Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Омский государственный технический университет»

| Факультет | информационных технологий и компьютерных систем |
| --- | --- |

| Кафедра | Математические методы и информационные технологии в экономике |
| --- | --- |

| Направление подготовки (специальность) | 09.04.02 Информационные |
| --- | --- |
| системы и технологии | |
| код, наименование | |

|  | ДОПУСКАЕТСЯ К ЗАЩИТЕ: | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Заведующий кафедрой | | | | | |  | | |
|  |  | | | | | подпись | | | |
|  | Бахмутский Ю.А. | | | | | | | | |
|  | ФИО | | | | | | | | |
|  | « |  | » |  | 20 | | | 25 | г. |

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

| Разработка прототипа онлайн-сервиса |
| --- |
| для организации стажировок студентов |
|  |
| тема работы |

| Шифр | МР-2068999-ИСТм3111-00.00.000. ПЗ |
| --- | --- |

| Студент | ИСТм-231 |  | Коваленко Е.А. |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | группа |  | ФИО |  | подпись |  | дата |

| Руководитель |  | | Голунова А.С. | |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  | | ФИО |  | подпись |  | дата |
| Нормоконтролер |  | | Анашкина Н.А. | |  |  |  |  |
|  | |  | | ФИО |  | подпись |  | дата |

| Омск | 2025 |
| --- | --- |

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Омский государственный технический университет»

|  |  |  | |  | УТВЕРЖДАЮ: | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | |  | Заведующий кафедрой | | |  |
|  |  | | |  |  |  | (подпись) | |
|  | | |  | |  | | | |
|  | | |  | | (ФИО) | | | |

|  |  |  |  |  |  |  | « |  | » |  | 20 |  | г. |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

**ЗАДАНИЕ**

**на выполнение выпускной квалификационной работы**

| Обучающийся | | Коваленко Евгений Анатольевич | | | | | Группа | | ИСТм-231 | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | (фамилия, имя, отчество) | | | | |  | |  | |
|  | |  | | | | |  | |  | |
| Факультет (институт) | | | Факультет информационных технологий и компьютерных систем | | | | | | | |
|  | | |  | | | | | | | |
| Кафедра | Математические методы и информационные технологии в экономике | | | | | | | | | |
|  |  | | | | | | | | | |
| Направление подготовки (специальность) | | | | | | 09.04.02 Информационные системы и технологии | | | | |
|  | | | | | | | | | | |
| Направленность (специализация) | | | | | Руководство разработкой цифровых решений | | | | |  |
|  | | | | | |  | | | | |
| Квалификация | | Магистр | | | | | |  | | |
|  | |  | | | | | |  | | |
|  | | | | | |  | | | | |
| Наименование темы ВКР | | | | Разработка прототипа онлайн-сервиса для организации стажировок | | | | | | |
| студентов | | | | | | | | | | |

| Срок сдачи обучающимся законченной ВКР | « |  | » |  | 20 |  | г. |
| --- | --- | --- | --- | --- | --- | --- | --- |

| Исходные данные к работе | Техническое задание на разработку онлайн-сервиса | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| Содержание ВКР (перечень подлежащих разработке разделов) | | | Введение, Глава 1 Основы | | | |  |
| Разработки онлайн-сервиса, Глава 2 Проектирование и функциональная концепция | | | | | |  |
| онлайн-сервиса, Глава 3 Реализация онлайн-сервиса, Заключение, Список использованных | | | | | | |
| источников | | | | | | |
| Перечень графического материала (с указанием обязательных чертежей | | | |  |
| и (или) иллюстративного материала) | | Презентация | | |

| Дата выдачи задания | « | 21 | » | января | 20 | 25 | г. |
| --- | --- | --- | --- | --- | --- | --- | --- |

| Руководитель |  |  | |  |  | | |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | (подпись) | |  | (ученая степень, ученое звание) | | |  | (ФИО) |
| Задание принято к исполнению | | |  | | |  |  | | |
|  | | | (подпись обучающегося) | | |  | (ФИО) | | |

**Аннотация**

Пояснительная записка к ВКР содержит: 92 страницы; 28 иллюстраций; 5 таблиц; 27 источников.

Перечень ключевых слов: онлайн-сервис, цифровая платформа, стажировка, трудоустройство студентов, фильтрация вакансий, отклики, сопровождение практики.

Цифровизация профессиональной ориентации и трудоустройства обучающихся является актуальной задачей в условиях модернизации образования и требований рынка труда. Современные студенты сталкиваются с проблемами поиска релевантных стажировок и практик, отсутствием структурированных источников информации и недостаточной поддержкой на этапе взаимодействия с работодателями. В этих условиях разработка специализированного цифрового онлайн-сервиса, ориентированного на потребности студентов и молодых специалистов, приобретает особую значимость.

Целью магистерской работы является разработка прототипа онлайн сервиса, оптимизирующей процесс поиска и подачи заявок на стажировки и практики для студентов

Задачи ВКР:

* + Изучить предметную область и выявить текущие проблемы трудоустройства студентов;
  + Сформировать функциональные и нефункциональные требования к системе;
  + Разработать архитектуру и модель базы данных;
  + Реализовать клиентскую и серверную части приложения;

**Содержание**

[Введение 5](#_zhdb2ln0w24w)

[1. Основы разработки онлайн-сервиса 8](#_l4d36mp0voko)

[1.1 Теоретические основы разработки онлайн-сервисов 8](#_vvu9n8gkjaep)

[1.2 Анализ аналогов 10](#_t4dobfnxs5pa)

[1.3 Анализ целевой аудитории 21](#_c70kn6xbim)

[1.4 Требования к цифровой системе 24](#_rgaheqgc4t78)

[1.5 Подход к разработке: принципы и этапы 28](#_g3tl44jkisx)

[1.6 Выбор инструментов для разработки 31](#_4fhjppvblz2z)

[1.7 Выводы по первой главе 39](#_h3ptdcf5cp2b)

[2. Проектирование и функциональная концепция онлайн-сервиса трудоустройства студентов 41](#_5v5gclf335ia)

[2.1 Концепция онлайн-сервиса и ее ключевые функции 41](#_67detybvdee5)

[2.2 Модель данных и организация хранения информации 50](#_bqsj1pl83lol)

[2.3 Система рекомендаций и персонализации 63](#_as10id4c5osn)

[2.4 Выводы по второй главе 65](#_udiaih6c0sxn)

[3. Реализация онлайн-сервиса 67](#_b1mi1vcoqs9)

[3.1 Стилистические решения онлайн-сервиса 67](#_or4cd84cwnxj)

[3.2 Реализация серверной логики и API 69](#_lt3zfqdoojyp)

[3.3 Реализация клиентской части приложения 73](#_9dppr6311dzn)

[3.4 Выводы по третьей главе 87](#_sa0forb4mo4)

[Заключение 88](#_v9pbsq674r6n)

[Список используемых источников 90](#_e9wg4fvpl7r6)

**Введение**

В условиях развития информационных технологий и цифровой трансформации образования задача интеграции образовательных учреждений с рынком труда приобретает особое значение [6]. Также параллельно с этим растет потребность компаний в квалифицированных специалистах, что делает актуальным создание онлайн-сервисов, способствующих эффективному взаимодействию между работодателями и студентами. Важным моментом является внедрение решений, позволяющих студентам не только находить вакансии, но и проходить тестирования, участвовать в дистанционных собеседованиях и получать персонализированные рекомендации по развитию карьеры.

Создание единого онлайн сервиса для трудоустройства, которая бы объединяла студентов, университеты и работодателей, становится необходимостью для повышения качества подготовки работников и улучшения интеграции студентов в рынок труда [2]. Современные цифровые решения, такие как сервисы для размещения вакансий с функциями фильтрации по специализациям и навыкам, способны значительно улучшить процесс поиска практик и стажировок для студентов, а также помогают упростить поиск подходящих кандидатов для компаний.

Основной проблемой, которую решает исследование, является недостаточная интеграция учебного процесса с реальными потребностями рынка труда. Трудности в поиске мест стажировок и практик, а также ограниченная коммуникация между студентами и работодателями усложняют процесс трудоустройства. В этом контексте разработка онлайн-сервиса, который объединил образовательные учреждения, студентов и работодателей, становится важным шагом для улучшения этого процесса.

Целью настоящего исследования является разработка прототипа онлайн сервиса, оптимизирующей процесс поиска и подачи заявок на стажировки и практики для студентов. Для достижения этой цели необходимо решить несколько задач:

* + Изучить предметную область и выявить текущие проблемы трудоустройства студентов;
  + Сформировать функциональные и нефункциональные требования к системе;
  + Разработать архитектуру и модель базы данных;
  + Реализовать клиентскую и серверную части приложения;

Ожидаемым результатом работы станет создание прототипа онлайн-сервиса, который упростит процесс поиска стажировок и практик для студентов и улучшит взаимодействие между университетами и работодателями.

Вопросы цифровизации образования и трудоустройства студентов активно исследуются в современных работах, таких как исследования по использованию онлайн-сервиса для поиска работы, а также в области информационных технологий в образовании [7]. Однако на текущий момент нет универсальной платформы, которая бы эффективно решала проблему интеграции студентов, университетов и работодателей, что подтверждает актуальность исследования.

Практическая значимость исследования заключается в том, что разработанный онлайн-сервис станет полезным инструментом для студентов, университетов и работодателей. Онлайн сервис позволит улучшить процесс поиска стажировок и практик, предоставляя студентам возможность напрямую взаимодействовать с работодателями, а работодателям — находить кандидатов с нужными навыками. Университеты смогут адаптировать свои образовательные программы, ориентируясь на реальные требования рынка труда.

Таким образом, создание специализированного онлайн-сервиса для трудоустройства студентов является важным шагом в цифровизации образовательного процесса и интеграции вузов в современные реалии рынка труда. Разработанный продукт обеспечит удобную платформу для взаимодействия студентов, образовательных учреждений и работодателей, а также позволит автоматизировать ключевые процессы, связанные с поиском и процессом прохождения практики и стажировок. Внедрение современных технологий, таких как автоматический подбор вакансий, онлайн-тестирование и система, предназначенная для общения между представителями компаний и студентами, поможет улучшить взаимодействие между всеми участниками образовательного, а также повысит шансы студентов на успешное трудоустройство.

# 1. Основы разработки онлайн-сервиса

## 1.1 Теоретические основы разработки онлайн-сервисов

Онлайн-сервисы – это сайты, которые предоставляют всевозможные услуги, что значительно облегчает работу и позволяет существенно сэкономить время [12]. Такие системы отличаются высокой доступностью, интерактивностью и независимостью от платформы конечного пользователя.

Разработка онлайн-сервиса представляет собой комплексный процесс, включающий проектирование архитектуры, реализацию функциональности, обеспечение надежности, масштабируемость и безопасность. В современной практике применяются различные архитектурные модели, которые представляют собой план, определяющий структуру и взаимодействие компонентов между собой в онлайн-сервисе.

Наиболее подходящей моделью для онлайн-сервисов является клиент-серверная архитектура. Данный подход обеспечивает разделение ответственности, что упрощает масштабирование и сопровождение системы на всем этапе ее жизненного цикла. Она представляет собой разделение системы на две части – клиентская (frontend) и серверная (backend).

Клиентская часть отвечает за отображение интерфейса и взаимосвязь с пользователем, но не имеет доступа к базе данных и основной бизнес-логике. Серверная часть отвечает за обработку данных, которые приходят после того, как пользователь ввел их в клиентской части. Также эта часть позволяет обрабатывать основную сложную бизнес логику, которая должна быть защищена от мошенников. Серверная часть отвечает за работу с базой данной, которая хранит в себе всю информацию, которая может потребоваться пользователю при взаимодействии с сервисом.

Для улучшения устойчивости и гибкости онлайн-сервиса также необходимо применять модульный подход, который помогает разделить блоки друг от друга для дальнейшего переиспользования и разделения бизнес-логики [13]. Каждый функциональный блок реализуется как независимый модуль, взаимодействующий с другими через интерфейс.

При разработке онлайн-сервиса необходимо учитывать множество требований, которые обеспечивают его масштабируемость и доступность [18]. Наиболее важными требованиями являются:

* Надежность – способность сервиса стабильно функционировать при различных нагрузках, даже в условиях сбоя. Это достигается за счет обработки исключений, логирования (записи действий пользователя и ответов системы на них) и применения отказоустойчивых архитектур;
* Безопасность – при работе с персональными и конфиденциальными данными обязательно требуется использование различных методов шифрования, разграничение прав доступа к различным элементам сервиса и защита API от злоумышленников;
* Масштабируемость – возможность расширения онлайн-сервиса, не ограничивая работу модулей, которые введены в эксплуатацию. Это достигается путем горизонтального или вертикального масштабирования компонентов;
* Юзабилити – простота и удобство использования интерфейса онлайн-сервиса. Важным аспектом является адаптивность под различные экраны и интуитивная навигация;
* Интеграция – возможность взаимодействия с внешними API, базами данных, платежными системами и облачными хранилищами.

Также важным аспектом при разработке онлайн-сервиса является понимание его жизненного цикла. Жизненный цикл программного обеспечения – период времени, который начинается с момента принятия решения о необходимости создания программного продукта и заканчивается в момент его полного изъятия из эксплуатации. Основными этапами жизненного цикла онлайн-сервиса являются:

1. Анализ требований – определяются задачи, целевая аудитория, функциональные и нефункциональные требования;
2. Проектирование архитектуры и базы данных – создаются схемы, описывающие взаимодействие компонентов и структуру хранения данных;
3. Разработка – программирование логики работы онлайн-сервиса, реализация интерфейса, настраивание взаимодействие с сервером;
4. Тестирование – проверка корректности выполнения сценариев, устойчивости системы и ее безопасности при работе с данными;
5. Развертывание – процесс, обеспечивающий доступность конечным пользователям работать с системой;
6. Сопровождение – исправление обнаруженных ошибок, добавление новых функций в сервис.

Все описанные пункт позволят создать онлайн-сервис, который будет соответствовать всем современным стандартам разработке, будет удобен конечному пользователю и команде, которая будет сопровождать проект.

## 1.2 Анализ аналогов

В условиях цифровизации труда и образования студенты и выпускники активно используют сайты или другие ресурсы, размещенные в интернете, для поиска стажировок и начала карьеры. Наиболее известные и широко используемые цифровые платформы в России — HeadHunter (hh.ru), SuperJob, Работа.ру и государственный портал «Работа в России» (trudvsem.ru). Также имеется платформа «Профстажировки 2.0», которая предоставляет больший функционал и позволяет проводить стажировки и практики непосредственно на самой платформе. Но если не обращать внимание на популярность и широкую функциональность, эти платформы имеют ряд ограничений, особенно в контексте нужд студентов и интеграции с образовательной системой.

Платформа HeadHunter (hh.ru) представляет собой одну из крупнейших цифровых систем поиска работы в России и СНГ. Она охватывает множество отраслей и предлагает миллионы вакансий по всей стране, а также предоставляет работодателям доступ к базе резюме. HeadHunter активно внедряет современные HR-технологии, такие как автоматическая оценка резюме, шаблоны откликов и инструменты для проведения онлайн-собеседований.

Для студентов на платформе предусмотрен специальный раздел «Начало карьеры», где публикуются предложения стажировок и вакансий без требований к опыту. Однако, несмотря на наличие такого раздела, его структура не адаптирована под образовательные цели. Отсутствует возможность фильтрации вакансий по учебной специальности, графику занятий или возможности прохождения практики. Вакансии в этом разделе часто имеют формальный характер и не содержат информации о согласовании с вузом.

Преимущества платформы HeadHunter:

1. Широкая география и база работодателей;
2. Возможность загрузки резюме и сопроводительных писем;
3. Наличие откликов, уведомлений и встроенных фильтров.

Недостатки:

1. Отсутствие образовательной интеграции с университетами;
2. Тет специфических фильтров по направлению подготовки студента;
3. Предложения не всегда ориентированы на студентов без опыта;
4. Низкая релевантность практик к учебным программам.

SuperJob также является одной из лидирующих платформ на рынке труда, предоставляя вакансии от крупных и средних компаний. Интерфейс платформы простой и интуитивно понятный, а регистрация быстрая, что делает её привлекательной для молодых пользователей. Платформа позиционируется как ориентированная на соискателей с различными уровнями подготовки, включая студентов.

Тем не менее, при анализе раздела с вакансиями для начинающих специалистов и стажеров становится очевидно, что SuperJob в большей степени ориентирован на выпускников, нежели на студентов младших курсов или тех, кто ищет производственную практику. Здесь также нет тесной привязки к вузовским программам, и отсутствует функционал для работодателей по учету прохождения практик.

Преимущества SuperJob:

1. Удобный интерфейс и простая регистрация;
2. Наличие фильтров по зарплате и графику;
3. Персонализированные подборки вакансий.

Недостатки:

1. Слабая ориентированность на студентов, не завершивших обучение;
2. Нет инструментов связи с вузом;
3. Ограниченные возможности по отслеживанию практик.

Работа.ру в последние годы активно развивает направление, связанное с поддержкой молодых специалистов. Согласно открытым данным, количество вакансий для студентов и стажёров на платформе с 2021 года выросло более чем в два раза. Платформа позволяет фильтровать вакансии по ключевым словам, региону и графику занятости, что делает её потенциально удобной для соискателей без опыта.

Однако, несмотря на позитивную динамику, структура платформы по-прежнему не адаптирована к специфике образовательного процесса. В ней отсутствуют инструменты для фиксации статуса прохождения практики, а вакансии зачастую дублируют предложения с других коммерческих ресурсов, не проходя дополнительной верификации. Это снижает доверие к информации, особенно у студентов, впервые ищущих место для стажировки.

Преимущества Работа.ру:

1. Активное развитие сегмента для студентов и выпускников;
2. Высокая скорость отклика работодателей;
3. Возможность публикации кратких резюме.

Недостатки:

1. Нет интеграции с вузами и сопровождения практик;
2. Отсутствие контроля качества вакансий;
3. Нет тематической фильтрации под образовательные направления.

Trudvsem.ru (портал «Работа в России») — это федеральная государственная информационная система, созданная Минтрудом РФ для содействия занятости населения. В отличие от коммерческих платформ, ресурс предоставляет бесплатный доступ к вакансиям, собранным от всех работодателей, зарегистрированных через службы занятости. Платформа также поддерживает размещение предложений о прохождении практики и стажировок, что делает ее потенциально полезной для студентов.

Ключевым преимуществом trudvsem.ru является достоверность информации, так как вакансии проходят государственную модерацию. Кроме того, платформа интегрирована с региональными центрами занятости и образовательными учреждениями в рамках некоторых программ поддержки молодежи. Однако такие механизмы работают далеко не во всех регионах и носят фрагментарный характер. Сама платформа слабо адаптирована к запросам студентов: интерфейс устаревший, поиск затруднен, отсутствует детализированная фильтрация и привязка к профилю образования.

Преимущества trudvsem.ru:

1. Государственная поддержка и верификация вакансий;
2. Бесплатность всех сервисов;
3. Наличие программ трудоустройства молодежи.

Недостатки:

1. Устаревший интерфейс и сложность в использовании;
2. Ограниченная адаптация под образовательные процессы;
3. Низкая активность работодателей по направлению студенческих практик.

Платформа «Профстажировки 2.0» является федеральным проектом, направленным на содействие прохождению студентам производственной практики в организациях различных отраслей экономики. Платформа ориентирована прежде всего на студентов старших курсов, обучающихся в вузах.

В отличие от коммерческих проектов, данная платформа ориентирована на прохождение производственным практик, в том числе с зачетом в университете. Работодатели публикуют задания, а студенты – откликаются на них, прикладывая резюме и ответы на поставленные вопросы. Платформа в основном способствует прохождению практик у крупных организаций и имеет ограниченную фильтрацию по параметрам.

Преимущества платформа «Профстажировки 2.0»:

1. Ориентация на прохождение учебной практики с возможностью зачета в вузе;
2. Поддержка со стороны государственных структур и крупных работодателей;
3. Централизованная система подачи заявок и получения обратной связи;
4. Ориентация на студентов без опыта работы;

Недостатки данной платформа:

1. Отсутствие гибкой фильтрации, по направлению подготовки, формату практики, региону и т.д.;
2. Ограниченное количество кейсов в определенных направлениях;
3. Механизм взаимодействия ограничен только выполнением кейсов, без возможности подавать отклики.

Сравнение российских аналогов между собой показано в таблице 1

Таблица 1 - Сравнение российских аналогов

| Критерий | HeadHunter | SuperJob | Rabota.ru | Trudvsem.ru | Профстажировки 2.0 | Разрабатываемая платформа |
| --- | --- | --- | --- | --- | --- | --- |
| Фокус на стажировках и практиках | 1 | 1 | 1 | 2 | 3 | 3 |
| Фильтрация по направлению подготовки | 1 | 1 | 2 | 2 | 1 | 3 |
| Интеграция с вузами | 1 | 1 | 1 | 2 | 3 | 3 |
| Поддержка студентов без опыта | 1 | 1 | 1 | 2 | 3 | 3 |
| Сопровождение процесса стажировки | 1 | 1 | 1 | 1 | 2 | 3 |
| Интерфейс, адаптированный под студентов | 1 | 1 | 1 | 2 | 2 | 2 |
| Система откликов и статусов | 3 | 3 | 3 | 2 | 1 | 3 |
| Модуль сообщений и общения | 3 | 1 | 1 | 1 | 1 | 3 |
| Система рекомендаций | 3 | 3 | 1 | 1 | 1 | 3 |
| Возможность загрузки сопроводительных писем | 3 | 3 | 3 | 1 | 1 | 3 |
| Публикация вакансий работодателями напрямую | 3 | 3 | 3 | 3 | 1 | 3 |

Сравнение отечественных цифровых решений в области трудоустройства студентов показывает, что ни одна из действующих платформ не охватывает в полной мере специфические потребности обучающихся, находящихся в процессе получения образования. Коммерческие сервисы, такие как HeadHunter, SuperJob и Rabota.ru, преимущественно ориентированы на поиск постоянной работы и соискателей с опытом, в то время как функциональность, связанная со стажировками и производственной практикой, реализована фрагментарно и носит скорее вспомогательный характер. Эти ресурсы не поддерживают согласование с вузами, не предоставляют механизмов сопровождения студентов и не адаптированы к их образовательному статусу.

Из представленных решений наиболее близким к нуждам студентов является государственный портал Trudvsem.ru и Профстажировки 2.0, которые предлагают отдельный раздел для молодежи и выпускников, ориентированный на первую занятость. Однако данные сервисы также не обеспечивает достаточной персонализации, гибкой фильтрации по направлению подготовки и цифрового сопровождения процесса стажировки.

Помимо российских решений, в мировой практике широко распространены специализированные цифровые платформы, ориентированные на студентов и молодых специалистов. Наиболее популярные из них — Handshake, LinkedIn Students, Graduateland и WayUp.

Handshake — это специализированная американская цифровая платформа, предназначенная для содействия трудоустройству студентов и выпускников вузов. Она активно используется университетами США и предоставляет доступ к вакансиям только для студентов, обучающихся в учебных заведениях, интегрированных с системой. Работодатели проходят предварительную модерацию, что позволяет обеспечить актуальность и достоверность предложений.

Ключевым преимуществом Handshake является тесная интеграция с университетскими карьерными центрами. Студенты получают подборку вакансий, адаптированных под их профиль обучения, а также могут участвовать в карьерных мероприятиях, вебинарах и виртуальных ярмарках вакансий. Интерфейс интуитивно понятен, а система предоставляет рекомендации и фильтрацию на основе интересов и опыта пользователя.

Преимущества Handshake:

* Точная адаптация вакансий под профиль обучения;
* Интеграция с университетами;
* Верификация работодателей;
* Поддержка карьерных мероприятий.

Недостатки:

* Платформа доступна только при участии вуза;
* Полная функциональность ограничена для иностранных студентов;
* Ориентация преимущественно на рынок США.

LinkedIn (раздел LinkedIn Students) — это международная профессиональная социальная сеть, включающая функциональность, ориентированную на студентов и начинающих специалистов. В разделе LinkedIn Students предлагаются рекомендации по стажировкам, вакансиям начального уровня и возможностям карьерного роста. Также доступны карьерные советы, профили работодателей и мероприятия.

Платформа отличается высоким уровнем охвата и возможностью установить профессиональные связи. Благодаря мощной системе рекомендаций, студент может получить доступ к персонализированным вакансиям. Однако LinkedIn не является специализированной студенческой платформой и не учитывает в полной мере особенности взаимодействия с образовательными учреждениями.

Преимущества LinkedIn Students:

* Широкая база вакансий и работодателей;
* Развитая система рекомендаций;
* Поддержка карьерного нетворкинга;
* Интеграция с профессиональным сообществом.

Недостатки:

* Отсутствие прямой связи с университетами;
* Ограниченный функционал именно для студентов;
* Не все вакансии доступны без премиум-доступа.

Graduateland — это европейская карьерная платформа, ориентированная на студентов и выпускников вузов. Платформа интегрирована с рядом университетов в Европе и предлагает вакансии, стажировки, карьерные мероприятия и профили работодателей. Студент может создать расширенный профиль и получать предложения, соответствующие его академическим данным и интересам.

Платформа поддерживает многоязычный интерфейс, но основная активность сосредоточена в Дании и странах Северной Европы. Для российских студентов возможности платформы ограничены отсутствием локальных партнерств и вакансий.

Преимущества Graduateland:

* Интеграция с университетами;
* Многоязычная поддержка;
* Возможность онлайн-собеседований;
* Персонализированная лента предложений.

Недостатки:

* Географическая ограниченность;
* Отсутствие вакансий на российском рынке;
* Некоторые функции доступны только при регистрации через вуз.

WayUp — американская платформа, ориентированная на студентов и молодых специалистов без опыта работы. Она предлагает вакансии начального уровня, программы стажировок и возможность прохождения виртуальных интервью. Платформа использует алгоритмы рекомендаций, предлагая вакансии на основе профиля пользователя.

Одной из особенностей WayUp является акцент на доступность: интерфейс ориентирован на быструю подачу заявки и участие в процессах отбора. Вместе с тем, платформа малоизвестна за пределами США и не поддерживает русскоязычные интерфейсы.

Преимущества WayUp:

* Упрощенный отклик и быстрая навигация;
* Адаптация под студентов без опыта;
* Поддержка видеоинтервью и карьерных советов.

Недостатки:

* Ограниченность англоязычной аудиторией;
* Отсутствие интеграции с вузами РФ;
* Недоступность локальных вакансий.
* Сравнение иностранных аналогов показано в таблице 2

Таблица 2 - Сравнение иностранных аналогов

| Критерий | Handshake | LinkedIn | Graduateland |
| --- | --- | --- | --- |
| Фокус на стажировках и практиках | 3 | 2 | 2 |
| Фильтрация по направлению подготовки | 3 | 2 | 2 |
| Интеграция с вузами | 3 | 1 | 3 |
| Поддержка студентов без опыта | 3 | 2 | 2 |
| Сопровождение процесса стажировки | 2 | 1 | 2 |
| Интерфейс, адаптированный под студентов | 3 | 2 | 2 |
| Система откликов и статусов | 3 | 2 | 3 |
| Модуль сообщений и общения | 2 | 2 | 2 |
| Система рекомендаций | 3 | 3 | 2 |
| Возможность загрузки сопроводительных писем | 3 | 2 | 3 |
| Публикация вакансий работодателями напрямую | 3 | 2 | 3 |

Зарубежные платформы (Handshake, Graduateland, WayUp) демонстрируют иной подход — они тесно связаны с университетами, предлагают персонализированные рекомендации, онлайн-собеседования и карьерную аналитику. Однако такие решения либо недоступны в России, либо не учитывают специфику отечественного образования и законодательства.

Таким образом, существует объективная потребность в разработке специализированного онлайн-сервиса, ориентированного на студентов российских вузов, с поддержкой практик, откликов, тестирования, карьерного сопровождения и защищенной обработки персональных данных [3].

## 1.3 Анализ целевой аудитории

Процесс трудоустройства студентов представляет собой взаимодействие между тремя основными участниками — студентами, работодателями и университетом. В современных условиях эффективное трудоустройство выпускников и прохождение стажировок невозможно без координации интересов всех сторон, но решающая роль в этом процессе принадлежит самим студентам и компаниям, предоставляющим возможности практического опыта. Университеты, в свою очередь, выступают как вспомогательные субъекты, информируют обучающихся о возможностях сервиса и рекомендующие ее в рамках сопровождения образовательного процесса.

Студенты, как основная категория пользователей цифровых решений в данной сфере, заинтересованы в получении профессионального опыта, который бы соответствовал их специализации, уровню подготовки и графику обучения. Также важную роль играют карьерные ожидания, необходимость практического портфолио и формирование первых профессиональных контактов. Поиск стажировки или места для прохождения практики без централизованного цифрового инструмента становится для студентов трудоемким, часто случайным и неэффективным. В условиях отсутствия структурированной поддержки со стороны образовательных учреждений онлайн-сервисы становятся единственным объективным средством навигации по рынку труда.

Работодатели рассматривают студентов как источник молодых кадров, обладающих современными знаниями и мотивацией к обучению. В отличие от крупных кампаний по массовому набору сотрудников, работа со студентами требует большей точности и гибкости. Для компаний важна возможность быстро получить доступ к базе резюме, отобрать кандидатов по профилю, оценить их компетенции и связаться с ними напрямую. Без специализированных систем этот процесс оказывается затрудненным, особенно при необходимости массового привлечения практикантов из разных вузов и регионов.

Образовательные учреждения выполняют преимущественно коммуникативную функцию, обеспечивая начальное информирование студентов о возможностях использования цифровых сервисов для поиска практики и трудоустройства. В рамках взаимодействия с платформами вузы могут осуществлять обмен информацией, предоставлять ссылки на сервисы в личных кабинетах студентов, проводить вводные презентации в начале семестра и оказывать консультационную поддержку. При этом академическое участие в процессе подбора или прохождения стажировки минимально, что соответствует современной модели децентрализованного взаимодействия.

Типовые сценарии взаимодействия между участниками строятся вокруг цифровой платформы. Студент самостоятельно регистрируется, заполняет профиль, получает персонализированные предложения, откликается на объявления работодателей, проходит первичное онлайн-тестирование или собеседование. Работодатель, в свою очередь, размещает описание вакансии или стажировки, настраивает фильтры по направлениям подготовки, получает отклики и выбирает кандидатов. Платформа выступает в роли посредника, обеспечивая удобный интерфейс, систему уведомлений, трекинг заявок и, при необходимости, техническую поддержку.

При отсутствии такого цифрового инструмента взаимодействие между студентами и работодателями оказывается фрагментарным. Студенты ищут предложения на сторонних ресурсах, часто не адаптированных под формат практик; работодатели не имеют централизованного доступа к профилям студентов, что затрудняет предварительный отбор; вузы же не могут отслеживать реальные карьерные траектории своих выпускников и корректировать программы на основе фактического трудоустройства. В результате каждый участник действует в условиях неопределенности и информационного дефицита.

Таким образом, цифровая платформа, выступающая как связующее звено между студентами и работодателями, обеспечивает прозрачность, доступность и структурированность процесса трудоустройства. Образовательные организации сохраняют роль коммуникативного посредника, способствующего внедрению платформы в студенческую среду, однако основное взаимодействие происходит напрямую между работодателями и обучающимися. Это соответствует современным тенденциям децентрализации сервисов и передачи ответственности за карьерный путь самим пользователям, при поддержке цифровых решений.

## 1.4 Требования к цифровой системе

Требования к разрабатываемому онлайн-сервису включают функциональные (описывающие, что должна делать система) и нефункциональные (характеризующие как система должна работать) компоненты. Согласно межгосударственному стандарту ГОСТ 34.602-2020, техническое задание (ТЗ) — это основной документ, определяющий требования к автоматизированной системе и порядок её разработки [4]. В ТЗ на АС обязательно выделяется раздел «Требования к автоматизированной системе», включающий подразделы с описанием структуры системы и выполняемых ею функций. Требования должны формулироваться так, чтобы каждая из функций была единичной, непротиворечивой, выполнимой, проверяемой и однозначной. Функциональные требования определяют конкретный функционал сервиса, необходимый для удовлетворения задач студентов и работодателей, а нефункциональные требования — качественные характеристики работы системы (производительность, безопасность, удобство и т.д.)

Функциональные требования:

1. Регистрация и управление пользователями различных ролей. Сервис должен обеспечивать создание и ведение учётных записей для студентов, работодателей и администраторов. При регистрации пользователи указывают обязательную информацию (для студентов — ФИО, учебную программу/специальность, контактные данные; для компаний — название, отрасль, требования к соискателям). Администратор при этом имеет возможности управлять учётными записями (модерировать регистрации, блокировать или удалять аккаунты);
2. Профили пользователей и резюме. Студенты должны иметь возможность заполнять подробные профили/резюме: образование, навыки, опыт (при наличии), направления подготовки. Это позволит сервису предлагать подходящие вакансии. Компании создают профиль с описанием сферы деятельности, требований к кандидатам и реквизитами работодателя. Администратор может редактировать конфиденциальную информацию и отслеживать активность пользователей;
3. Публикация и управление вакансиями. Работодатели размещают объявления о вакансиях и стажировках, указывая требования к кандидату, условия стажировки/работы, статус (активно/закрыто) и дополнительные параметры (район, график работы и др.)[1]. Администратор модерирует вакансии, контролирует достоверность размещаемых данных и может при необходимости удалять некорректные или устаревшие объявления;
4. Поиск и фильтрация вакансий. Студентам предоставляется функционал поиска по ключевым словам и фильтрация по различным критериям: направлению подготовки, специализации, уровню (бакалавр, магистр), региону, компании и др. Это ускоряет подбор актуальных предложений. Сервис должен поддерживать сочетание фильтров (например, направление + география) и сохранять часто используемые шаблоны поиска;
5. Отклики на вакансии и контроль статуса. Студенты могут подавать заявки на интересующие вакансии через онлайн-сервис, прикладывая резюме и сопроводительные документы. Работодатель просматривает отклики и назначает их статус (новое, рассмотрено, принято, отклонено). Статусы заявок фиксируются в системе, а пользователи оповещаются об изменениях. Функционал ведения истории заявок (посланных и полученных) должен быть реализован для обеих сторон;
6. Процесс прохождения стажировки через онлайн-сервис. Процесс прохождения стажировки на платформе начинается после того, как работодатель принимает отклик студента. Сервис фиксирует статус «принято», после чего вакансия отображается в личном кабинете как активная. Студент получает доступ к прикреплённым заданиям и может загружать итоговые материалы, такие как отчёт о прохождении практики. Работодатель по завершении стажировки обновляет статус на «завершено», что позволяет сформировать историю прохождения стажировок и сохранить результаты взаимодействия обеих сторон.
7. Административные функции. Для администратора сервиса необходимы специальные сервисы: управление пользователями (создание, блокировка, снятие блокировок, выдача/снятие прав), модерация контента, мониторинг работы системы (логи, отчёты об активности).

Нефункциональные требования определяют качественные характеристики работы онлайн-сервиса и описывают, как система должна выполнять свои функции. Основные категории таких требований следуют из модели качества ПО по стандарту ISO/IEC 25010:2014 (ГОСТ Р ИСО/МЭК 25010-2014) – к ним относятся функциональная пригодность, производительность, удобство использования, надежность, безопасность и совместимость [20]. В контексте веб-платформы для трудоустройства студентов ключевыми нефункциональными требованиями являются:

1. Интерфейс должен быть интуитивно понятным, с единообразным дизайном, минимальным числом кликов для основных операций. Требуется адаптивная верстка для корректного отображения на разных устройствах (ПК, планшет, смартфон) и в разных браузерах. Важно обеспечить доступность для пользователей с ограниченными возможностями (участие в проекте WCAG), следуя принципам универсального дизайна: крупные шрифты, контрастная цветовая схема, поддержка навигации с клавиатуры, альтернативные способы представления информации [27]. Согласно международным стандартам, веб-платформы должны поддерживать совместимость с разными устройствами и браузерами, а интерфейс строиться с учетом требований эргономики (например, ГОСТ Р ИСО 9241-210).
2. Онлайн-сервис должна быстро обрабатывать запросы пользователей даже при высокой нагрузке. Время ответа на действия (загрузка страниц, фильтрация результатов, отправка откликов) должно соответствовать приемлемым нормативам (например, не более нескольких сотен миллисекунд на серверной стороне). Проектирование должно предусматривать эффективное использование ресурсов (CPU, память, БД), поддержку кэширования и балансировки нагрузки. В случаях массового одновременного посещения (августовские интенсивы, открытые вебинары) система обязана сохранять приемлемую скорость работы и не терять данные.
3. Архитектура должна позволять расширять сервис (горизонтальное и вертикальное масштабирование) при увеличении числа пользователей, объявлений и объёма данных. Используемые решения (серверы, СУБД, очереди задач) должны легко масштабироваться, обеспечивая рост производительности. Следует предусмотреть как рост нагрузки в пиковые периоды, так и плановое увеличение функциональности (подключение новых сервисов, интеграция с внешними системами);
4. Онлайн-сервис обязан защищать персональные данные пользователей и конфиденциальную информацию [3]. Необходимо реализовать современные механизмы аутентификации и авторизации, хранение паролей в зашифрованном виде (например, с использованием bcrypt), защиту от уязвимостей (SQL-инъекции, XSS и т.д.). Все критические данные (вакансии, резюме, личная информация) должны передаваться по защищённому каналу (HTTPS/SSL). В соответствии с российским законодательством (Федеральный закон «О персональных данных» (ФЗ-152) и сопутствующими нормативами) следует обеспечить обязательное шифрование и хранение персональной информации на сертифицированных серверах. Государственные регуляторы требуют применения международных стандартов безопасности (например, ISO/IEC 27001) и сертификатов ФСТЭК/ФСБ для криптографической защиты данных;
5. Код и архитектура должны быть организованы так, чтобы облегчить дальнейшее сопровождение, добавление новых функций и исправление ошибок. Следует применять модульный дизайн, стандартизированную документацию (по ГОСТ 19.201 «Техническое задание») и лучшие практики кодирования [5]. Это позволит снизить трудозатраты на поддержку и обеспечит долгосрочную жизнеспособность системы.

Все перечисленные нефункциональные требования опираются на международные и государственные стандарты качества ПО. Так, модель качества по ISO/IEC 25010 (ГОСТ Р ИСО/МЭК 25010-2014) выделяет ключевые характеристики, позволяющие формулировать показатели надёжности, удобства и безопасности системы [20]. Соблюдение требований безопасности и защиты данных регламентируется законом «О персональных данных» и связанными стандартами ФСТЭК/ФСБ. Удобство интерфейса и доступность платформы должны соответствовать международным рекомендациям по UX и доступности (WCAG). Таким образом, функциональные и нефункциональные требования взаимосвязаны и дополняют друг друга, обеспечивая создание эффективного, безопасного и удобного онлайн-сервиса для взаимодействия студентов и работодателей.

## 1.5 Подход к разработке: принципы и этапы

Разработка онлайн сервиса необходимо вести на основе практико-ориентированного подхода, сочетающего требования пользовательской логики с реалиями программной реализации. Основное внимание при проектировании уделяется модульности архитектуры, масштабируемости, удобству сопровождения и соответствию современным требованиям к безопасности и интерфейсу. В процессе реализации учитываются как технические ограничения, так и особенности целевой аудитории — студентов и работодателей.

В процессе проектирования платформы используются следующие методологические принципы:

* Модульность: каждый функциональный блок сервиса (аутентификация, вакансии, отклики, уведомления, сообщения и т.д.) реализован как независимый программный модуль. Это позволяет вносить изменения, не затрагивая остальной код, и обеспечивает легкое масштабирование системы в будущем.
* Разделение ответственности: серверная и клиентская части сервиса строго разделены. Сервер обрабатывает запросы, отвечает за безопасность, валидацию данных и работу с базой, а клиент отображает интерфейс и взаимодействует с API.
* Принцип DRY (Don't Repeat Yourself): в рамках архитектуры исключалось дублирование логики, что позволило сделать код более устойчивым и читаемым.
* Адаптивность интерфейса: при проектировании структуры пользовательского взаимодействия учитывалось, что студенты и компании могут заходить в сервис с разных устройств и иметь разный уровень цифровой грамотности. Интерфейс организован так, чтобы доступ к ключевым действиям был интуитивно понятен.
* Безопасность и контроль доступа: с самого начала закладывались механизмы разграничения прав на уровне API и пользовательского интерфейса, а также защита персональных данных в соответствии с нормативными требованиями.

Работа над проектом ведется поэтапно, с последовательным переходом от анализа к реализации и тестированию компонентов. Были выделены следующие ключевые этапы:

* Анализ предметной области - этап, определяющий основные потребности студентов в процессе поиска и прохождения стажировок, выявляются узкие места существующих решений, анализируются бизнес-процессы трудоустройства, собираются функциональные требования к онлайн-сервису.
* Формализация требований и проектирование архитектуры – этап, включающий в себя определение сценариев использования и разработка структура базы данных, обеспечивающая хранение и обработку всей необходимой информации.
* Реализация серверной логики – этап, на котором создается API для взаимодействия с клиентской частью. Также эта часть обеспечивает работу с базой данных, валидацию, авторизацию и маршрутизацию. На этом этапе также прорабатываются механизмы обработки ошибок и возврата статусов.
* Реализация клиентской части включает в разработку интерфейса пользователя, с учетом ролевой модели (студент/работодатель). Интерфейс взаимодействует с сервером через HTTP-запросы, обрабатывает ответы API и отображает данные в адаптивной форме.
* Тестирование пользовательских сценариев – этап, где изначально происходит ручное тестирования ключевых сценариев онлайн-сервиса: регистрация, вход, подача заявки, публикация вакансии, переписка, прохождение стажировки, выполнение заданий. Этот этап позволяет выявить и устранить критические ошибки на ранней стадии.
* Оптимизация и подготовка к демонстрации - финальнй этап, где устраняются дублирующие запросы, добавляются уведомления, корректируются роли и обеспечивается стабильность базовых функций.

Такой последовательный подход обеспечивает управляемость разработки, минимизирует технические риски и позволяет выстроить логичную структуру, соответствующую как требованиям пользователей, так и ожиданиям по качеству реализации программного продукта.

## 1.6 Выбор инструментов для разработки

Выбор технологического стека для разработки онлайн-сервиса был обусловлен требованиями к функциональности, масштабируемости и удобству сопровождения сервиса. Основной задачей являлось создание устойчивой клиент–серверной архитектуры, обеспечивающей динамическое взаимодействие между студентами и работодателями, а также работу с вакансиями, откликами, сообщениями и рекомендательным механизмом.

В качестве среды разработки была выбрана модель с разделением на frontend и backend, что обеспечивает независимость интерфейса от логики обработки данных, а также позволяет в дальнейшем масштабировать проект или развернуть его в распределенной среде.

Для разработки клиентской части онлайн-сервиса важно выбрать фреймворк, который не только будет обеспечивать высокую производительность и удобство разработки, но и соответствует масштабу проекта, его уровню сложности, а также возможностями поддержки и развития в долгосрочной перспективе. Существующие JavaScript-фреймворки обладают различными архитектурными подходами, стилем организации коода и инструментами работы с интерфейсом. В процесс анализа были рассмотрены три наиболее распространенные решения: React, Angular и Vue.js, каждое из которых широко применяется в современной frontend-разработке.

React – библиотека для построения пользовательских интерфейсов [26]. Он реализует концепцию декларативного программирования и использует виртуальный DOM для повышения производительности интерфейса. Среди ключевых достоинств данного фреймворка можно выделить:

1. Высокая гибкость и модульность;
2. Наличие большого количества сторонних решений и библиотек;
3. Частые обновления

Однако React, являясь библиотекой, а не полноценным фреймворком, требуется подключение сторонних инструментов для реализации таких базовых функций, как маршрутизация, глобальное хранилище и взаимодействие с API. Это увеличивает сложность начальной конфигурации проекта и требует более глубокого понимания архитектурных решений со стороны разработчика. Кроме того, отсутствует единый стиль организации кода, что усложняет сопровождение больших проектов.

Angular — это полнофункциональный фреймворк, разработанный Google и основанный на языке TypeScript [24]. Он предоставляет разработчику обширный набор встроенных средств для построения сложных одностраничных приложений, включая систему маршрутизации, форму обработки данных, внедрение зависимостей и модульную структуру. Основные преимущества Angular:

1. Строгая архитектура и единый подход к организации кода;
2. Встроенные инструменты для тестирования и валидации;
3. Поддержка масштабируемых корпоративных решений.

Тем не менее, Angular считается наиболее сложным для освоения среди трёх рассматриваемых фреймворков. Его синтаксис отличается высокой формализованностью, а объём кода, необходимый для реализации даже базовых интерфейсных компонентов, может быть избыточным. Кроме того, архитектурные ограничения делают Angular менее гибким для небольших или средних проектов, в которых не требуется весь набор встроенного функционала.

Vue.js — это прогрессивный JavaScript-фреймворк, ориентированный на создание интерфейсов с использованием компонентного подхода [25]. Vue.js сочетает лучшие идеи React и Angular, при этом предлагает разработчику более простой синтаксис и низкий порог входа. Преимущества Vue.js включают:

1. Встроенную поддержку маршрутизации и управления состоянием;
2. Интуитивно понятную структуру компонентов и декларативный синтаксис;
3. Высокую производительность и реактивность пользовательского интерфейса;
4. Гибкость в архитектуре и возможности постепенного внедрения.

Относительным недостатком Vue.js может считаться его меньшая популярность в крупных западных компаниях по сравнению с React и Angular. Однако на российском рынке и в сфере средних и малых проектов он активно используется и поддерживается, особенно благодаря лёгкости обучения и стабильному API.

Сравнение аналогов для разработки клиентской части онлайн-сервиса показано в таблице 3

Таблица 3 – Сравнение аналогов для разработки frontend части

| Критерий | Vue.js | React | Angular |
| --- | --- | --- | --- |
| Тип решения | Полноценный фремворк | Библиотека для реализации интерфейса | Фреймворк корпоративного уровня |
| Язык | JavaScript | JavaScript + jsx | TypeScript |
| Порог входа | Низкий | Средний | Высокий |
| Компонентная архитектура | Простая в освоении | Требуются сторонние решения | Сложен в освоении |
| Встроенные инструменты | Маршрутизация, управление состоянием | - | Маршрутизация, формы, управление состоянием |
| Удобство настройки | Простая настройка | Средняя сложность настройки | Тяжелая сложность настройки |
| Подходит для проектов | Малых и средних | Гибких SPA-приложений | Крупных |
| Документация и сообщество | Большая документация, активное сообщество | Очень большое сообщество и документация | Официальная поддержка от Google |
| Основные недостатки | Небольшая популярность | Слишком свободная архитектура | Сложность и избыточность |

На основании анализа возможностей, архитектурной гибкости, уровня сложности и требований проекта, фреймворк Vue.js был выбран как наилучшее решение для реализации клиентской части платформы. Он обеспечивает баланс между простотой использования, мощным функционалом и удобством сопровождения, что особенно важно в условиях ограниченного времени и ресурсов, характерных для учебной проектной деятельности.

Для разработки серверной части онлайн-сервиса требуется выбор подходящей платформы, обеспечивающей стабильную работу API, поддержку обработки большого количества запросов, удобную интеграцию с базой данных и совместимость с клиентской частью. Кроме того, важную роль играют масштабируемость, гибкость архитектуры и доступность экосистемы инструментов. В современной веб-разработке используется несколько технологий для создания backend-решений, среди которых наибольшее распространение получили Node.js, Django и Spring Boot [21]. Каждая из них имеет свои особенности и применяется в зависимости от целей и масштаба проекта.

Node.js — это серверная среда выполнения JavaScript, основанная на движке V8 [10]. Она позволяет запускать JavaScript вне браузера, что открыло возможности для использования одного языка программирования как на клиенте, так и на сервере. В сочетании с фреймворком Express.js, Node.js предоставляет простой и гибкий способ создания RESTful API и управления HTTP-запросами [14]. Основные преимущества данного решения:

1. Использование одного языка разработки (JavaScript) на всём протяжении проекта;
2. Асинхронная неблокирующая модель ввода-вывода, что повышает производительность при большом числе одновременных запросов;
3. Гибкая архитектура и лёгкость подключения сторонних библиотек;
4. Высокая скорость разработки и простота настройки серверной логики.

Express.js позволяет быстро организовать маршрутизацию, обрабатывать POST- и GET-запросы, работать с cookie, заголовками, сессиями и middleware [11]. Это делает его идеальным выбором для небольших и средних проектов, а также для тех случаев, когда важна скорость прототипирования и понятность архитектурных решений.

Django — это высокоуровневый фреймворк на языке Python, ориентированный на быстрое создание веб-приложений с акцентом на безопасность, масштабируемость и соблюдение архитектурных паттернов. Среди преимуществ Django можно выделить:

1. Встроенную административную панель;
2. Автоматическое создание ORM-моделей на основе базы данных;
3. Наличие множества готовых компонентов (аутентификация, формы, шаблоны);
4. Строгое следование архитектуре MVC (Model-View-Controller).

Однако Django предъявляет более высокие требования к структуре проекта, требует использования языка Python, что делает его менее совместимым с JavaScript-интерфейсами. В проектах, где клиентская часть реализуется на JS-фреймворках, возникает необходимость в более сложной конфигурации и раздельной разработке frontend и backend, что усложняет взаимодействие между слоями.

Spring Boot — это фреймворк на языке Java, предназначенный для разработки масштабируемых и отказоустойчивых корпоративных приложений. Он предоставляет расширенные возможности конфигурации, безопасности и интеграции с различными модулями (базы данных, REST-сервисы, системы логирования и мониторинга). Его сильные стороны:

1. Строгая типизация и высокая стабильность;
2. Развитая система аннотаций и шаблонов;
3. Широкая поддержка микросервисной архитектуры.

Вместе с тем, Spring Boot требует значительных ресурсов, сложен в освоении и избыточен для небольших учебных или MVP-проектов. Он подходит в первую очередь для крупных распределённых систем, но может быть неэффективным в условиях ограниченного времени и команды без опыта Java-разработки.

Сравнение аналогов для разработки серверной части онлайн-сервиса показано в таблице 4

Таблица 4 – Сравнение аналогов для разработки серверной части

| Критерий | Node.js + Express.js | Django | Spring Boot |
| --- | --- | --- | --- |
| Тип решения | Легкий фреймворк | Полноценный фреймворк | Фреймворк корпоративного уровня |
| Язык | JavaScript | Python | Java |
| Порог входа | Низкий | Средний | Высокий |
| Структура проекта | Гибкая | Строгое разделение по слоям | Строгое структура с аннотациями |
| Скорость разработки | Высокая | Высокая | Средняя |
| Поддержка API | Быстрая настройка | Встроенные классы | Полноценная поддержка |
| Подходит для проектов | Малых и средних | Административных систем | Крупных |
| Инструменты и экосистема | Npm, большое количество middleware | Встроенные компоненты, ORM | Расширенные модули, поддержка микросервисов |
| Основные недостатки | Небольшая популярность | Слишком свободная архитектура | Сложность и избыточность |

На основании анализа был сделан выбор в пользу Node.js и Express.js как наиболее подходящей связки для реализации серверной части сервиса. Такое решение обеспечивает:

1. Совместимость с клиентской частью, разработанной на Vue.js;
2. Быструю разработку RESTful API;
3. Простое подключение реляционной базы данных (MySQL);
4. Удобную структуру маршрутов и middleware;
5. Возможность обработки асинхронных событий, что критично при реализации откликов и уведомлений.

Таким образом, Node.js с Express.js предоставляет оптимальное сочетание простоты, гибкости и производительности, что делает его обоснованным выбором для создания серверной логики онлайн-сервиса, направленный на поддержку взаимодействия студентов и работодателей.

Для хранения и обработки данных в разрабатываемой платформе используется система управления базами данных MySQL. Это одно из самых популярных реляционных решений с открытым исходным кодом, зарекомендовавшее себя как надёжная, стабильная и производительная СУБД. Она активно применяется как в небольших проектах, так и в высоконагруженных системах, включая коммерческие и корпоративные приложения.

Ключевыми преимуществами MySQL являются поддержка внешних ключей, транзакций, индексов, а также гибкий язык запросов SQL, соответствующий стандартам [15]. Система предоставляет все необходимые механизмы обеспечения целостности данных, что особенно важно при реализации связей между множеством сущностей: пользователями, вакансиями, откликами, уведомлениями, сообщениями, результатами тестов и другими объектами. Использование реляционной модели хранения данных позволяет явно задавать структуру и взаимосвязь между таблицами, обеспечивая логическую строгость и надёжность операций.

Кроме того, MySQL предлагает широкий выбор инструментов администрирования и мониторинга, включая phpMyAdmin, MySQL Workbench и другие утилиты [16]. Это облегчает отладку, миграции и резервное копирование, что делает систему особенно удобной для небольших команд и учебных проектов.

Альтернативным вариантом могла бы выступить PostgreSQL, ещё одна популярная реляционная СУБД, отличающаяся поддержкой более сложных операций, расширяемостью и строгим соблюдением стандартов SQL. PostgreSQL считается более мощным инструментом в части обработки геоданных, работы с JSON и выполнения вложенных запросов. Однако она предъявляет более высокие требования к настройке и администрированию, что может усложнить разработку в рамках ограниченного по времени дипломного проекта.

Также были рассмотрены нереляционные решения, в частности MongoDB, основанная на документно-ориентированном подходе. Её плюсом является гибкость структуры и скорость прототипирования. Тем не менее, из-за отсутствия строгой схемы и сложности построения связей между сущностями MongoDB менее подходит для систем, где важна логическая согласованность и взаимосвязь данных, как в случае с сервисом для стажировок.

На основании этих факторов выбор был сделан в пользу MySQL как проверенного инструмента, обеспечивающего баланс между надёжностью, функциональностью и простотой в использовании. Структура базы данных спроектирована с соблюдением принципов нормализации, что позволило избежать избыточности и обеспечить целостность данных при высокой читаемости и масштабируемости модели.

Для хранения паролей применяется алгоритм хэширования bcrypt, обеспечивающий базовую безопасность учётных записей. Идентификация ролей реализуется на стороне сервера с проверкой прав доступа к различным разделам.

Дополнительно в процессе разработки использовались следующие инструменты:

* Postman для тестирования API;
* phpMyAdmin — как средство визуального управления базой данных;
* Git для версионного контроля проекта;
* Visual Studio Code в качестве основной среды разработки.

Таким образом, выбранные инструменты обеспечивают не только реализацию текущего функционала, но и создают основу для масштабируемого, модульного решения. Их применение позволяет добиться надёжной работы платформы, удобства поддержки и прозрачности логики взаимодействия между её компонентами.

## 1.7 Выводы по первой главе

В первой главе проведён обзор теоретических основ, касающихся разработки онлайн-сервисов, цифрового взаимодействия студентов и работодателей, а также проанализирована предметная область с рассмотрением существующих аналогов. Рассмотрены ключевые архитектурные подходы, включая клиент-серверную модель, принципы модульности, масштабируемости и безопасности, что позволило сформировать представление о базовых требованиях к разработке эффективного цифрового решения в данной области. Обоснована необходимость соблюдения функциональных и нефункциональных требований, определены критерии, влияющие на надёжность, доступность и удобство использования онлайн-сервисов, в том числе с опорой на международные стандарты.

На основании сравнительного анализа отечественных и зарубежных платформ выявлены значительные ограничения текущих решений, среди которых — отсутствие адаптации под образовательный процесс, слабая интеграция с вузами и недостаточная поддержка студентов без опыта. Выделены ключевые аспекты, необходимые для реализации платформы, ориентированной именно на студентов, включая фильтрацию по направлениям подготовки, сопровождение практик и возможность обратной связи. Также подробно изучена целевая аудитория и сценарии её взаимодействия с платформой.

Сделан обоснованный выбор технологического стека, включая фреймворки Vue.js и Node.js, а также систему управления базами данных MySQL, основываясь на их преимуществах для создания гибкого, производительного и поддерживаемого программного продукта. Результатом анализа стало формирование требований к функциональности, архитектуре и интерфейсу будущей платформы, что создаёт прочную основу для последующих этапов проектирования и реализации.

# 2. Проектирование и функциональная концепция онлайн-сервиса трудоустройства студентов

## 2.1 Концепция онлайн-сервиса и ее ключевые функции

Разрабатываемая платформа представляет собой специализированный онлайн-сервис, ориентированный на цифровое упрощение процесса поиска практики и стажировок студентами вуза. В отличие от универсальных сайтов трудоустройства, платформа сфокусирована исключительно на потребностях студентов без опыта и работодателей, заинтересованных в привлечении молодых кадров для временной занятости, стажировок или подготовки резерва специалистов. Концепция платформы основывается на устранении выявленных ранее проблем: фрагментарности источников, не структурированности вакансий и отсутствии обратной связи по результатам взаимодействия, а также предоставляет возможность прохождения стажировки.

Система функционирует как связующее звено между двумя целевыми группами пользователей — студентами и работодателями — и организует их взаимодействие в рамках сервиса. Каждый пользователь сервиса выполняет набор операций в соответствии со своей ролью, а логика построена таким образом, чтобы все ключевые этапы — от поиска до подачи заявки — проходили централизованно, прозрачно и с возможностью последующего контроля. Основу архитектуры составляет клиент-серверная модель, обеспечивающая быстрый отклик и масштабируемость, а все функции сгруппированы в логические модули.

Ключевые функции платформы сформированы на основании анализа практических затруднений, с которыми сталкиваются обе стороны. Центральными среди них являются:

1. Онлайн-сервис предоставляет возможность создания аккаунта для студентов и работодателей с разграничением ролей и прав доступа. Студенты указывают при регистрации направление подготовки, курс, образовательную программу и контактные данные, что позволяет системе формировать персонализированный интерфейс и предлагать соответствующие вакансии. Работодатели создают карточку организации с описанием деятельности, контактным лицом и юридическим статусом, что необходимо для обеспечения доверия и модерации публикуемого контента;
2. Каждый студент получает доступ к структурированному цифровому профилю, который включает данные об образовании, изучаемых дисциплинах, навыках, уровне владения языками, а также информацию о предыдущем опыте, если таковой имеется. В профиле также предусмотрена возможность загрузки мотивационного письма и других приложений. Работодатели используют эти сведения для предварительного отбора кандидатов, а сервис — для автоматического подбора подходящих вакансий;
3. Работодатели могут размещать объявления о стажировках с указанием ключевых параметров: формат (очная, дистанционная, гибридная), длительность, график, требования к студенту, описание задач и направлений подготовки. Платформа предоставляет студентам расширенную систему фильтрации по параметрам, соответствующим их учебной деятельности: специальности, графику, продолжительности и географическому расположению. Это обеспечивает релевантность предложений и экономию времени при поиске;
4. После выбора интересующей стажировки студент может отправить отклик, прикрепив резюме и сопроводительные документы. Работодатель получает уведомление о поступлении заявки и может изменить ее статус на одном из этапов: «рассматривается», «отклонено», «приглашение», «принять», «завершено». История откликов сохраняется в личных кабинетах обеих сторон, обеспечивая прозрачность и отслеживаемость взаимодействия на протяжении всего процесса;
5. Для повышения вовлечённости пользователей реализована система уведомлений о ключевых событиях: откликах, изменении статуса заявок, новых вакансиях, изменениях в профиле или требованиях. Уведомления реализованы в виде email-оповещений и всплывающих сообщений на платформе;
6. На основе информации из пользовательских профилей, истории активности и предпочтений сервис формирует подборку рекомендованных вакансий. Это позволяет минимизировать ручной поиск, ускоряет выбор и повышает вероятность нахождения подходящей стажировки без дополнительных усилий со стороны студента;
7. Для удобства прохождения стажировок и практик используется модуль, который помогает контролировать этот процесс. Работодатель может выставлять задания, а студенты могут их выполнять и присылать отчеты в текстовом виде и с прикреплением файлов. Весь прогресс прохождения стажировки/практики будет отображаться в личном кабинете пользователей;
8. Для контроля качества и поддержания актуальности контента сервис включает модуль администрирования. Администратор получает доступ к функциям модерации профилей, вакансий, управления правами доступа, а также формирования статистических отчетов по активности пользователей и эффективности взаимодействия. Это обеспечивает стабильность работы системы, соблюдение правил и накопление аналитических данных;

Выбор именно этих функций обусловлен необходимостью обеспечить полноту пользовательского сценария без привлечения сторонних каналов. Регистрация и профили создают базу для персонализированного взаимодействия. Система вакансий и откликов решает задачу формального оформления заинтересованности. Уведомления и рекомендации делают работу с платформой удобной и своевременной. Вместе эти компоненты формируют цифровую среду, в которой студент может пройти весь путь от поиска до прохождения стажировки, а работодатель — эффективно взаимодействовать с релевантными кандидатами.

Также были созданы диаграммы последовательности по основным функциям онлайн-сервиса. Данные диаграммы показаны на рисунках 1-4

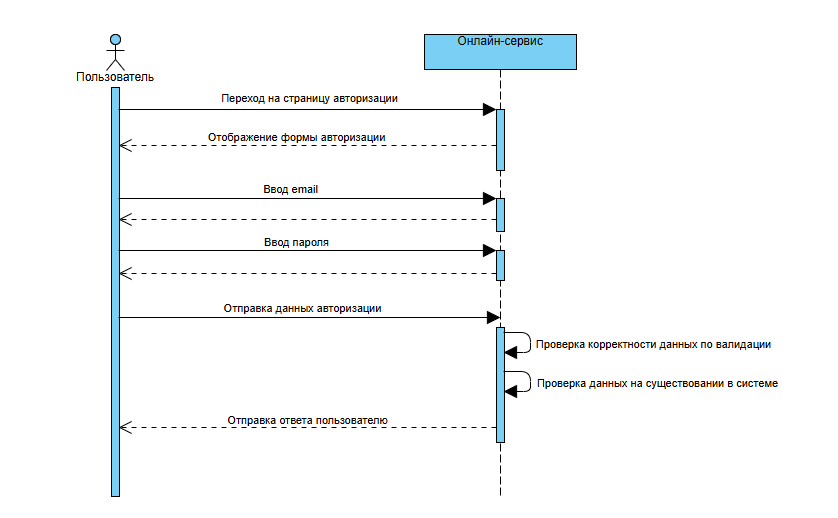


Рисунок 1 – Диаграмма последовательности авторизации

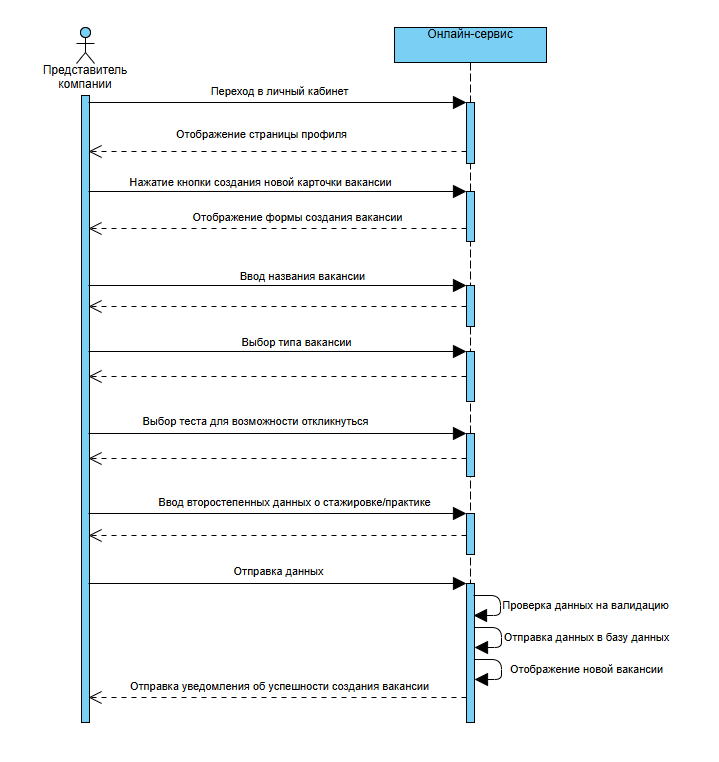


Рисунок 2 – Диаграмма последовательности создания новой вакансии

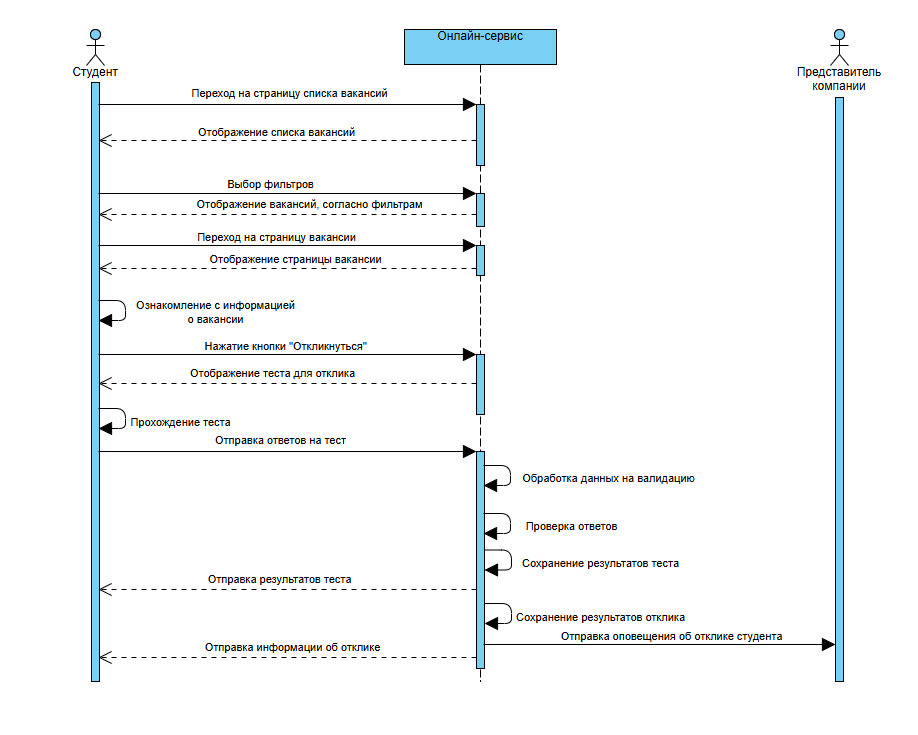


Рисунок 3 – Диаграмма последовательности отклика студента на вакансию

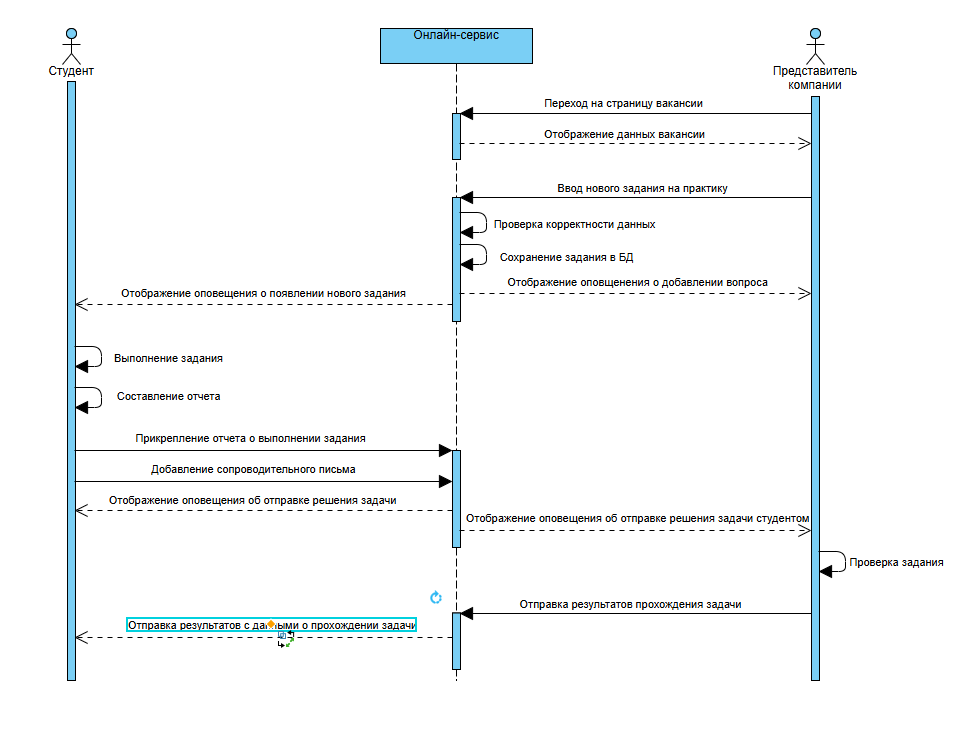


Рисунок 4 – Диаграмма последовательности прохождения практики

Также для более корректного понимания последовательности выполнения основных сценариев в системе были использованы диаграммы бизнес-процессов по основным функциям системы. Бизнес-процессы представляют собой совокупность взаимосвязанных действий, направленных на достижение конкретного результата в рамках функционирования организации или системы. В контексте цифровых платформ бизнес-процессами принято считать формализованные сценарии взаимодействия пользователей с системой: регистрация, оформление заявки, обработка информации, модерация, отклик и прочее [8]. Такие процессы характеризуются устойчивой логикой выполнения, повторяемостью и ориентированы на достижение определенных целей, соответствующих общей архитектуре и назначению платформы.

Эффективное проектирование бизнес-процессов способствует структурированному подходу к разработке информационных систем. Четкое описание процессов позволяет выявить узкие места, повысить управляемость, определить ответственных участников, улучшить взаимодействие между компонентами системы, а также автоматизировать критически важные этапы.

Для формализованного представления и документирования бизнес-процессов применяются различные нотации моделирования [17]. Одной из наиболее распространённых является BPMN (Business Process Model and Notation) [9]. Это международный стандарт (ISO/IEC 19510:2013), который позволяет визуально описывать процессы в виде блок-схем, понятных как техническим специалистам, так и бизнес-пользователям. Основной целью BPMN является предоставление единого языка, с помощью которого можно отразить последовательность действий, условия принятия решений, параллельные ветви, взаимодействие участников и другие элементы процесса [23].

BPMN включает в себя набор графических элементов: события, задачи, шлюзы, потоки, пулы и дорожки [22]. Эти элементы позволяют моделировать как простые, так и сложные процессы с различными сценариями развития и взаимодействием между участниками. Например, событие может отражать начало процесса или получение внешнего сигнала, задача — конкретное действие, шлюз — точку принятия решений, а пул или дорожка — конкретного исполнителя (роль в системе).

Использование BPMN при проектировании цифровых платформ позволяет:

* Описывать текущие (AS-IS) и целевые (TO-BE) процессы;
* Улучшать документацию требований и технических заданий;
* Выстраивать единое понимание между заказчиком, разработчиками и конечными пользователями;
* Визуализировать логику работы платформы, тем самым повышая её прозрачность и масштабируемость.

В ходе разбора основных функций системы были спроектированы схемы бизнес-процессов в нотации BPMN. Эти схемы показаны на рисунках 5-8

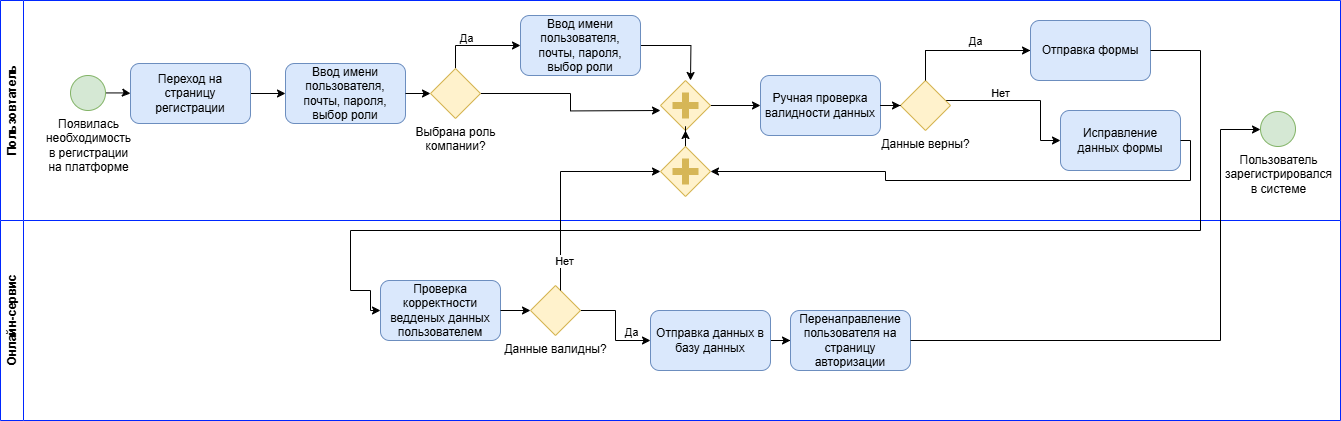


Рисунок 5 - Регистрация пользователя в системе

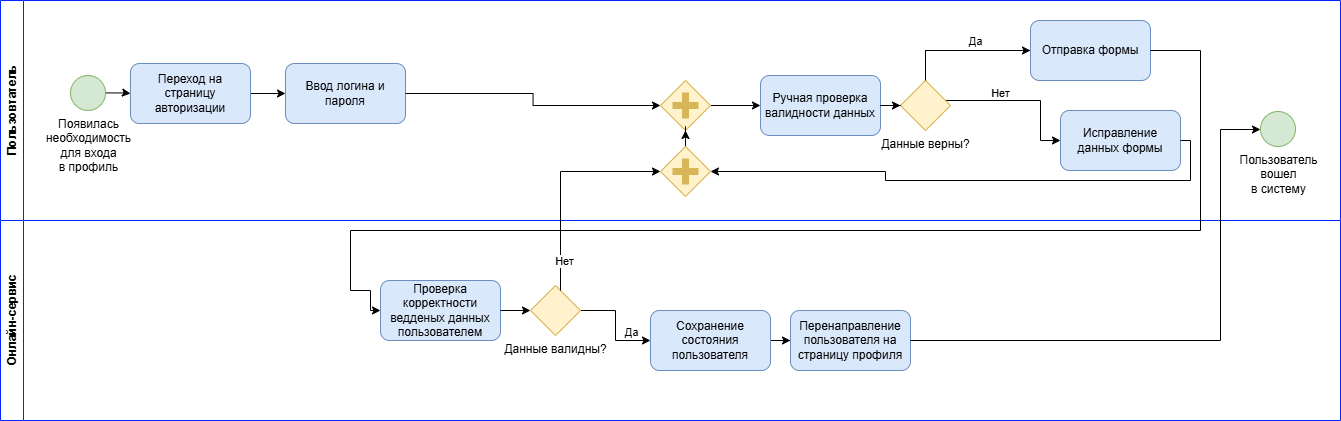


Рисунок 6 - Авторизация пользователя в системе

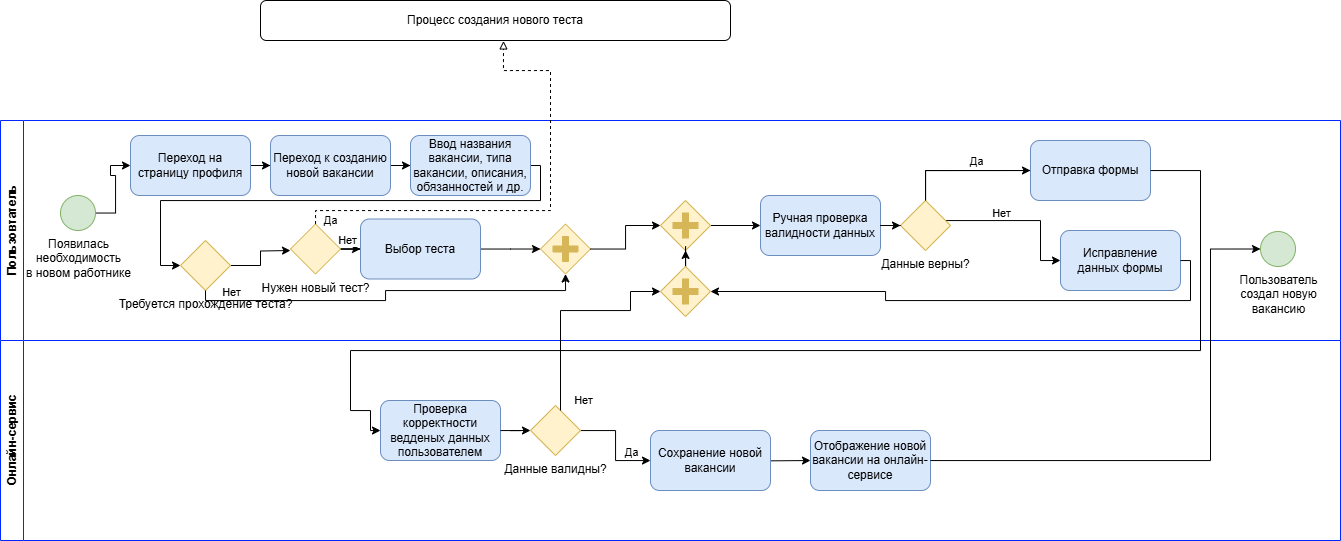


Рисунок 7 - Бизнес-процесс создания новой вакансии

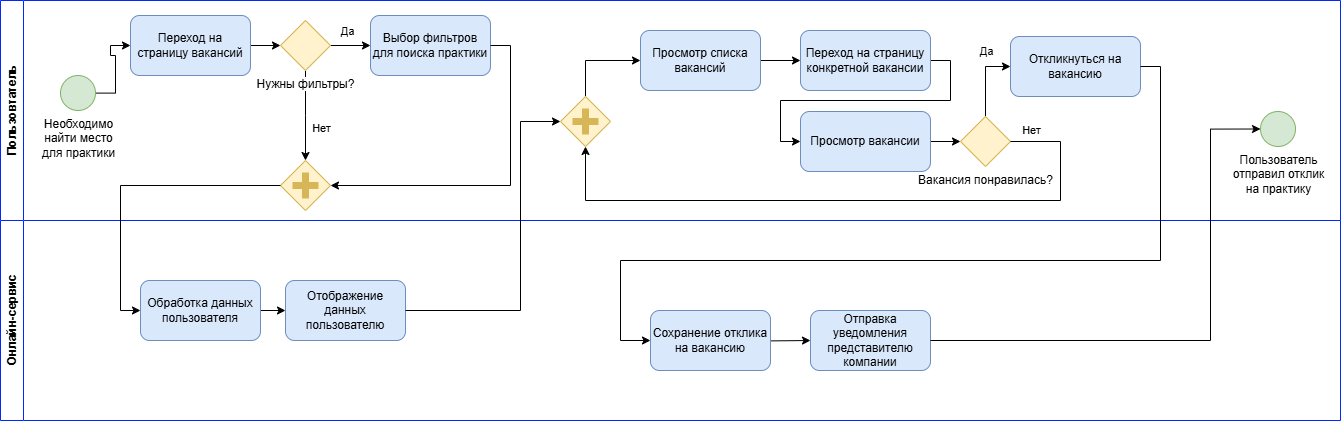


Рисунок 8 - Схема бизнес-процесса отклика на вакансию

Таким образом, концепция онлайн-сервиса направлена на формирование целостной цифровой инфраструктуры, упрощающей, ускоряющей и структурирующей процесс профессионального старта студентов. Выделенные функции обеспечивают не только удобство, но и соответствие потребностям каждой категории пользователей, а также служат фундаментом для масштабируемого и нормативно обоснованного ИТ-решения.

## 2.2 Модель данных и организация хранения информации

Модель данных онлайн-сервиса разрабатывается на основе реляционного подхода и отражает ключевые сущности, участвующие в процессе взаимодействия между студентами и работодателями. Структура хранения организована таким образом, чтобы поддерживать все основные функции сервиса: от регистрации пользователей и размещения вакансий до откликов, рекомендаций, уведомлений и прохождения стажировки.

Процесс регистрации и авторизации пользователей представляет собой один из первичных этапов взаимодействия с платформой и служит основой для разграничения прав доступа, формирования индивидуальных персонализированных данных и обеспечения их безопасности. В рамках данного процесса участвуют две основные таблицы: users и roles.

Таблица roles является самодостаточной, она содержит в себе всего два поля: id и name. Первое поле – идентификатор роли, чтобы можно было определять права доступа к конкретным типам пользователей. Поле name имеет строковое значение, которое содержит имя данной роли, как например студент. Изначально в данной таблице содержится 3 основных роли: студент, работодатель и администратор. Данная структура таблицы позволяет безопасно добавлять новые роли в систему, что даст ей в будущем возможность для большего масштабирования.

Таблица users содержит информацию о зарегистрированных пользователях онлайн-сервиса. Она включает в себе 10 полей:

1. id – уникальный идентификатор каждого пользователя. Благодаря ему, большинство таблиц сможет настроить взаимодействие между разными пользователями;
2. username – строковое поле, которое будет хранить псевдоним пользователя. В основном пользователи будут указывать свое имя, но также у них имеется возможность указывать псевдоним и потом уже в чате представляться;
3. email – поле контактного адреса, на который будут приходить уведомлению пользователю о действиях в системе, а также он будет использоваться для входа в систему;
4. password – пароль от личного кабинета;
5. role\_id – внешний ключ, указывающий на тип пользователя (студент, работодатель, администратор);
6. created\_at – временная метка создания учетной записи, которое заполняется автоматически;
7. updated\_at – временная метка последнего изменения в учетной записи, которое заполняется автоматически;
8. company\_name – необязательное поле, которое требуется для указания названия компании, которое потом будет отображаться в личном кабинете и на страницах вакансий;
9. company\_website - необязательное поле, которое требуется для указания пути к вебсайту компании, которое потом будет отображаться в личном кабинете и на страницах вакансий. Это поле не обязательно для заполнения и у компании, если вебсайта нет, но для полного понимания студентов – поле рекомендуется заполнять;
10. company\_description - необязательное поле, которое требуется для указания дополнительной информации о компании, которое потом будет отображаться в личном кабинете и на страницах вакансий. Это поле не обязательно для заполнения и у компании, но оно поможет студентам лучше понимать, чем занимается компания, предоставляющая место стажировки/практики;

Пользователи при регистрации предоставляют основные идентификационные данные, которые валидационно обрабатываются на стороне сервера. Пароль хранится в зашифрованном виде, обеспечивая соответствие требованиям информационной безопасности.

Связь между users.role\_id и roles.id определяет поведение интерфейса и доступный функционал для конкретного пользователя. Это позволяет реализовать разграничение прав на уровне клиентской и серверной логики: студенты могут просматривать вакансии и откликаться на них, работодатели — публиковать предложения, а администратор — управлять пользовательскими данными.

На этапе авторизации осуществляется проверка на соответствие введённого email и хешированного пароля с данными в таблице users. В случае успешной аутентификации платформа определяет роль пользователя и направляет его к соответствующему разделу системы. Сессионная информация или токены аутентификации могут использоваться для поддержания доступа на протяжении определённого периода.

Таблицы, необходимые для создания пользователя показаны на рисунке 9

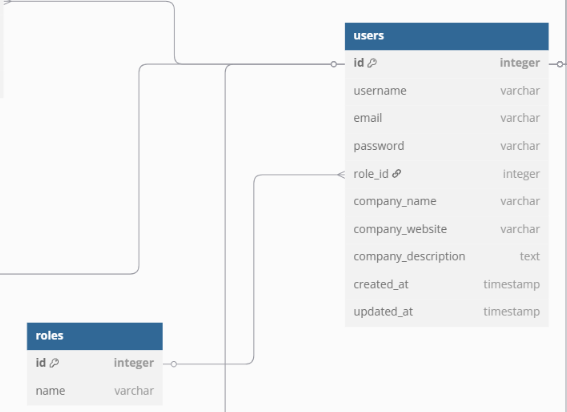


Рисунок 9 – Модели таблицы пользователей и ролей

Процесс добавления новой вакансии представляет собой один из ключевых этапов в работе работодателя с онлайн-сервисом и лежит в основе взаимодействия между организациями и студентами. Он позволяет формировать структурированное описание стажировок, устанавливать критерии отбора, настраивать параметры отображения и сопровождать предложения дополнительной информацией. В рамках данного процесса задействованы несколько таблиц, основными из которых являются: jobs, job\_requirements, job\_responsibilities, job\_tags, а также вспомогательные tags и users.

Таблица jobs является центральной сущностью в механизме публикации вакансий и содержит полное описание предложения работодателя. Она включает в себя 22 поля, каждое из которых отвечает за определённый аспект:

1. id — уникальный идентификатор вакансии;
2. title — заголовок вакансии, отображаемый в общем списке;
3. description — расширенное описание задач и условий прохождения стажировки;
4. category — необязательное поле, отражающее тематическое направление;
5. location — местоположение стажировки (город или регион);
6. remote — булево значение, указывающее на возможность удалённого формата;
7. salary — при необходимости, указанная сумма оплаты (при наличии);
8. employer\_id — внешний ключ, связывающий вакансию с аккаунтом работодателя из таблицы users;
9. test\_id — опциональное поле, указывающее на связанный тест для прохождения отбора (если задан);
10. posted\_at — дата и время публикации;
11. deadline — крайний срок подачи откликов;
12. status — текущий статус публикации: draft или published;
13. work\_schedule — тип графика (например, полный день, гибкий график);
14. employment\_type — форма занятости (например, стажировка, частичная занятость);
15. experience\_level — ожидаемый уровень подготовки;
16. education\_level — желаемый уровень образования кандидата;
17. benefits — текстовое поле с описанием преимуществ;
18. mentor\_support — флаг, указывающий, будет ли доступна поддержка наставника;
19. certificate — наличие сертификата по результатам стажировки;
20. possibility\_of\_employment — возможность дальнейшего трудоустройства;
21. paid — информация о возможности получать деньги в ходе прохождения стажировки (булевое значение);
22. created\_at — дата создания записи.

При добавлении вакансии работодатель предоставляет обязательные сведения, а также может дополнительно указать требования и обязанности.

Для формирования требований и ожиданий от кандидата используются вспомогательные таблицы:

Таблица job**\_**requirements содержит перечень требований к соискателю. Каждая запись в таблице включает в себя следующие обязательные поля:

1. id — идентификатор требования;
2. job\_id — ссылка на вакансию;
3. text — текстовое описание требования от студента, рассчитывающего пройти на стажировку/практику;
4. order\_index — порядок отображения в интерфейсе.

Таблицаjob**\_**responsibilities аналогично используется для описания задач, которые предстоит выполнять студенту в рамках стажировки:

1. id — идентификатор задачи;
2. job\_id — ссылка на вакансию;
3. text — описание обязанности, которые придется выполнять студенту в ходе прохождения практики/стажировки;
4. order\_index — порядок отображения в интерфейсе.

Таблицаtags содержит предварительно определённые ключевые теги, описывающие профессиональные области и навыки (например, JavaScript, UX/UI, Backend). Вакансии могут быть снабжены несколькими тегами через промежуточную таблицуjob**\_**tags, которая реализует связь «многие-ко-многим» между вакансиями и тегами:

* job\_id — ссылка на вакансию;
* tag\_id — идентификатор тега.

Такое разделение структуры вакансии на основную и вспомогательные таблицы обеспечивает гибкость в описании предложений, возможность расширения функционала (например, через фильтрацию по тегам и направлениям подготовки), а также соответствует принципам нормализации и удобства представления данных.

Работодатель может создать черновик вакансии и впоследствии опубликовать его, при этом все связанные записи остаются привязанными к основному job\_id. Связь с таблицей users осуществляется через поле employer\_id, что обеспечивает однозначную принадлежность записи к конкретному работодателю.

Таблицы, которые необходимы для возможности создания вакансии показаны на скриншотах 10 и 11

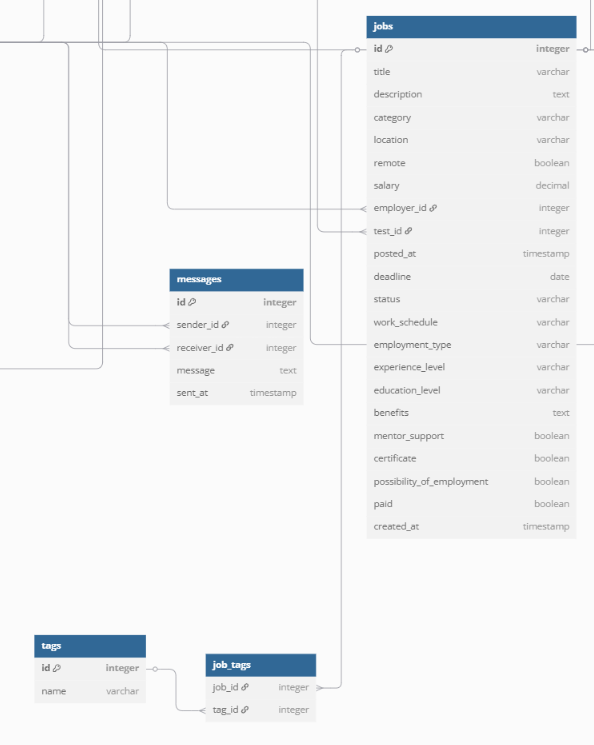


Рисунок 10 – Таблицы jobs, tags и job\_tags

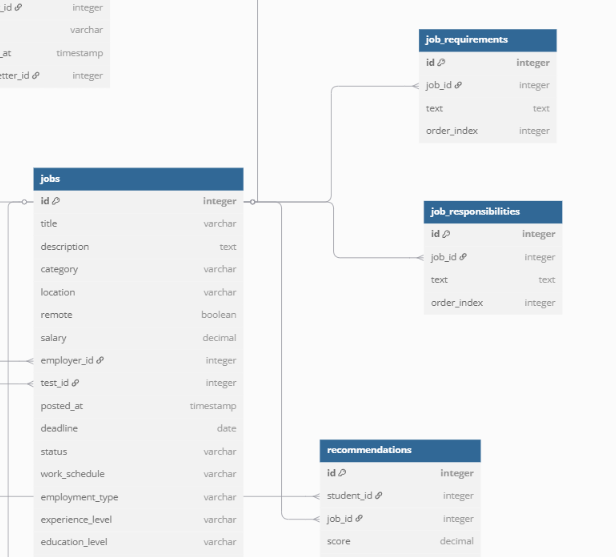


Рисунок 11 – Связь между таблицей вакансий и таблицами обязанностей и задач

Функциональность тестирования в рамках онлайн-сервиса служит средством предварительной оценки знаний и навыков студента, а также дополнительно повышает релевантность отбора на стажировку. Механизм реализован через совокупность нескольких таблиц: tests, test\_questions, test\_answers, test\_results.

Таблицаtests содержит обобщённую информацию о созданных тестах, которые разрабатываются работодателями и прикрепляются к вакансиям. Основные поля:

1. id — уникальный идентификатор теста;
2. employer\_id — внешний ключ на работодателя, который создал тест;
3. title — наименование теста (например, «Проверка базовых знаний JavaScript»);
4. description — необязательное поле с пояснением или инструкциями;
5. created\_at — временная метка создания.

Каждый тест состоит из одного или нескольких вопросов, хранящихся в таблице test\_questions.

Таблицаtest**\_**questions представляет собой перечень вопросов, входящих в конкретный тест:

* id — уникальный идентификатор вопроса;
* test\_id — внешний ключ на тест;
* text — тест вопроса.

К каждому вопросу может быть привязано несколько вариантов ответа, которые представлены в таблице test\_answers.

Таблицаtest**\_**answers содержит варианты ответов на конкретный вопрос. Поля включают:

* id — идентификатор ответа;
* question\_id — связь с вопросом;
* text — текст ответа;
* is\_correct — булево значение, определяющее правильность ответа.

Таким образом, каждый вопрос может иметь как один, так и несколько корректных вариантов.

Таблицаtest**\_**results фиксирует результаты прохождения тестирования студентами:

* id — уникальный идентификатор результата;
* test\_id — внешний ключ на соответствующий тест;
* student\_id — студент, прошедший тест;
* score — итоговый балл (в процентах или абсолютном выражении);
* submitted\_at — дата и время прохождения теста.

На рисунках 12-13 изображены модели таблиц, которые требуются для создания и прохождения тестов

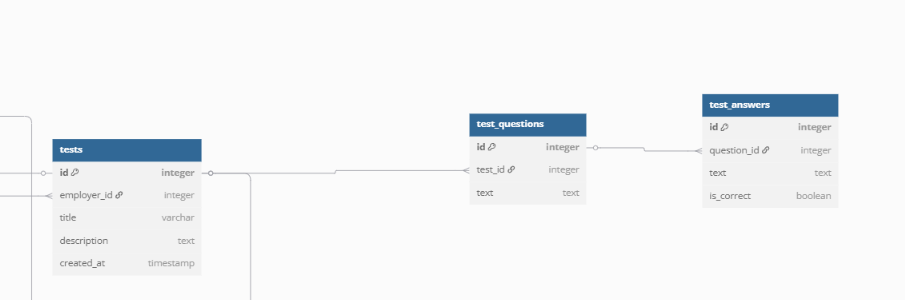


Рисунок 12 – Таблицы тестов, вопросов и ответов

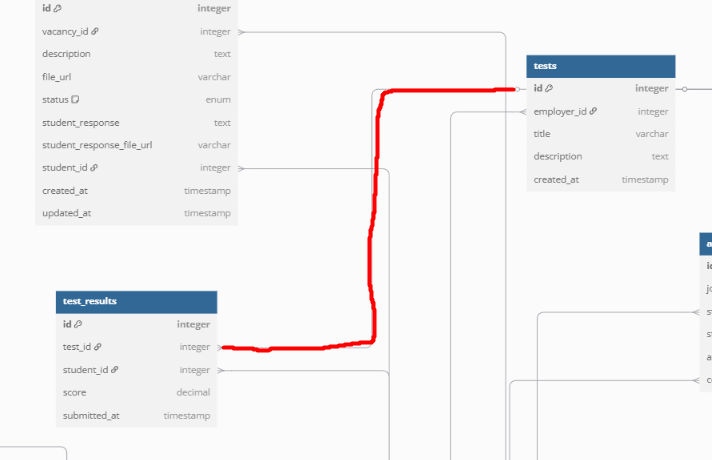


Рисунок 13 – Связь между таблицей тестов и результатами ее прохождения

Для фиксации ключевых этапов взаимодействия студентов и работодателей после подачи заявки на вакансию используется таблица applications, а также вспомогательные таблицы, отражающие выполнение заданий, коммуникацию и рекомендации.

Таблицаapplications фиксирует сам факт отклика студента на опубликованную вакансию. Каждая запись таблицы включает в себя поля:

1. id — уникальный идентификатор;
2. job\_id — идентификатор вакансии, на которую отправлен отклик;
3. student\_id — идентификатор пользователя с ролью студента;
4. status — статус заявки (например: applied, interview, accepted, rejected);
5. applied\_at — временная метка подачи заявки;
6. cover\_letter\_id — внешний ключ на сопроводительное письмо (при наличии).

Процесс прохождения стажировки предполагает, что работодатель может отслеживать отклики, изменять их статус и формировать представление о прогрессе.

Таблицаtasks позволяет работодателям назначать конкретные задания для студентов в рамках стажировки. Каждое задание связано с конкретной вакансией и студентом и включает:

* Описание задания, файл (если требуется), статус выполнения;
* Ответ студента в виде текста и/или файла;
* Временные метки и связи с таблицами jobs и users.

Это решение обеспечивает цифровое сопровождение процесса стажировки, позволяет отслеживать выполнение задач и организует прозрачную структуру обратной связи.

Дополнительно процесс сопровождения включает в себя следующие таблицы:

* messages — для отправки текстовых сообщений между студентом и работодателем;
* notifications — для автоматических оповещений о новых событиях (например, изменение статуса заявки или назначения задания);
* recommendations — для формирования индивидуальных подборок вакансий на основе профиля студента и истории взаимодействий.

Таблицы по отклику и прохождению стажировок показаны на рисунках 12-13

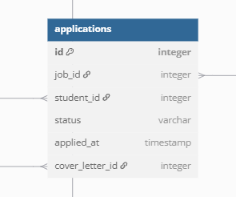


Рисунок 12 – Таблица откликов

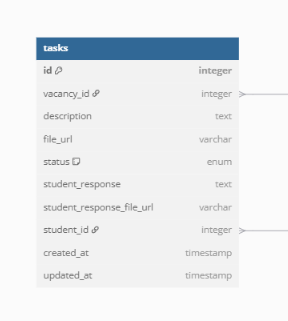


Рисунок 13 – Таблица заданий

Таким образом, разработанная архитектура базы данных и её таблицы формируют ядро цифрового решения, предназначенного для эффективного взаимодействия студентов и работодателей в контексте организации практик и стажировок. Онлайн-сервис опирается на принципы логической связности, ролевого разграничения доступа, прозрачности пользовательских действий и формализации образовательной активности. Реализованные сущности охватывают все ключевые участки жизненного цикла взаимодействия: от создания профиля до назначения заданий и фиксации результатов прохождения стажировок.

Функциональная полнота структуры отражает современные требования к информационным системам в сфере образования и трудоустройства, обеспечивая как удобство пользователей, так и потенциальную интеграцию с внешними сервисами. Подобный подход не только оптимизирует процесс поиска практики для студентов, но и создаёт единое цифровое пространство с возможностью дальнейшего расширения, сопровождения и анализа результатов профессионального становления обучающихся.

## 2.3 Система рекомендаций и персонализации

Механизм персонализации является важным компонентом онлайн-сервиса и направлен на повышение релевантности отображаемого контента для студентов. Основная задача — минимизировать ручной поиск подходящих стажировок, предлагая пользователю только те вакансии, которые соответствуют его профилю, интересам и образовательной траектории.

Система рекомендаций реализуется на стороне серверной части и основана на заранее собранных данных о пользователях, вакансиях и истории откликов. В архитектуре сервиса предусмотрена специальная таблица recommendations, в которой хранятся сформированные для студента предложения. Каждая запись включает идентификатор студента, идентификатор рекомендованной вакансии, рассчитанный рейтинг соответствия и временную метку генерации. Это позволяет системе регулярно обновлять рекомендации и поддерживать их актуальность.

Алгоритм формирования рекомендательной выборки использует несколько источников информации:

* Направление подготовки — ключевой параметр, определяющий релевантность вакансий студенту, используется для сопоставления образовательной специализации с категорией вакансии, тегами и требованиями к кандидату;
* Теги и технологии — учитываются на основе информации, указанной в профиле или ранее просмотренных вакансиях, и позволяют находить предложения, соответствующие навыкам студента, таким как «JavaScript», «UI/UX», «Backend» и другим;
* История откликов — формирует предпочтения на основе ранее выбранных вакансий, анализируя схожие параметры: направление, формат участия, территориальное расположение и уровень требований;
* Тип занятости и формат стажировки — используется для приоритезации вакансий в зависимости от предпочтений студента по графику, месту прохождения практики (очная, дистанционная, гибридная) и типу занятости;
* Популярность и новизна вакансий — повышает приоритет тех предложений, которые были недавно размещены, активно просматривались и получили положительные отклики от студентов с аналогичными характеристиками профиля;

На стороне клиента рекомендуемые вакансии отображаются в отдельном блоке — «Вам может подойти». Это повышает вовлеченность пользователей и снижает время, необходимое для нахождения подходящего варианта стажировки.

Таким образом, система рекомендаций выступает как один из важнейших механизмов персонализации взаимодействия, обеспечивая соответствие интересов студентов содержанию предложений и способствуя более целенаправленному профессиональному старту. С учётом возможности масштабирования и внедрения новых алгоритмов (например, на основе анализа активности или машинного обучения), данный модуль платформы может быть расширен в дальнейшем без изменения основного ядра системы.

## 2.4 Выводы по второй главе

В результате проведённого проектирования была разработана системная концепция проектирования онлайн-сервиса, ориентированного на сопровождение студентов в процессе поиска практик и стажировок. Последовательно были проанализированы основные аспекты архитектуры и логики взаимодействия между ключевыми участниками процесса — студентами, работодателями и администрацией системы. Особое внимание было уделено структурной организации функционала, распределению ролей пользователей, возможностям их интерфейсного и логического взаимодействия, а также подходу к персонализации и автоматизации подбора подходящих вакансий.

Сервис спроектирован таким образом, чтобы обеспечивать высокий уровень прозрачности бизнес-процессов, поддержку двустороннего взаимодействия и формализацию действий пользователей. Реализация механизма фильтрации, откликов, статусов, системы рекомендаций, уведомлений и административного контроля позволила заложить технологический фундамент для создания масштабируемого и жизнеспособного решения.

Значительное внимание было также уделено проектированию базы данных — как опорной части всей информационной системы. Сформированная структура хранения данных отражает основные сценарии использования сервиса, обеспечивает логическую целостность и учитывает специфику образовательной сферы.

В результате выполненного проектирования была получена целостная архитектурная и функциональная модель онлайн-сервиса, способная удовлетворить актуальные потребности студентов и работодателей, и одновременно обеспечивающая предпосылки для дальнейшего развития системы, в том числе в направлении аналитики, интеграции и адаптации под расширенные цели цифровой трансформации.

# 3. Реализация онлайн-сервиса

## 3.1 Стилистические решения онлайн-сервиса

Одним из факторов удобства взаимодействия пользователей с онлайн-сервисом выступает единообразие визуального оформления и продуманная организация пользовательского интерфейса. В рамках разработки внимание было уделено формированию современного, интуитивно-понятного и лаконичного вида, адаптированного под различные задачи, с которыми будут сталкиваться пользователи.

Цветовая схема онлайн-сервиса реализована в двух вариантах: светлая и темная тема, между которыми пользователь может переключаться в зависимости от предпочтений. Светлая тема оформлена в нейтральных и контрастных тонах, а темная построена на инверсии этих цветов и сохраняет визуальный акцент, что обеспечивает высокую контрастность и адаптацию под условия слабого освещения. Цветовая схема показана на рисунке 14



Рисунок 14 – Цветовая гамма онлайн-сервиса

В качестве основного шрифта был выбран Arial, sans-serif. Он обеспечивает кросс-браузерную совместимость и высокую читаемость текста. Заголовки различных уровней оформлены с четкой иерархией размеров от 2.5rem (h1) до 1rem (h6), где основной текст идет размером 1rem с межстрочным интервалом 1.6, что оптимально для восприятия информации. UI Kit шрифтов показан на рисунке 15

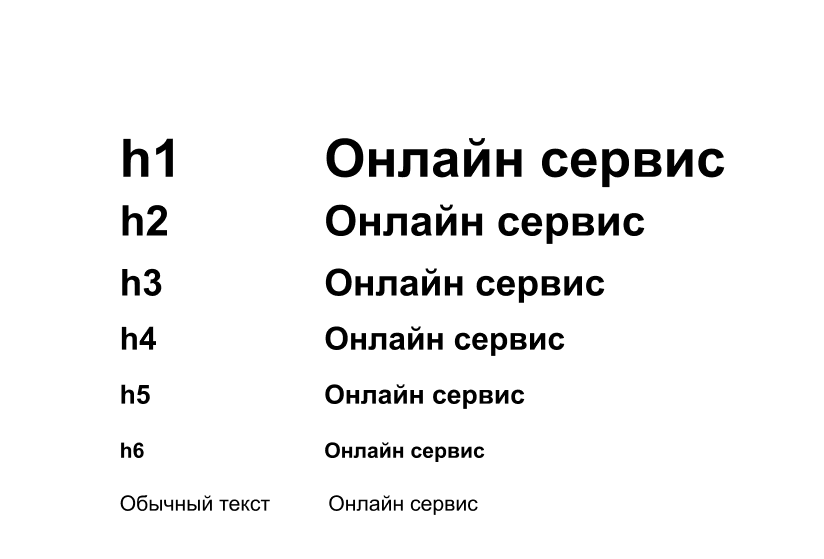


Рисунок 15 – UI Kit шрифтов

Компоненты пользовательского интерфейса разработаны с использованием библиотеки PrimeVue и ее темы Aura, что позволило обеспечить единый визуальный стиль и ускорить разработку. Кнопки имеют строгие различия между друг другом: основные действия выделяются зелёным цветом, второстепенные — синим, кнопки удаления и отказа — красным. Для всех элементов используется одинаковый радиус скругления (5px), плавные отступы и адаптивная ширина, соответствующая контексту. Карточки информации (например, вакансий) дополнительно оформлены с тенью, округлением 8px и анимацией при наведении, что подчёркивает интерактивность и делает интерфейс визуально динамичным. Формы ввода стилизованы в соответствии с общей стилисткой остального контента сайта: они имеют полную ширину, скругление, тонкую рамку и отступы, позволяющие удобно вводить данные как на большом мониторе, так и на мобильных устройствах.

Дополнительные визуальные особенности включают в себя добавления различных SVG-иконок, плавных переходов, которые длятся в среднем около 0.3 секунды, анимации кнопок, открытия различных элементов, например, модальные окна или ответ на вопрос в секции часто-задаваемых вопросах. Также внимание было уделено визуальным статусам заявок и заданий, которые оформлены в виде цветовых индикаторов и с текстовым пояснением, что помогает в восприятии текущего состояния задания или процесса.

Таким образом, визуальное оформление онлайн-сервиса выполняет не только эстетическую, но также функциональную роль, повышая удобство работы с системой.

## 3.2 Реализация серверной логики и API

Серверная часть онлайн-сервиса реализована на базе фреймворка Express.js и обеспечивает логику взаимодействия между клиентом и базой данных. Главной задачей backend-модуля является обработка HTTP-запросов, выполнение бизнес-логики, обеспечение безопасности и стабильной передачи данных. В качестве базы данных используется MySQL, что позволило организовать чёткую структуру хранения информации и надежное выполнение транзакций.

Архитектура backend-части построена по модульному принципу: каждый блок — аутентификация, вакансии, отклики — вынесен в отдельные файлы контроллеров и маршрутов [19]. Это упрощает поддержку и масштабирование проекта. Запросы проходят через цепочку промежуточных обработчиков (middleware), отвечающих за логирование, проверку аутентификации, валидацию данных и обработку ошибок. Структура backend части показана на рисунке 16

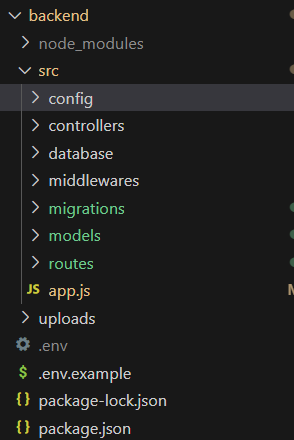


Рисунок 16 – Структура backend части

Для реализации механизма авторизации в онлайн-сервисе используется технология JSON Web Token (JWT), представляющая собой современный и безопасный способ передачи идентификационных данных между клиентом и сервером. После успешного прохождения аутентификации (проверки почты и пароля на наличие), сервер формирует цифровой токе, содержащий зашифрованную информацию о пользователе и сроке действия конкретной сессии. Этот токен возвращается клиенту и сохраняется на стороне пользователя в локальном хранилище браузера.

При последующих обращениях к защищенным маршрутам клиент автоматически прикрепляет полученный JWT к заголовкам HTTP-запросов. Сервер в свою очередь осуществляет проверку подлинности токена, расшифровывает его содержимое и, при подтверждении действительности, предоставляет доступ к запрашиваемому ресурсу. В случае недействительного, отсутствующего или просроченного токена – доступ к некоторым маршрутам блокируется и часть функционала становится недоступной.

Обработка безопасности сосредоточена в middleware. Здесь реализована проверка JWT-токенов, извлечение ролей пользователей, разграничение доступа по ролям, валидация данных при помощи схем, централизованная обработка ошибок и логирование. Все ошибки, возникающие при выполнении запросов, регистрируются и обрабатываются с возвратом понятных сообщений пользователю. Система использует bcrypt для безопасного хранения паролей, CORS для ограничения внешнего доступа и дополнительные настройки заголовков безопасности. Проверка на SQL-инъекции и XSS реализована валидацией входных данных и ограничением потенциально опасных операций.

Для повышения удобства работодателей и снижения временных затрат при формировании контента предусмотрена интеграция с языковой моделью GPT, выполняющей вспомогательные функции при создании тестов и заданий. Пользователь может ввести тему, общее описание компетенций или примерную формулировку, после чего система на основе модели формирует задание или набор вопросов с вариантами ответов. Предусмотрена возможность редактирования сгенерированного текста вручную, что позволяет сохранить контроль за содержанием и соответствие требованиям конкретной вакансии.

Применение данного механизма позволяет обеспечить гибкость при разработке проверочных материалов, ускорить процесс генерации типовых заданий, а также повысить методическую насыщенность тестов. В перспективе такая функция может быть дополнена адаптацией под профиль вакансии и анализом полноты сформулированных требований. Использование языковой модели осуществляется строго в рамках вспомогательных сценариев, без доступа к персональным данным пользователей, что соответствует требованиям безопасности и конфиденциальности.

Описание всех основных реализованных методов API сервера показаны в таблице 5

Таблица 5 – Список методов API онлайн сервиса

| Тип метода | Вызов метода | Описание метода |
| --- | --- | --- |
| POST | /api/auth/register | Регистрация нового пользователя |
| POST | /api/auth/login | Вход в систему |
| GET | /api/users/:id | Получение информации о пользователе по ID |
| GET | /api/jobs | Получение списка всех вакансий |
| GET | /api/jobs/:id | Получение информации о конкретной вакансии |
| POST | /api/jobs | Создание новой вакансии |
| PUT | /api/jobs/:id | Обновление информации о вакансии |
| DELETE | /api/jobs/:id | Удаление вакансии |
| GET | /api/applications/:id | Получение информации о заявке по ID |
| PUT | /api/applications/:id/status | Обновление статуса заявки |
| GET | /api/tests | Получение списка всех тестов |
| GET | /api/tests/:id | Получение информации о конкретном тесте |
| GET | /api/tests/:id/take | Прохождение теста |
| POST | /api/tests | Создание нового теста |
| PUT | /api/tests/:id | Обновление теста |
| DELETE | /api/tests/:id | Удаление теста |
| POST | /api/tests/:testId/questions | Добавление вопроса к тесту |
| PUT | /api/questions/:id | Обновление вопросов |
| DELETE | /api/questions/:id | Удаление вопроса |
| POST | /api/questions/:questionId/answers | Добавление ответа к вопросу |
| PUT | /api/answers/:id | Редактирование ответа |
| DELETE | /api/answers/:id | Удаление ответа |
| GET | /api/tags | Получение списка всех тегов |
| GET | /api/tasks | Получение списка заданий для вакансии |
| POST | /api/tasks | Создание нового задания |
| POST | api/tasks/:taskId/response | Отправка ответа на задание |
| POST | /api/tasks/:taskId/approve | Принятие задания |
| POST | /api/tasks/:taskId/reject | Отклонение задания |
| POST | /api/tasks/:taskId/reject-and-block | Отклонение и блокировка задания |

Таким образом, серверная часть онлайн-сервиса реализует модульную, масштабируемую и безопасную структуру взаимодействия. Она обеспечивает корректную работу всех пользовательских ролей, обрабатывает данные в соответствии с бизнес-логикой и служит надежной основой всего сервиса трудоустройства студентов.

## 3.3 Реализация клиентской части приложения

Бизнес-логика клиентской части онлайн-сервиса реализует ключевые пользовательские сценарии, обеспечивая последовательность взаимодействий с интерфейсом и корректную передачу данных на сервер. В рамках интерфейса обеспечена поддержка всех этапов пользовательского пути — от регистрации и входа в систему до отклика на вакансии и отслеживания статусов заявок. Каждое действие пользователя инициирует соответствующие запросы к серверу, формирует состояние компонентов интерфейса и управляется через механизмы маршрутизации и локального хранилища.

Процесс регистрации реализован через визуальную форму, в которой пользователь вводит адрес электронной почты, пароль и выбирает соответствующую роль, а также если выбрана роль представителя компании – вводятся сразу дополнительная информация о ней. После ввода – происходит отправка данных на сервер в виде POST-запроса, где проходит валидация и шифрование пароля. При успешной регистрации интерфейс автоматически перенаправляет пользователя к форме входа.

В процессе авторизации пользователь вводит учётные данные, и после успешной проверки от сервера получает access- и refresh-токены. Access-токен сохраняется в памяти клиента и прикрепляется ко всем последующим защищённым запросам через заголовки авторизации. Refresh-токен используется для автоматического продления сессии без необходимости повторного входа. В клиентской части логика авторизации реализуется через middleware и перехватчики (интерцепторы), обеспечивающие проверку статуса токена и его автоматическое обновление при истечении срока действия. Регистрация и авторизация пользователей показана на рисунках 17-19

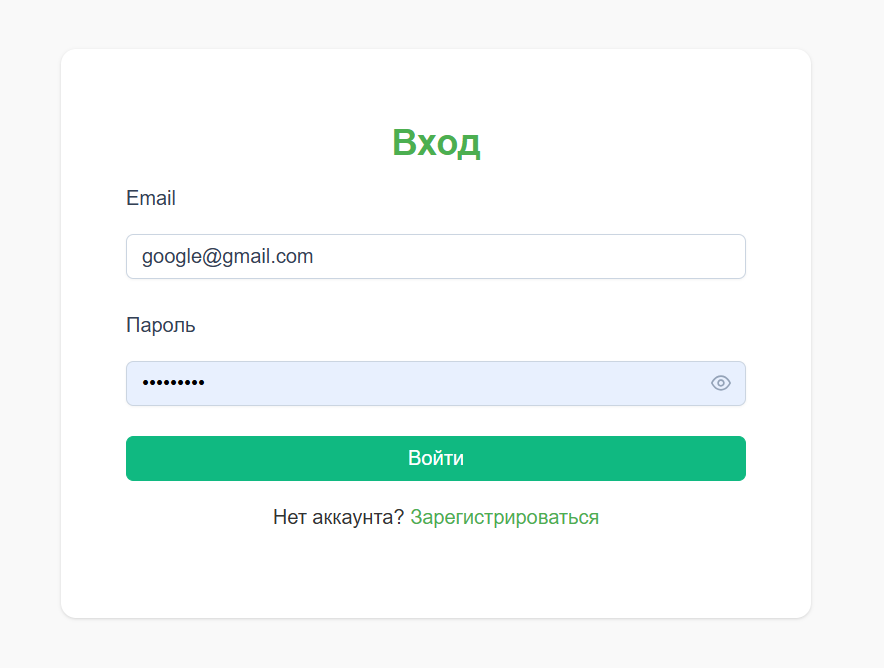


Рисунок 17 - Общая форма авторизации

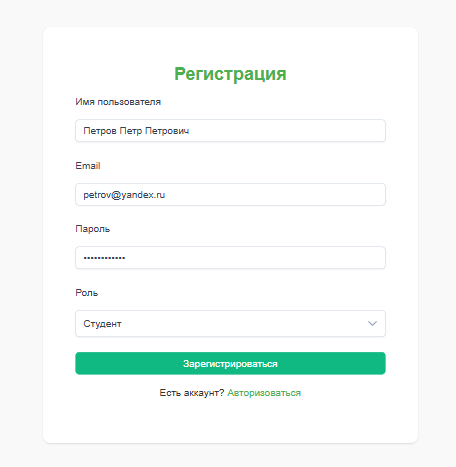


Рисунок 18 - Форма регистрации студента

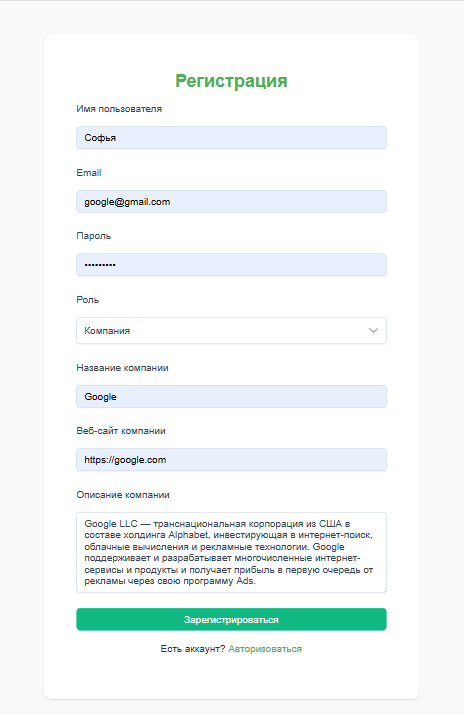


Рисунок 19 - Форма регистрации компании

После входа в систему каждый пользователь получает доступ к функциональности в соответствии со своей ролью. Для студента это просмотр доступных вакансий, возможность отклика, отслеживание статуса откликов. Для работодателя — создание и управление вакансиями, работа с откликами. Модератор имеет возможность просматривать и контролировать публикации, при необходимости блокировать нежелательный контент. Каждая операция проверяется с точки зрения прав доступа, а действия логируются.

Процесс авторизации в систему - является основой для формирования всех последующих сценариев. В дальнейшем на основе этих токенов осуществляется проверка прав и контроль доступа ко всем бизнес-операциям, от подачи откликов до изменения статусов откликов и работы с вакансиями.

После авторизации пользователь перенаправляется в личный кабинет, интерфейс которого адаптирован под его роль в системе. Для студентов отображается персонализированная панель с доступом к рекомендуемым вакансиям, списку отправленных откликов и текущему статусу заявок. Также предусмотрена возможность редактирования профиля, загрузки сопроводительных писем и отслеживания результатов прохождения тестов, если таковые предусмотрены вакансией.

Для работодателей личный кабинет предоставляет доступ к управлению опубликованными вакансиями. В соответствующем разделе отображается перечень всех созданных предложений с возможностью их просмотра, редактирования, удаления и приостановки публикации. Кроме того, реализована функция создания новой вакансии через пошаговую форму с указанием всех необходимых параметров: названия, описания, требований, формата прохождения практики и условий. Интерфейс позволяет также отслеживать поступившие отклики на каждую вакансию, просматривать сопроводительные документы студентов и изменять статус заявок. Личный кабинет пользователей показан на рисунках 20-21.

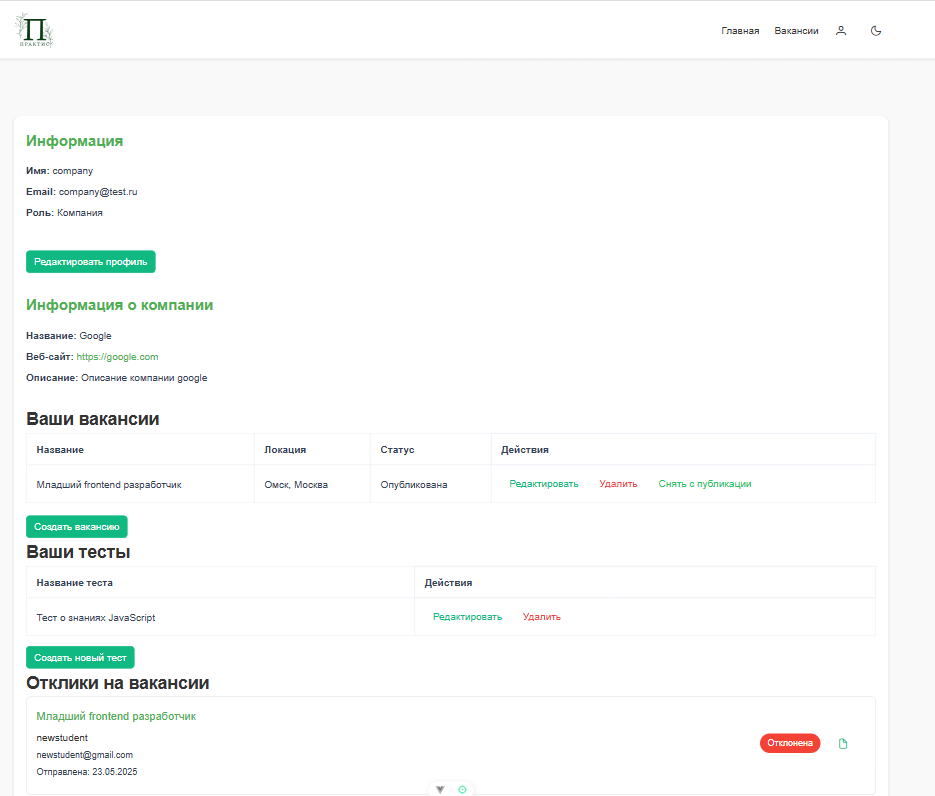


Рисунок 20 - Личный кабинет студента

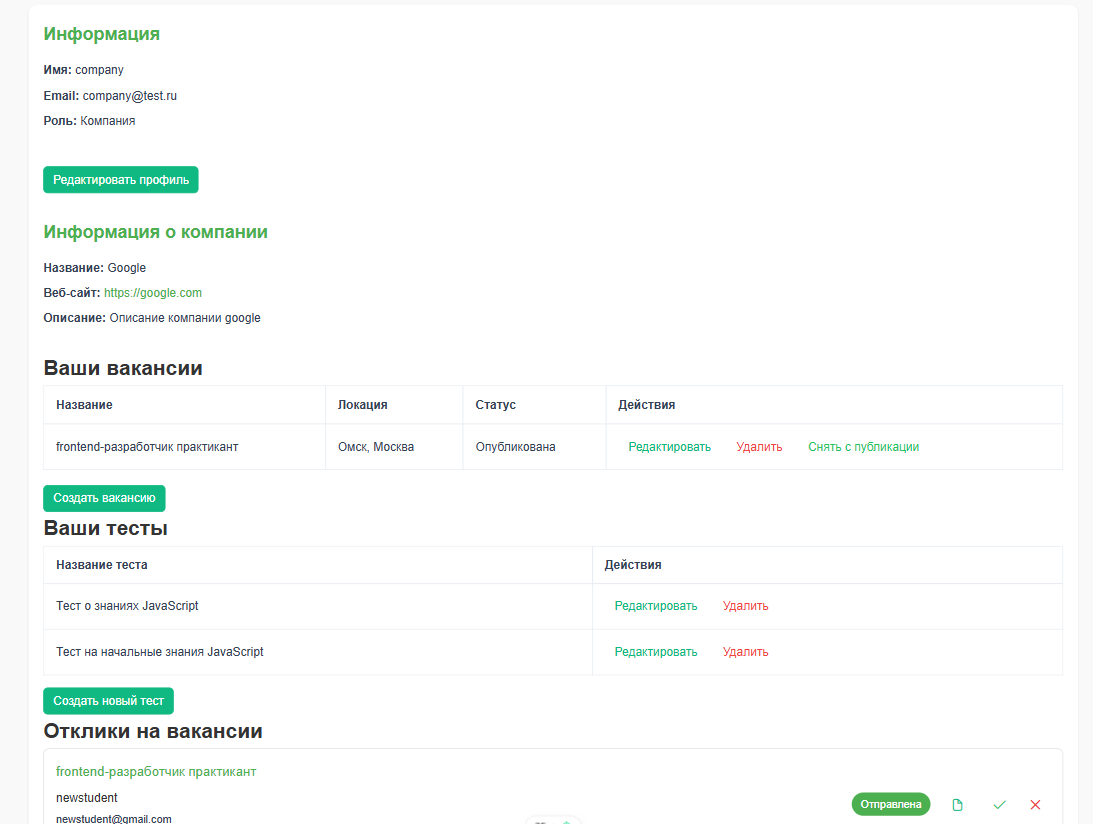


Рисунок 21 – Личный кабинет представителя компании

Процесс создания вакансии на стороне клиентской части начинается с перехода работодателя в соответствующий раздел личного кабинета и нажатия интерактивной кнопки «Добавить вакансию». После этого интерфейс динамически загружает форму создания вакансии, представляющую собой набор полей для ввода основной информации: названия должности, описания, направления подготовки, требований к студенту, условий стажировки, города и других параметров.

Валидация данных реализуется как на стороне клиента, так и с учетом серверных ограничений. В frontend-части предусмотрены проверки на заполненность обязательных полей, соответствие формата ввода (например, числовые значения, email, текстовые поля заданной длины), а также ограничения на длину описаний и выбор доступных опций. Ошибки отображаются в режиме реального времени, что позволяет пользователю корректировать данные до отправки формы.

После успешного заполнения формы все введённые данные собираются в структуру объекта и отправляются на сервер с помощью HTTP-запроса (чаще всего через метод POST). Передача осуществляется с прикреплением JWT-токена в заголовке авторизации, что необходимо для подтверждения роли пользователя как работодателя. В случае успешного ответа от сервера интерфейс автоматически уведомляет пользователя о создании вакансии и обновляет список доступных предложений, добавляя новую позицию в общий перечень. Форма, которая используется для добавления новой вакансии показана на рисунках 22 и 23

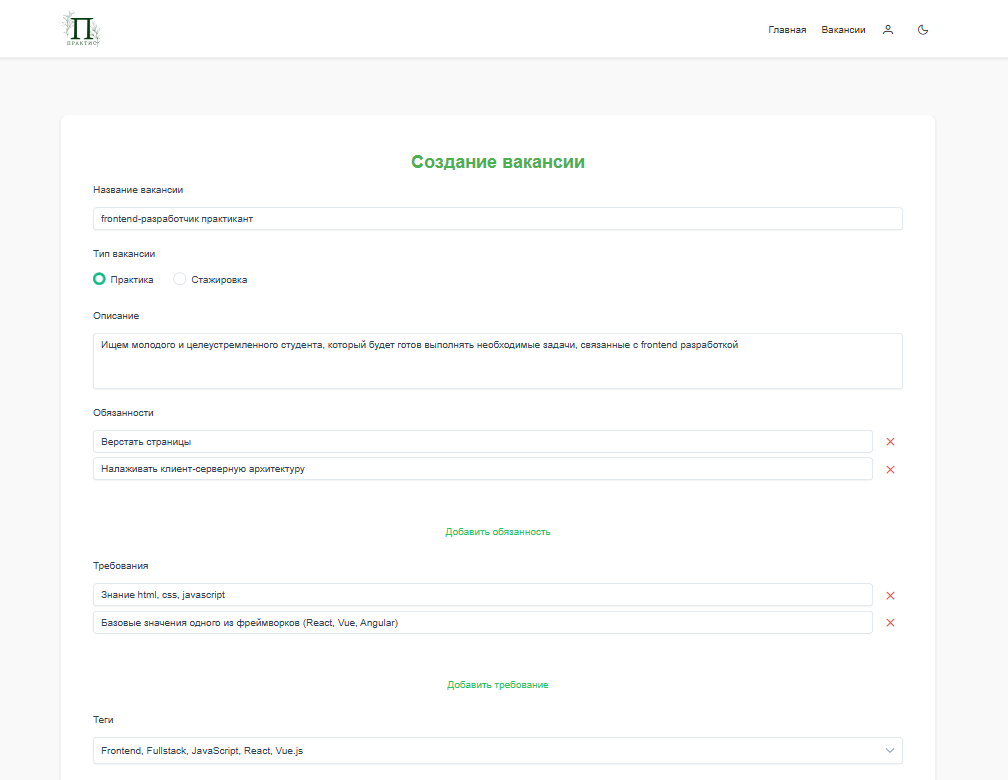


Рисунок 22 - Основные данные для создания вакансии

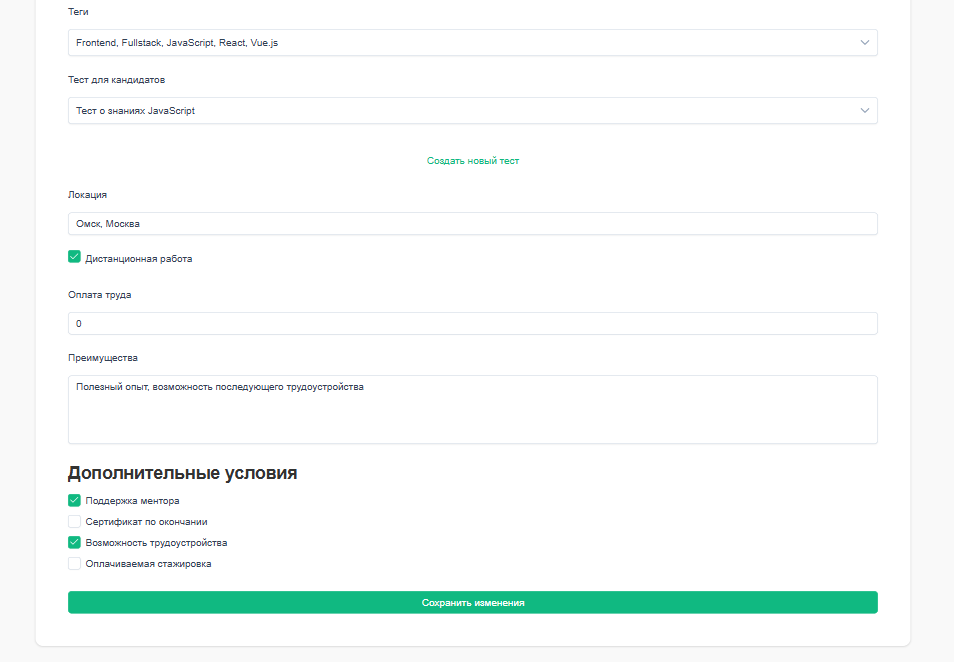


Рисунок 23 - Дополнительные требования для создания вакансии

Кроме базовой информации, работодатель может расширить вакансию, прикрепив к ней тестовое задание. Для этого можно создать тест предварительно в разделе «Тесты» личного кабинета компании, либо же кликнуть по ссылке «Создать новый тест», после чего произойдёт перенаправление представителя компании на страницу создания теста. Для редактирования теста необходимо перейти через личный кабинет к конкретному тесту, т.к. добавлять дополнительный функционал для редактирования теста на странице вакансии – не является необходимым. Создание теста представляет собой заполнение формы, включающей название теста и один или несколько вопросов. Для каждого вопроса указываются варианты ответов и отмечается правильный вариант (или несколько, если вопрос многовариантный). При необходимости, можно воспользоваться помощью чата GPT, на который будет отсылаться тема теста и в ответ будет приходить объект, содержащий в себе готовые вопросы и ответы, которые при необходимости можно отредактировать или отправить еще раз запрос, с уже более конкретными требованиями. Все данные отправляются на сервер, где осуществляется валидация и сохранение структуры теста в соответствующих таблицах базы данных. Один и тот же тест может быть использован в нескольких вакансиях — это реализовано через связующую таблицу, которая позволяет многократно привязывать уже созданный тест к новым объявлениям.

После создания теста работодатель может выбрать его при создании или редактировании вакансии. При этом серверная логика проверяет, принадлежит ли тест данному работодателю, существует ли он в базе и корректно ли сформирован. Если все условия соблюдены, тест прикрепляется к вакансии. В случае, если вакансия содержит тест, при отклике студенту будет предложено пройти его как часть отклика. Таким образом, работодатель получает дополнительный инструмент для первичной фильтрации кандидатов. Процесс создания теста показан на рисунке 24.

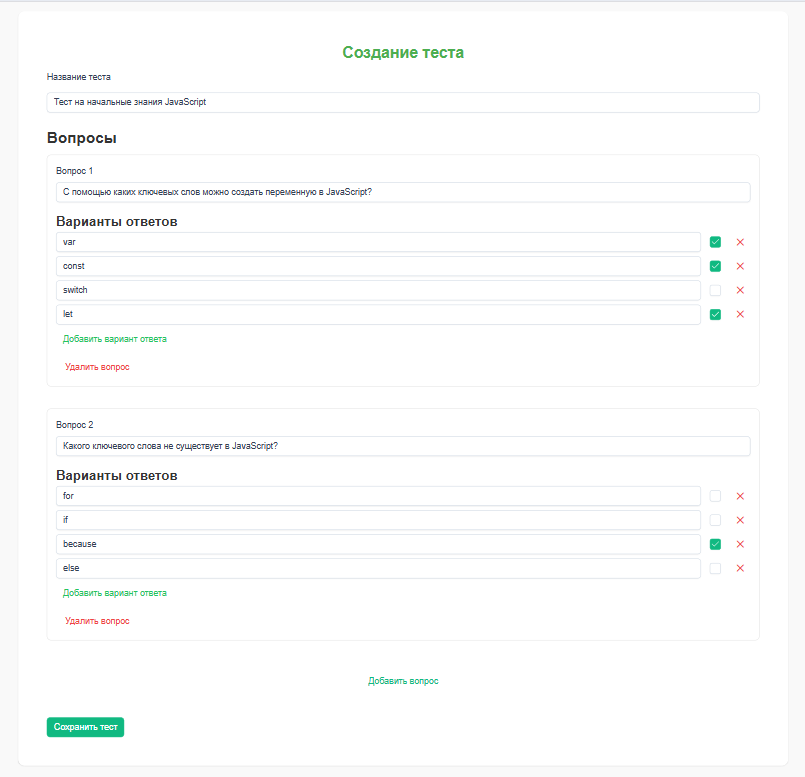


Рисунок 24 - Форма создания теста

Работодатель также может управлять уже созданными тестами: редактировать вопросы, добавлять или удалять пункты, а также полностью удалить тест. Все действия контролируются правами доступа и защищены от несанкционированных изменений.

После входа в систему студент получает доступ к основному разделу онлайн-сервиса — списку доступных вакансий. Интерфейс предоставляет удобную ленту с возможностью применения фильтров по направлению деятельности, городу, формату занятости (очно/удаленно), дате публикации и другим параметрам. Запросы, отправляемые при фильтрации, обрабатываются на сервере с учётом переданных параметров и возвращают только те вакансии, которые соответствуют условиям. Список доступных вакансий с фильтрами показан на рисунках 25 и 26.

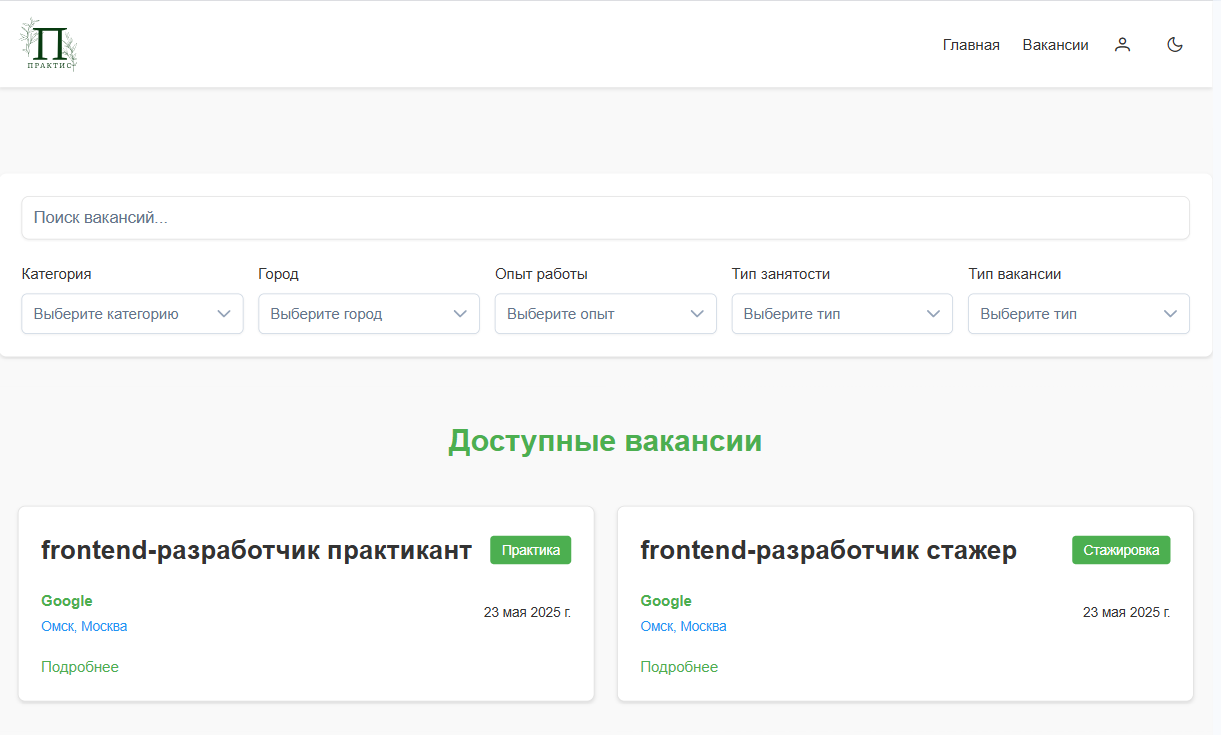


Рисунок 25 - Список вакансий без применения фильтров

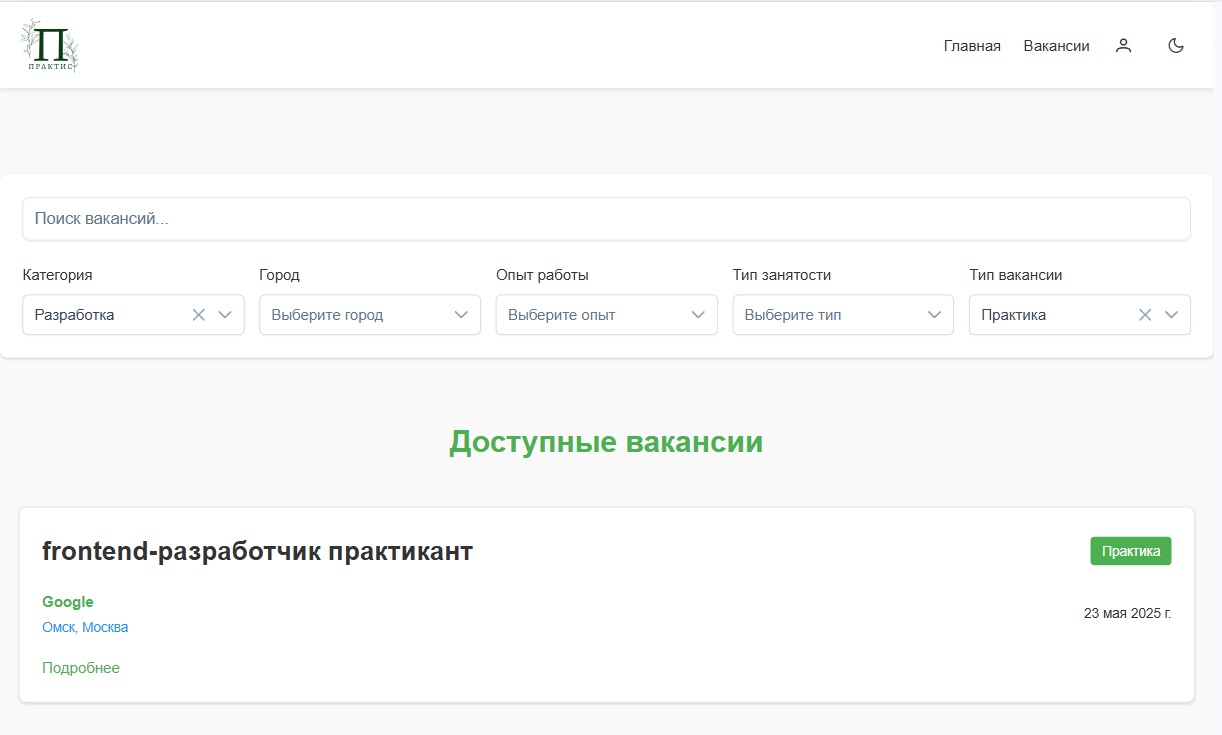


Рисунок 26 - Список вакансий с применением фильтров

При переходе к подробному описанию вакансии студент может ознакомиться с её содержанием: требования, условия, описание, а также наличие прикрепленного теста (если такой был добавлен работодателем). Отклик на вакансию возможен только при наличии авторизации и подтвержденной роли студента. Кнопка «Откликнуться» инициирует открытие формы отклика, в которой студент может оставить сообщение (в произвольной форме — например, мотивационное обращение). Это сообщение сохраняется как часть отклика и в дальнейшем доступно как работодателю, так и самому студенту. Отдельной сущности для сопроводительного письма не предусмотрено, что упрощает бизнес-логику и снижает избыточность данных. Страница вакансии показана на рисунке 27.

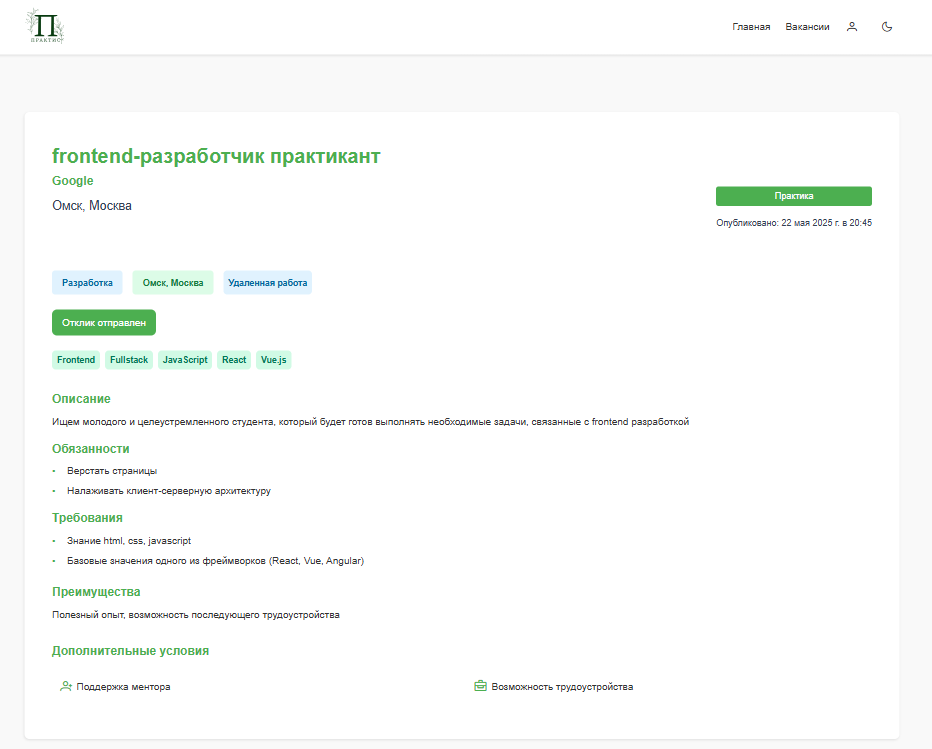


Рисунок 27 - Страница вакансии

Если к вакансии прикреплен тест, после отправки сообщения система автоматически перенаправляет студента на страницу прохождения теста. При успешном прохождении теста ответы фиксируются и становятся частью отклика.

После отправки отклика студент получает соответствующее уведомление и может отслеживать статус всех своих откликов в личном кабинете. Каждый отклик содержит информацию о вакансии, дате отклика, сопроводительном сообщении и текущем статусе: «рассматривается», «приглашен», «отказ» или «отклик отозван». Все статусы обновляются со стороны работодателя, а студенту приходит автоматическое уведомление при изменении на сайте. Страница студента после отправки отклика показана на рисунке 28

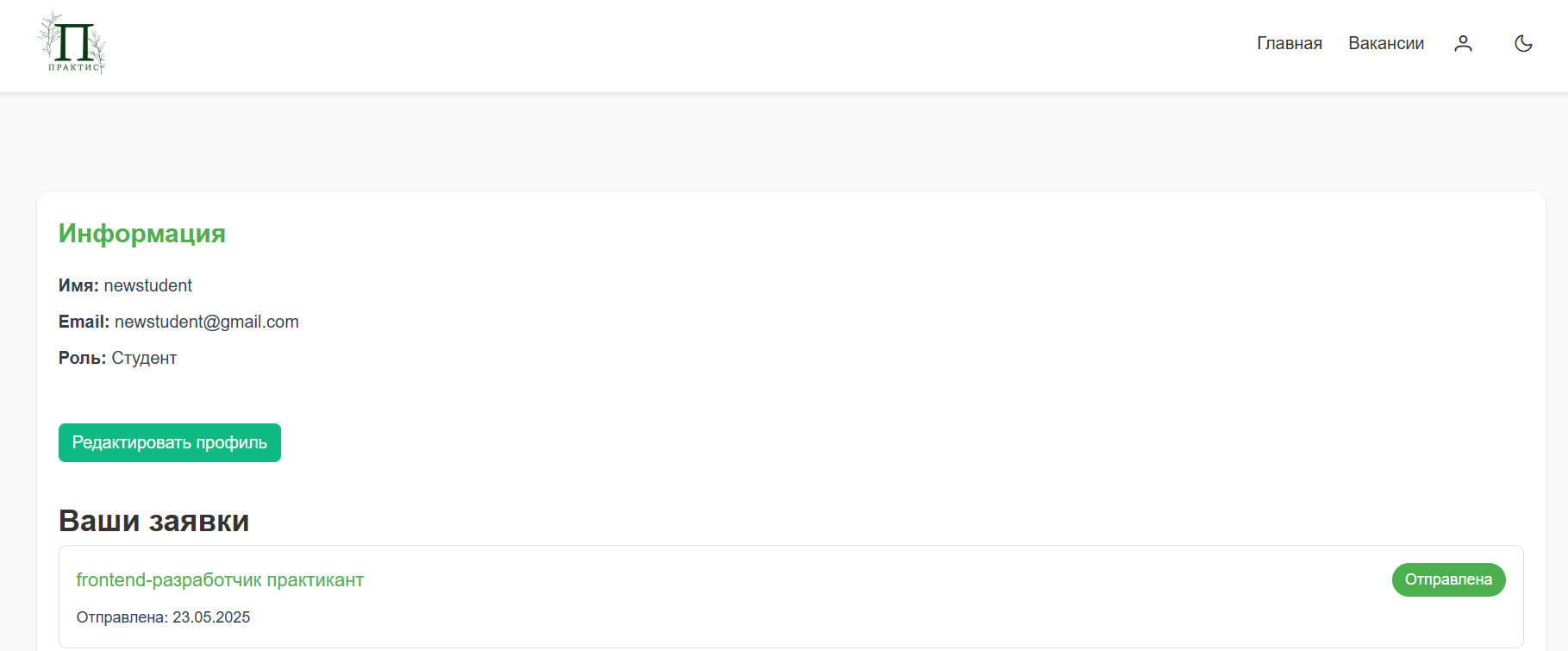


Рисунок 28 - Профиль студента после отправки отклика

Реализация клиентской части онлайн-сервиса отражает целенаправленный подход к обеспечению доступности, интуитивности и функциональности пользовательского взаимодействия. Разработка выполнена с использованием Vue.js, что позволило создать компонентно-ориентированную структуру интерфейса, легко масштабируемую и адаптируемую под различные пользовательские роли. Особое внимание уделено логике отображения, управлению состоянием и взаимодействию с сервером через REST API. Интерфейс обладает адаптивной версткой, поддержкой темной темы, интерактивными компонентами и системой уведомлений, что делает работу в сервисе максимально комфортной. Комбинация визуальной доступности, логической структурированности и технологической гибкости обеспечивает стабильную и современную пользовательскую среду, соответствующую целям проекта.

## 3.4 Выводы по третьей главе

В третьей главе была проведена комплексная проработка архитектурных, технических и логических основ реализации онлайн-сервиса, направленного на помощь в поиске и прохождению практики/стажировки студентам. Все аспекты разработки – от интерфейсных решений до маршрутов API – были созданы в соответствии с принципами модульности, безопасности и масштабируемости.

Серверная логика, построенная на базе Express.js и Node.js, реализует полноценную поддержку RESTful API с валидацией данных, защитой маршрутов и обработкой статусов. Все защищенные действия пользователей сопровождаются проверкой токенов доступа, что гарантирует соблюдение правил доступа к данным. Благодаря использованию JWT, было достигнуто надежное управление сессиями.

Клиентская часть онлайн-сервиса охватывает все ключевые этапы пользовательского взаимодействия: от создания и отклика на вакансию до прохождения стажировки и выполнения заданий. Все процессы сопровождаются логической проверкой, уведомлениями, сменой статусов и фиксацией действий. Применение модульной структуры позволило обеспечить независимость компонентов и упростить дальнейшее сопровождение.

**Заключение**

В результате выполнения выпускной квалификационной работы был разработан онлайн-сервис, ориентированный на содействие трудоустройству студентов и прохождение ими производственной практики.

В ходе исследования была проведена проработка проблемной области, что позволило выявить недостатки существующих цифровых решений, не учитывающих потребности обучающихся без опыта и небольших компаний.

В ходе анализа был сформулирован основной функционал сервиса. Определены как прикладные возможности системы (авторизация, регистрация, создание вакансий, отклики, рекомендации и пр.), так и технические ограничения, связанные с производительностью, безопасностью и масштабируемостью. В рамках проектной части была разработана архитектура онлайн-сервиса с использованием клиент-серверной модели, а также определена структура базы данных, обеспечивающая поддержку всех ключевых бизнес-процессов.

Клиентская часть системы реализована с использованием Vue.js, что позволило создать отзывчивый интерфейс с адаптацией под пользовательскую роль. Серверная логика разработана на базе Node.js с применением Express.js, что обеспечило надёжное взаимодействие между компонентами и позволило реализовать защищённые маршруты. Функциональные модули системы объединены в единое цифровое пространство, ориентированное на целевую аудиторию и практическое применение.

Все поставленные в начале работы цели и задачи достигнуты: изучена предметная область, сформулированы требования к системе, спроектированы архитектура и структура данных, реализован прототип сервиса.

# 

**Список используемых источников**

1. Трудовой кодекс Российской Федерации [Текст]: от 30.12.2001 № 197‑ФЗ (ред. от 07.04.2025) // Собрание законодательства РФ. – 07.01.2002. – № 1 (ч. 1). – Ст. 3
2. Федеральный закон «Об образовании в Российской Федерации» [Текст]: от 29.12.2012 № 273‑ФЗ (ред. от 25.07.2023) // Собрание законодательства РФ. – 02.01.2013. – № 1 (ч. 1). – Ст. 14
3. Федеральный закон «О персональных данных» [Текст]: от 27.07.2006 № 152‑ФЗ (ред. от 02.07.2021) // Собрание законодательства РФ. – 31.07.2006. – № 31. – Ст. 3561
4. ГОСТ 34.602–2020. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы. – Введ. 2021–01–01. – М.: Стандартинформ, 2021. – 21 с.
5. ГОСТ 19.201–78. Единая система программной документации. Техническое задание. – Введ. 1979–01–01. – М.: Изд-во стандартов, 1978. – 9 с.
6. Жукова Г. С., Борисова Л. Р., Седых И. Ю. Цифровые трансформации в современном образовании: монография. – М.: КноРус, 2023. – 198 с
7. Клячко Т. Л., Логинов Д. М. Трудоустройство молодежи с высшим образованием и уроки пандемии: монография. – М.: Дело, 2021. – 85 с
8. Долганова О. И., Виноградова Е. В., Лобанова А. М. Моделирование бизнес-процессов: учебник и практикум для вузов / под ред. О. И. Долгановой. – М.: Юрайт, 2020. – 289 с
9. Зуева А. Н. Моделирование бизнес-процессов в нотации BPMN 2.0: учебное пособие. – М.: РТУ МИРЭА, 2021. – 105 с
10. Прохоренок Н. А., Дронов В. А. JavaScript и Node.js для веб-разработчиков: учебное пособие. – СПб.: БХВ-Петербург, 2022. – 768 с
11. Дронов В. А. Node.js, Express, MongoDB и React: 23 урока для начинающих: учебное пособие. – СПб.: БХВ-Питер, 2024. – 608 с
12. Чистов Д. В. (под общ. ред. Д. В. Чистова) Проектирование информационных систем: учебник и практикум для вузов. – М.: Юрайт, 2021. – 258 с
13. Григорьев М. В., Григорьева И. И. Проектирование информационных систем: учебное пособие для вузов. – М.: Юрайт, 2022. – 318 с.
14. Brown I. Web-разработка с применением Node и Express: полное использование стека JavaScript : учебное пособие. – СПб.: Питер, 2021. – 336 с
15. Алексеев Д. В. MySQL. Практическое руководство по администрированию и разработке: учебное пособие / Д. В. Алексеев. – М.: Питер, 2021. – 384 с.
16. Иванов С. А. MySQL 8. Руководство для начинающих: учебное пособие / С. А. Иванов. – СПб.: БХВ-Петербург, 2023. – 320 с.
17. Долганова О. И. Бизнес-процессы: анализ, моделирование, технологии совершенствования: учебник. – М.: КноРус, 2025. – 336 с
18. Заботина Н. Н. Проектирование информационных систем: учебник. – М.: НИЦ Инфра‑М, 2024. – 331 с
19. Ананченко И. В., Войтюк Т. Е., Марченко Е. В. Архитектура информационных систем: учебное пособие. – СПб.: Университет ИТМО, 2024. – 58 с
20. ISO/IEC 25010:2014. Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models. – Geneva: International Organization for Standardization, 2014. – 31 с
21. Express/Node introduction [Электронный ресурс]. – URL: https://developer.mozilla.org/ru/docs/Learn\_web\_development/Extensions/Server-side/Express\_Nodejs/Introduction (дата обращения: 25.05.2025)
22. Краткое описание нотации BPMN [Электронный ресурс]. – URL: https://habr.com/ru/companies/auriga/articles/667084/ (дата обращения: 11.05.2025)
23. Нотация BPMN 2.0: ключевые элементы и описание [Электронный ресурс]. – URL: https://www.comindware.ru/blog/нотация-bpmn-2-0-элементы-и-описание/ (дата обращения: 14.05.2025)
24. Руководство по Angular [Электронный ресурс]. – URL: https://angular.io/docs (дата обращения: 13.05.2025)
25. Документация Vue.js [Электронный ресурс]. – URL: https://vuejs.org/guide/introduction.html (дата обращения: 18.05.2025)
26. Официальная документация React [Электронный ресурс]. – URL: https://reactjs.org/docs/getting-started.html (дата обращения: 19.05.2025)
27. Web Content Accessibility Guidelines (WCAG) 2.1 [Электронный ресурс] // World Wide Web Consortium (W3C). – Режим доступа: <https://www.w3.org/TR/WCAG21/> (дата обращения: 25.05.2025).