" }}>
      <Toolbar>
        <IconButton
          edge="start"
          color="inherit"
          onClick={handleClose}
          aria-label="close"
        >
          <CloseIcon />
        </IconButton>
        <Box sx={{ ml: 5 }}>
          <Typography variant="h5" component="h5" sx={{ fontWeight: 500 }}>
            Профиль
          </Typography>
        </Box>
      </Toolbar>
    </AppBar>
    <Box className={styles["Profile"]} >...
  </Box>
</Dialog>
);

```

Рисунок 23 – Код всплывающего окна «Профиль»

Язык Go с пакетом `firestore` служит инструментарием для работы с базами данных в Cloud Firestore. Разработка собственных инструментов не требуется, поскольку готовые библиотеки легко интегрируются с нашими системами. После успешного импорта осуществляется написание кода, задача которого заключается в обработке данных. Операция извлечения коллекций данных из Cloud Firestore характеризуется своей прямоотой – необходимо указать желаемую коллекцию, чтобы получить полный перечень элементов. Следующий этап включает отправка этих данных на клиент, где они представляются в виде JSON. Код данного блока представлен на рисунке 24.

```

func (*proposalsRepo) FindAllProposals() ([]entity.Proposal, error) {
    client, ctx := utils.InitFirestore()

    iter := client.Collection("proposals").Documents(ctx)

    var proposals []entity.Proposal

    for {
        doc, err := iter.Next()

        if err != nil {
            break
        }

        proposal := entity.Proposal{
            Id:          doc.Data()["id"].(string),
            User_id:      doc.Data()["user_id"].(string),
            Amount:      doc.Data()["amount"].(string),
            Date:        doc.Data()["date"].(time.Time),
            Type:        doc.Data()["type"].(string),
            Description: doc.Data()["description"].(string),
            Title:      doc.Data()["title"].(string),
            Status:     doc.Data()["status"].(string),
        }

        var responses []entity.ResponseProposal

        itera := client.Collection("responses").Where("proposal_id", "=", proposal.Id).Documents(ctx)

        for {
            doc, err := itera.Next()

            if err != nil {
                break
            }

            response := entity.ResponseProposal{
                Id:          doc.Data()["id"].(string),
                User_id:      doc.Data()["user_id"].(string),
                Proposal_id: doc.Data()["proposal_id"].(string),
                Description: doc.Data()["description"].(string),
                Status:     doc.Data()["status"].(string),
            }
        }
    }
}

```

Рисунок 24 – Код получения списка предложений из БД

Ниже, на рисунке 25, представлен скриншот документа, демонстрирующее одно из отдельных предложений из базы данных.

рассмотрение удобства интерфейса через практическое взаимодействие потребителя.

Для достижения полноты тестирования, составляется детальный план, охватывающий стратегию и методологию исследования, обозначая применяемые инструменты и последовательность действий. Ключевые задачи, вынесенные на испытание, включают:

- функциональное исследование для гарантирования, что все элементы веб-сайта исполняют возложенные на них функции;
- конфигурационное исследование, целью которого является подтверждение неизменности качества отображения на разнообразных платформах;
- нагрузочное тестирование, основой которого является проверка работоспособности системы при высоких нагрузках;
- тестирование реальными пользователями для оценки комфортности и интуитивности использования.

В ходе проведенного исследования были получены определенные результаты, которые мы и представим далее. На рисунке 26 изображен основной сегмент домашней страницы вместе с отображаемым интерфейсом подтверждения личности пользователя.

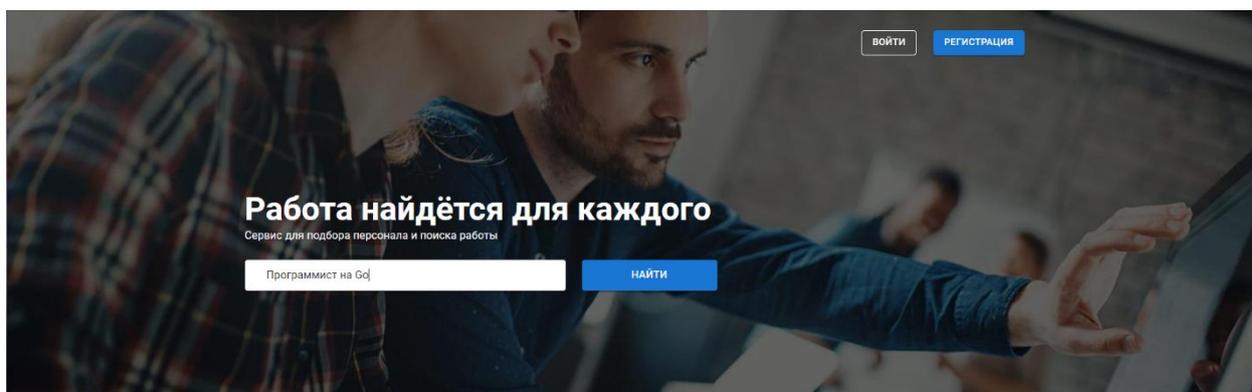


Рисунок 26 – Скриншот главного блока домашней страницы

На рисунке 27 показана раздел проекта, на которой представлен список предложений, доступных к рассмотрению.

Предложения для вас

Вакансия
07.10.2024

Разработчик на TypeScript
350000 Р

В компанию по продуктовой разработке для банка требуется разработчик со знанием программирования на TypeScript. Зарплата высокая, удобный график и дружный коллектив

Резюме
07.10.2024

Программист на Go
200000-250000 Р

5 лет занимаюсь программированием на Golang. Работал в крупных компаниях г. Москвы, имею опыт в разработке с использованием микросервисной архитектуры

Рисунок 27 – Скриншот главного блока домашней страницы

В дополнение, информационные модули, обратная связь и актуальные новости, заимствованные из внешних ресурсов, представлены на главной странице веб-портала. Данный раздел изображен на рисунке 28.

Вам нужна помощь?

Напишите телефон, чтобы работодатели могли предложить вам работу или сотрудников

Телефон

ОТПРАВИТЬ

Нажимая «Отправить», вы подтверждаете, что полностью принимаете условия Соглашения об оказании услуг по содействию в трудоустройстве и ознакомились с политикой конфиденциальности

Новости и статьи

 <p>Отпуск в 2025 году: когда брать выгоднее?</p>	 <p>Работа в Астрахани: подборка вакансий</p>
 <p>Рынок труда начинающих синих воротничков</p>	 <p>Судостроение: итоги 7 месяцев 2024 года</p>
 <p>Как составить резюме: образец</p>	 <p>Пять профессий с «быстрым входом»</p>
 <p>СТД «Петрович»: Компания, в которой комфортно работать</p>	 <p>Банки.ру: аналитика, динамика и карьера «в тапочках»</p>

Работа в России

Работа составляет большую часть жизни почти каждого из нас. Но ничто не вечно: случается, что однажды приходится менять место работы и с головой погружаться в поиски вакансий — хочется ведь найти хорошую альтернативу текущей должности.

Рисунок 28 – Скриншот блока с информацией на «Домашней странице»

Пользовательское взаимодействие с нашим веб-ресурсом начинается с инициирования сессии через активацию интерфейса авторизации, по клику на кнопку «Войти». При запросе запускается окно для авторизации. Данный интерфейс изображен рисунке 29.

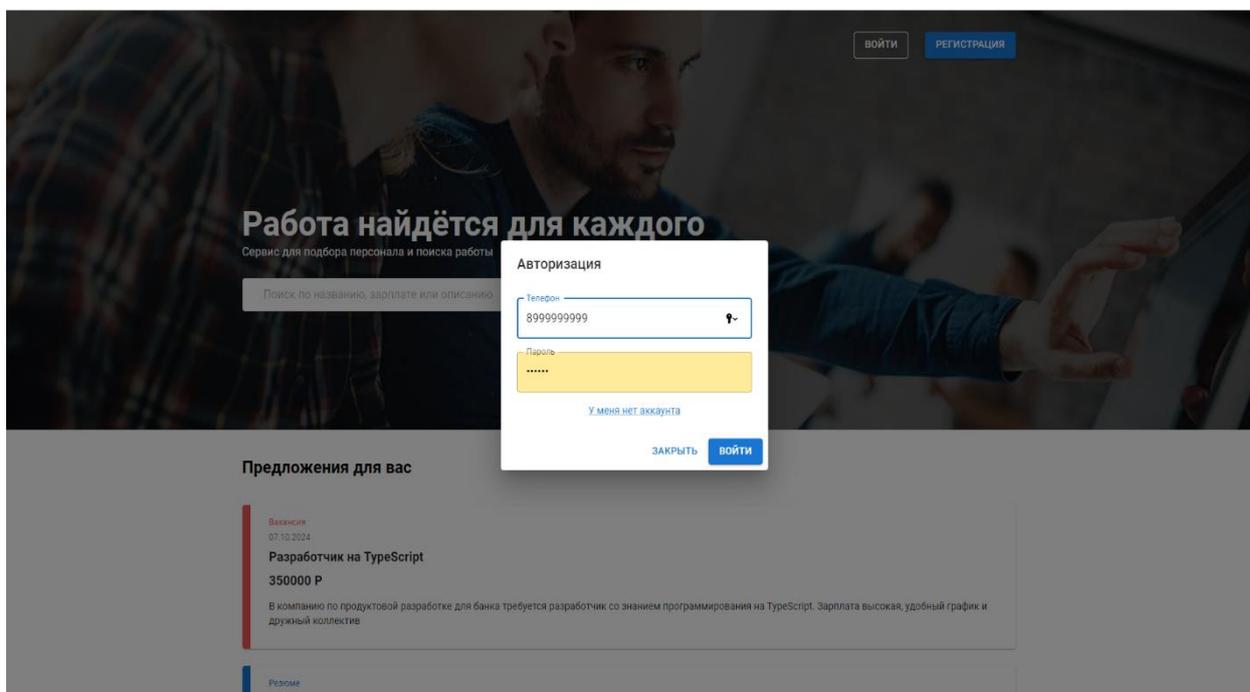


Рисунок 29 – Скриншот блока авторизации

При попытке зарегистрироваться в системе, необходимо инициировать процесс, при помощи клика на кнопку, обозначенную как «Регистрация», либо, при нахождении на странице, где происходит авторизация, выбрать гипертекстовую надпись, сообщающую об отсутствии учетной записи. Визуализация процедуры регистрации представлена на рисунке 30.

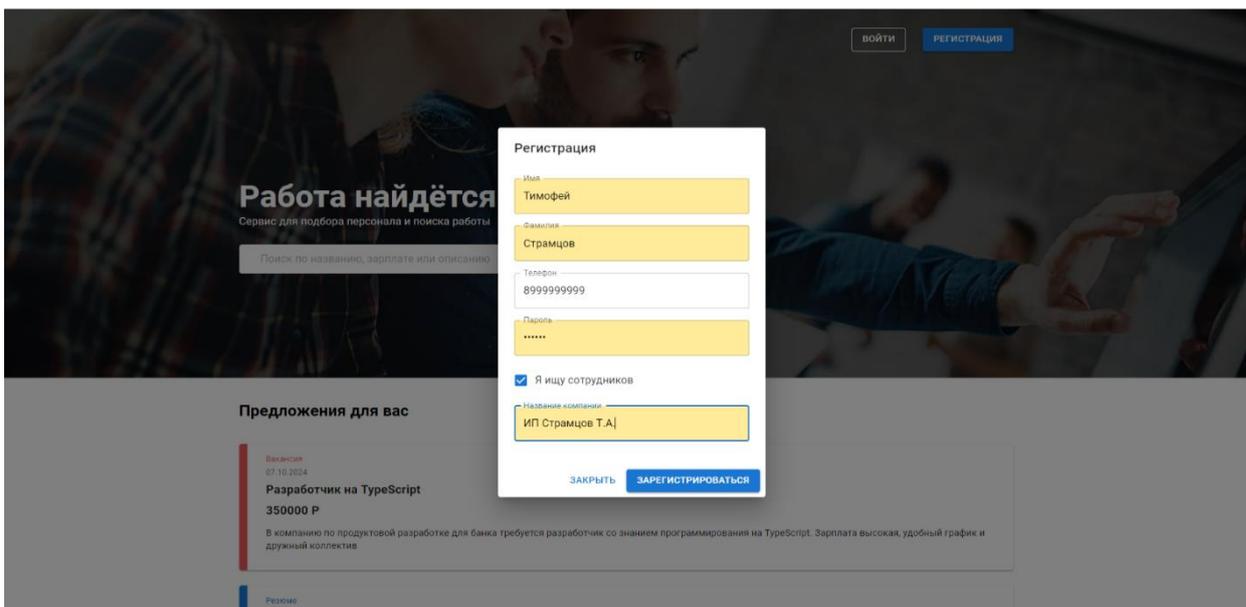


Рисунок 30 – Скриншот блока Регистрации

Ошибка регистрации может проявиться в случае, когда учетная запись пользователя предварительно создана. Инцидент с аналогичным типом сбоя представлен на рисунке 31.

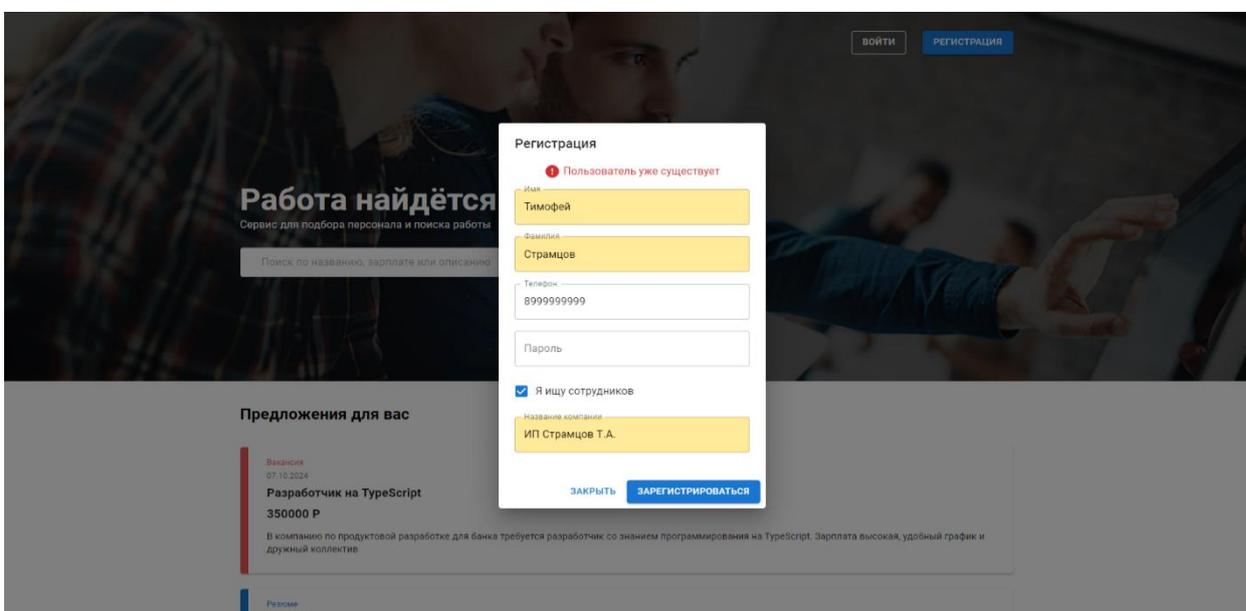


Рисунок 31 – Скриншот ошибки при регистрации

При успешной аутентификации, учётная запись пользователя активируется, предоставляя доступ к интерфейсу личного кабинета. В этом

разделе он может инициировать составление своего профессионального резюме или разместить предложение. Помимо этого, имеется опция просмотра поступающих предложений. Визуальное воплощение маршрутизации по portalу представлено на графическом материале под номером 32.

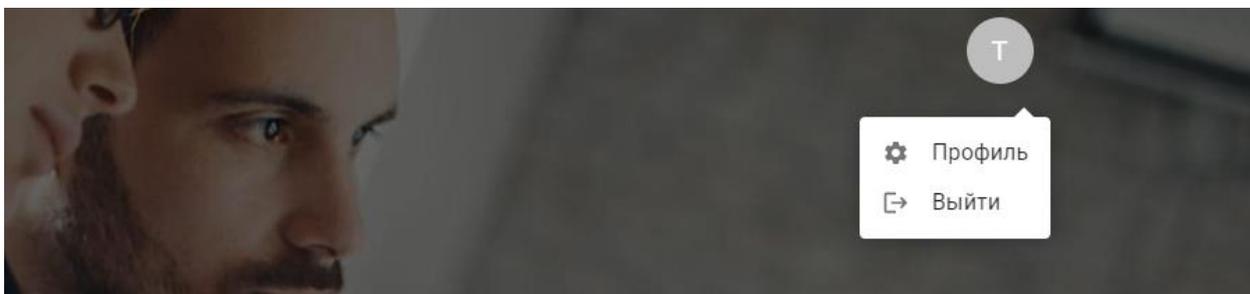


Рисунок 32 – Скриншот блока навигации после прохождения регистрации

На рисунке 33 демонстрируется блок страницы «Профиль», где представлена информация для ознакомления, касающаяся пользователя, а также предусмотрена возможность внесения собственного предложения.

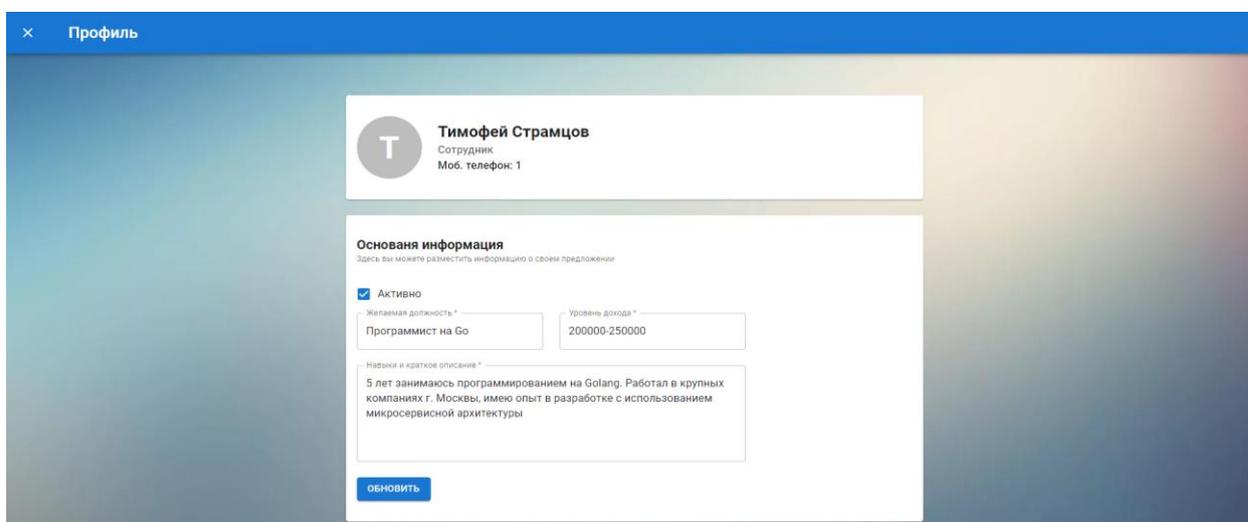


Рисунок 33 – Скриншот блока «Профиля»

Имеются две отдельные секции, обозначенные как «Мои отклики» и «Мои приглашения», размещённые на странице личного профиля. Участники платформы могут обдумать и решить, следует ли принять предложения о

работе либо же отказаться от них, а также оценить текущее состояние своих ранее отправленных откликов на вакансии. Данный раздел демонстрируется на рисунке 34.

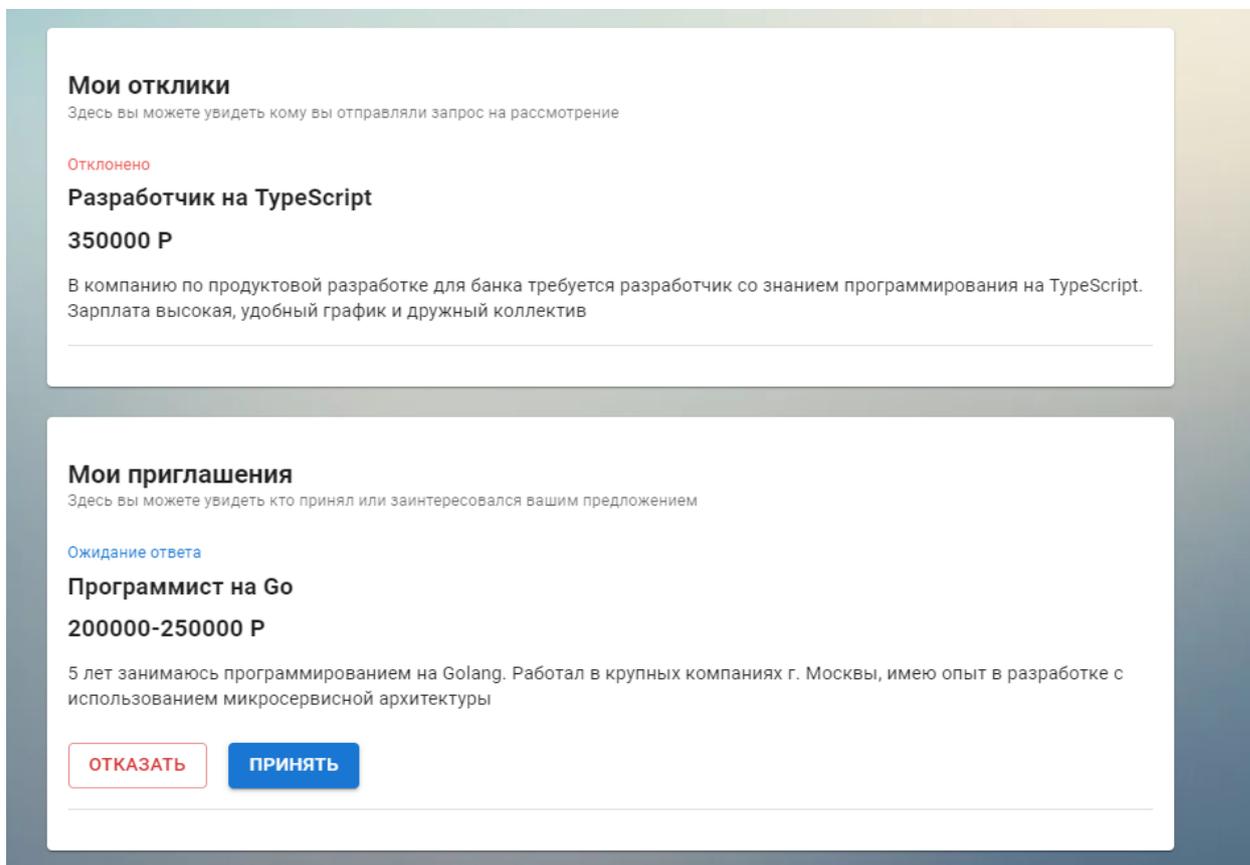


Рисунок 34 – Скриншот блока «Мои отклики» и «Мои приглашения»

Применение Container из комплекта инструментов Material-UI освобождает от необходимости дополнительных манипуляций для оптимизации веб-ресурса на мобильных платформах. Благодаря своей адаптивной ширине, данный элемент обеспечивает надлежащее визуальное представление компонентов на широком спектре гаджетов. Наглядная демонстрация интерфейса при использовании на смартфоне представлена на рисунке 35.

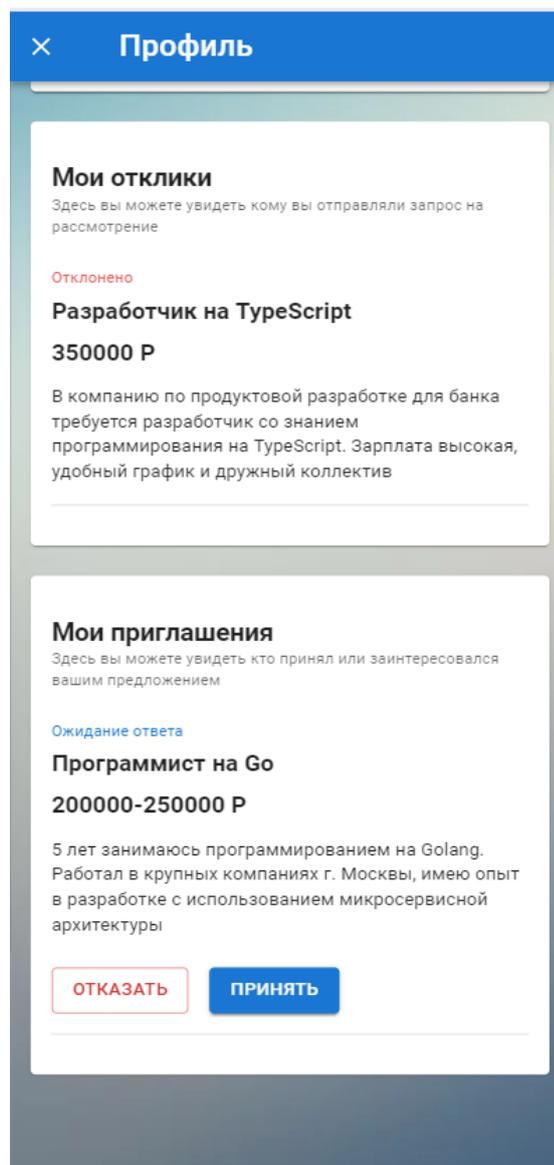


Рисунок 35 – Скриншот блока «Мои отклики» и «Мои приглашения» на мобильном устройстве

Исследование функциональности веб-ресурса выявило его соответствие оговорённым спецификациям. Учитывая тестируемый результат, обнаружена адекватность отображения интерфейса на множестве устройств и программ для просмотра веб-страниц.

Выводы по главе 3

В этой главе мы завершили реализацию и тестирование нашего веб-приложения. Для разработки мы использовали Visual Studio Code, простой и бесплатный текстовый редактор, который обладает достаточным функционалом для быстрой и удобной работы.

Спектр технологических инструментов, задействованных в процессе разработки, включал несколько языков программирования и фреймворков, таких как HTML, CSS, JavaScript, React; библиотеки и инструменты как Material-UI, Firebase; а также инструменты серверной разработки, включая Go и Node.js.

Оценка качества работоспособности была проведена с помощью всестороннего тестирования, результаты которого свидетельствуют о безупречной функциональности ресурса на различных платформах, в том числе разнообразных браузерах и мобильных устройствах, что удостоверило его производительность, удобство использования и адаптивность к разным форматам экранов.

Таким образом, разработанное веб-приложение было успешно реализовано и протестировано, и теперь готово к использованию.

Заключение

В данной бакалаврской работе была разработана веб-представительская система кадрового агентства. В ходе работы был проведен анализ предметной области, выявлены потребности целевой аудитории и особенности веб-представительств кадровых агентств, проведен анализ аналоговых веб-представительств. На основе полученных данных были разработаны функциональные и нефункциональные требования к системе.

В процессе проектирования было проведено логическое моделирование данных и функциональности системы, разработаны диаграммы прецедентов, деятельности и классов.

Также была создана логическая модель данных и разработана внутренняя и внешняя структуры сайта.

Для реализации проекта были выбраны современные и эффективные технологии, такие как React и Go, стейт-менеджер Zustand, библиотека компонентов Material-UI, а также документо-ориентированная база данных Firestore.

Выбранные инструменты позволили реализовать весь необходимый функционал, включая личный кабинет, создание и редактирование предложений, отображение списка всех предложений и информации о вакансиях, предложениях и откликах, а также адаптивный дизайн сайта для различных устройств.

Тщательное мануальное тестирование системы, при помощи инструментов: браузера и панели разработчика, подтвердило ее соответствие заявленным требованиям.

Разработанная система веб-представительства является современным и эффективным инструментом для привлечения новых пользователей, повышения узнаваемости бренда и, как следствие, роста прибыли кадрового агентства.

Список используемой литературы и используемых источников

1. Баканов, А. С. Проектирование пользовательского интерфейса: эргономический подход / А. С. Баканов, А. А. Обознов. – 2-е изд. – Москва: Издательство «Институт психологии РАН», 2019. – 184 с.
2. Белов, В. В. Проектирование информационных систем [Текст] / В. В. Белов, В. И. Чистякова – М.: Академия, 2013. – 352 с.
3. Варзунов, А. В. Анализ и управление бизнес-процессами: учебное пособие. Санкт-Петербург: Университет ИТМО. 114 с. [Электронный ресурс]. URL: <https://www.iprbookshop.ru/65772.html> (дата обращения: 10.09.2024).
4. Дакетт, Д. HTML и CSS. Разработка и дизайн веб-сайтов / Д. Дакетт. - М.: Эксмо, 2018. - 208 с.
5. Документация Firebase [Электронный ресурс] – URL: <https://firebase.google.com/docs?hl=en> (дата обращения: 20.09.2024).
6. Документация Firebase Hosting [Электронный ресурс] – URL: <https://firebase.google.com/docs/hosting?hl=ru> (дата обращения: 20.09.2024).
7. Документация React [Электронный ресурс] – URL: <https://ru.react.js.org/docs/getting-started.html> (дата обращения: 20.09.2024).
8. Документация Material-ui [Электронный ресурс] – URL: <https://mui.com/material-ui/getting-started/> (дата обращения: 20.09.2024).
9. ИНТУИТ [Электронный ресурс] – URL: <https://intuit.ru/studies/courses/2188/174/lecture/4714?page=2> (дата обращения: 05.09.2024).
10. Кадровое агентство SuperJob [Электронный ресурс] – URL: <https://www.superjob.ru/> (дата обращения: 20.09.2024).
11. Кадровое агентство АСПЕКТ [Электронный ресурс] – URL: <https://www.jobaspect.ru/> (дата обращения: 20.09.2024).
12. Кадровое агентство КАДРОФФ [Электронный ресурс] – URL: <https://kadroffgroup.ru/> (дата обращения: 20.09.2024).

13. Котлинский, С. В. Разработка моделей предметной области автоматизации – СПб.: Лань, 2021. – 412 с.
14. Круг С. Не заставляйте меня думать. Веб–юзабилити и здравый смысл. – Эксмо, 2014. 250 с.
15. Лаврищева, Е. М. Программная инженерия. Парадигмы, Технологии и CASE–средства. Учебник для вузов [Текст] / Е. М. Лаврищева М.: Юрайт, 2017. – 280 с.
16. Новиков, Б. А. Основы технологий баз данных. Учебное пособие [Текст] / Б. А. Новиков, Е. А. Горшкова – М.: ДМК Пресс, 2019. – 240 с.
17. Правильная структура сайта: классические составляющие и фишки // Envуbox блог. – [Б. м.], 2015–2023. URL: <https://envybox.io/blog/strukturajasajta/> (дата обращения: 25.08.2024).
18. Советов, Б. Я., Базы данных. Учебник. 3–изд. [Текст] / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской – М.: Юрайт, 2018. – 420 с.
19. Флегонтов, А. В. Моделирование информационных систем [Текст] / А. В. Флегонтов, И. Ю. Матюшичев – М.: Лань, 2018. – 112 с.
20. Introduction to Database Systems, 8th Edition. / C.J. Date – Boston: Pearson/Addison Wesley 2005. – 1328 с.
21. Learning PHP, MySQL, JavaScript, CSS & HTML5, Third Edition. / Robin Nixon – O'Reilly Media, 2014 – 729 с.
22. Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics 5th Edition / Jennifer Robbins – O'Reilly Media, 2018. – 811с.
23. Responsive Web Design with HTML5 and CSS3 Illustrated Edition. /Ben Frain – Packt Publishing, 2012. – 324 с.
24. Visual Paradigm [Электронный ресурс]. URL: <https://online.visualparadigm.com/> (дата обращения: 20.09.2024).
25. You Don't Know JS Yet (book series) - 2nd Edition. / Kyle Simpson – GetiPub & Leanpub, 2020 – 145 с.