МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение  
высшего образования

«КРЫМСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ им. В. И. ВЕРНАДСКОГО»  
ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ

Кафедра компьютерной инженерии и моделирования

Круглеков Дмитрий Вячеславович

**ВЕБ-РЕСУРС ПОИСКА ПАРТНЕРОВ ДЛЯ ПРОЕКТОВ**

Выпускная квалификационная работа (уровень бакалавриата)

Направления подготовки 09.03.04 «Программная инженерия»

Научный руководитель В.В. Чабанов

Старший преподаватель (подпись, дата)

К ЗАЩИТЕ ДОПУСКАЮ:

Зав. кафедрой В.В. Милюков

Доцент, к. т. н. (подпись, дата)

Симферополь, 2023

**Реферат**

Круглеков Д.В. Веб-ресурс поиска партнеров для проектов: выпускная квалификационная работа: 09.03.04. / Физико-технический институт – Симферополь: КФУ им В.И. Вернадского, 2023.

Дипломная работа состоит из введения, 5 глав и заключения. При подготовке использовалось 12 источников. Общий объем работы составляет 38 страниц.

Объектом разработки является сайт, призванный помочь студентам начальных курсов с поиском партнеров для реализации собственных проектов.

Целью работы является разработка всех модулей: сайта, базы данных и telegram бота, также, получение новых навыков и пополнение портфолио.

При реализации проекта были использованы такие методы исследования, как анализ литературы, сравнительный анализ, анализ технологий. При разработке и тестировании использовался персональный компьютер с операционной системой Windows 10 и смартфон на базе операционной системы Android 10.

Все поставленные задачи были выполнены, получен полностью функциональный сервис, который готов к работе. При разработке были учтены проблемы подобных проектов и устранены в собственном.

Данный проект сможет заинтересовать студентов ВУЗа и будет мотивировать их разрабатывать проекты в команде.

РАЗРАБОТКА, ПОИСК ПАРТНЁРОВ, ВЕБ-РЕСУРС, ПРОЕКТИРОВАНИЕ, JAVASCRIPT, HTML, CSS, NODE

**Оглавление**

[Реферат 2](#_Toc138898221)

[Оглавление 3](#_Toc138898222)

[Список сокращений и условных обозначений 5](#_Toc138898223)

[Введение 6](#_Toc138898224)

[Глава 1. Анализ аналогов и актуальности темы 8](#_Toc138898225)

[1.1 Существующие аналоги 8](#_Toc138898226)

[1.2 Цель и задачи 11](#_Toc138898227)

[1.3 Научная новизна 11](#_Toc138898228)

[1.4 Теоретическая и практическая значимость работы 12](#_Toc138898229)

[1.5 Методы исследования 13](#_Toc138898230)

[1.6 Целевая аудитория 13](#_Toc138898231)

[Глава 2. Инструментальные средства разработки 14](#_Toc138898232)

[2.1 Описание проекта 14](#_Toc138898233)

[2.2 Обзор технологий 14](#_Toc138898234)

[2.2.1 Среда разработки 14](#_Toc138898235)

[2.2.2 Frontend 15](#_Toc138898236)

[2.2.3 Backend 15](#_Toc138898237)

[Глава 3. Разработка концептуальной модели сервиса 17](#_Toc138898238)

[3.1 Frontend сайта 17](#_Toc138898239)

[3.1.1 Верхняя информационная панель страницы 17](#_Toc138898240)

[3.1.2 Нижняя информационная панель страницы 17](#_Toc138898241)

[3.1.3 Главная страница 18](#_Toc138898242)

[3.1.4 Страница “Информация” 18](#_Toc138898243)

[3.1.5 Страница регистрации 18](#_Toc138898244)

[3.1.6 Страница входа 19](#_Toc138898245)

[3.1.7 Страница “Размещение объявления” 19](#_Toc138898246)

[3.2 Backend сайта 20](#_Toc138898247)

[3.2.1 База данных 20](#_Toc138898248)

[3.2.2 Навигация 20](#_Toc138898249)

[3.2.3 Регистрация и вход 20](#_Toc138898250)

[3.2.4 Размещение объявления 21](#_Toc138898251)

[3.2.5 Алгоритм фильтрования объявлений по тегам 21](#_Toc138898252)

[Глава 4. Реализация сервиса на базе разработанной модели 22](#_Toc138898253)

[4.1 Реализация frontend 22](#_Toc138898254)

[4.1.1. Верхняя информационная панель страницы 22](#_Toc138898255)

[4.1.2 Нижняя информационная панель страницы 23](#_Toc138898256)

[4.1.3 Главная страница 24](#_Toc138898257)

[4.1.4 Страница “Информация” 25](#_Toc138898258)

[4.1.5. Страница регистрации аккаунта 26](#_Toc138898259)

[4.1.6. Страница авторизации 27](#_Toc138898260)

[4.1.7. Страница размещения объявления 28](#_Toc138898261)

[4.1.8. Страница “Личный кабинет 29](#_Toc138898262)

[4.2 Реализация backend 29](#_Toc138898263)

[4.2.1 Реализация базы данных 30](#_Toc138898264)

[4.2.2 Реализация алгоритма фильтрования 30](#_Toc138898265)

[4.2.3 Реализация алгоритма регистрации пользователя 31](#_Toc138898266)

[4.2.4 Реализация алгоритма входа в аккаунт 31](#_Toc138898267)

[4.2.5 Реализация алгоритма подачи объявления 32](#_Toc138898268)

[Глава 5. Тестирование сервиса 33](#_Toc138898269)

[Заключение 37](#_Toc138898270)

# **Список сокращений и условных обозначений**

ВУЗ Высшее учебное заведение.

ИТ Информационные технологии.

ПО Программное обеспечение.

JS JavaScript

IDE Integrated Development Environment

Header Верхняя часть сайта, расположенная выше блока с основным контентом и отображаемая на всех страницах.

**Footer** Блок в нижней части страницы. Содержит полезную, но не первостепенную информацию. Виден на всех страницах сайта.

# **Введение**

Веб-разработка – это одно из самых быстроразвивающихся и перспективных направлений разработки. Каждый день создается сотни сайтов и загружаются в мировую сеть. На данный момент существует огромное количество веб-сервисов, которые охватывают все сферы жизни человека. Каждый пользователь может найти себе помощника в лице какого-либо сайта для любых нужд.

Тема дипломной работы действительно очень актуальна для студентов ВУЗа. Мой ресурс призван повысить степень взаимодействия студентов между собой и помочь определиться с направлением работы в будущем.

Учащиеся начальных курсов имеют не много навыков для того, чтобы создать проект от начала и до конца, реализовав все сферы разработки. Благодаря моему веб-ресурсу, студент может найти себе партнера для создания проекта, указав навыки, необходимые для работы. Не редко бывает так, что программист не разбирается в дизайне, сочетаемости цветов и т.д. Всегда есть возможность выложить заявку на сайт с требованиями для дизайнера, который может обучаться на другом факультете. Студенты могут работать в тандеме, реализовав модули проекта по своим направлениям подготовки. Это поможет повысить качество проектов. Каждый студент сможет презентовать работу по своему направлению и получить отметку от преподавателя.

Также, разработка не будет бесполезна для старших курсов бакалавриата, или магистратуры. Full Stack разработка – это очень сложный процесс, в данный период обучения многие программисты уже выбирают узкую специализацию, в которой развиваются и планируют работать в дальнейшем. Для такого типа проектов, необходим долгий срок обучения и большое количество времени. Для разработчика, уже определившегося с вектором технологий, в которых он разбирается, полный цикл будет очень трудоемким, в какой-то степени бесполезным и не пойдет на пользу качеству проекта. Если в проекте поучаствует несколько специалистов, которые будут работать по своим направлениям, качество возрастет.

Данная разработка поможет начинающему специалисту научиться работать в команде и приблизить процесс работы над проектом к реалиям работы в настоящей компании, где присутствует разграничение по обязанностям.

**Глава 1.  
Анализ аналогов и актуальности темы**

**1.1 Существующие аналоги**

Сайтов для фриланса огромное множество. Большое количество команд и компаний создавали свои вариации сервисов. Однако, это совсем не значит, что тема полностью разработана. Есть множество возможностей для улучшения по данной теме.

Хотелось бы провести обзор наиболее популярных сервисов, выделить их плюсы и минусы. Проанализировать технологии, которые можно применить в моем проекте.



Рисунок 1.1 Логотип Upwork

Upwork (до мая 2015 — Elance-oDesk) – компания, предлагающая глобальную площадку по поиску работы и ряд программных продуктов для работодателей, которые хотят нанять и сотрудничать с удаленными специалистами. oDesk основали в 2003 году.

Upwork позволяет клиентам создавать онлайн-команды разработчиков. Координация и оплата происходит с помощью программного обеспечения компании и сайта. Потенциальные клиенты могут размещать проекты бесплатно, а фрилансеры могут создавать профили и делать ставки на проекты, также бесплатно. Компания собирала 10 % (с 2016 года — от 5 % до 20 %) от оплаты за работу от работодателя фрилансеру. В дополнение к площадке проектов, услуг платежей и бухгалтерии, компания предлагает программное обеспечение, которое позволяет клиентам видеть процесс работы фрилансера в то время, как он находится в режиме оплачиваемого времени.

7 марта 2022 года Upwork приостановил работу с фрилансерами и клиентами из России и Белоруссии, а 1 июня заморозил аккаунты Upwork.

Это отличный пример сервиса для фриланса, который собрал в себе большое количество интересных решений, но на данный момент сайт недоступен в России и Белоруссии.



Рисунок 1.2 Логотип FL.ru

FL.ru (Free-lance.ru) — сервис для поиска [удалённой работы](https://ru.wikipedia.org/wiki/%D0%A3%D0%B4%D0%B0%D0%BB%D1%91%D0%BD%D0%BD%D0%B0%D1%8F_%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0) и сотрудников в России и СНГ. Портал был запущен в мае 2005 года и по состоянию на 2012 год являлся крупнейшей [биржей](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D1%80%D0%B6%D0%B0) в России и странах СНГ. В конце 2004 года состоялся пилотный запуск Free-lance.ru. Вскоре после старта разработчики закрыли сайт для его полной переработки. Обновленная версия запустилась 14 мая 2005 года и к концу года обзавелась платными услугами. В апреле 2013 года был анонсирован короткий домен биржи — FL.ru.

Данная биржа имеет ряд плюсов, некоторые из которых стоит взять на заметку в своем проекте.

1. Большое количество заказчиков.
2. Большое количество направлений работы.
3. База знаний, которая поможет начинающему пользователю.

Также, необходимо рассказать о минусах сайта, так как они существуют и являются довольно весомыми:

1. Дизайн сайта устроен таким образом, что новичку в нем сложно разобраться, он излишне усложнен.
2. Есть ощущение, что на сайте давно не проводились работы по модернизации и улучшению внешнего вида.
3. Присутствует сложная система PRO-аккаунтов, которые очень сильно увеличивают разницу между их владельцами и теми, кто не обладает такой привилегией.



Рисунок 1.3 Логотип [Workzilla](https://work-zilla.com)

[Workzilla](https://work-zilla.com) — биржа фриланса для исполнителей и заказчиков по всему миру. Здесь большой выбор работы самого разного профиля: от небольших задач для начинающих, где не требуются глубокие знания и навыки, до проектов для профессионалов. Чтобы брать заказы, необходимо зарегистрироваться и пройти тест.

Из плюсов стоит ответить:

1. Приятный дизайн.
2. Хорошая техническая поддержка.
3. Наличие несложных заданий для начинающих специалистов.

Из минусов стоит отметить:

1. Комиссия в некоторых случаях достигает 15% для пользователя.
2. Отзывы об исполнителе видны 6 месяцев, после чего пропадают.

Исходя из анализа различных сервисов, можно понять, что данная ниша недостаточно развита в России. Во всех сайтах присутствуют плюсы и минусы, которые можно учесть при разработке собственного проекта. Мной не были найдены сервисы, которые будут направлены исключительно на студента – это в очередной раз доказывает уникальность проекта.

**1.2 Цель и задачи**

Целью работы является создание сайта, который облегчит поиск партнеров для любого проекта.

Для достижения поставленной цели необходимо решить ряд задач, в соответствии с которыми построена структура работы.

1. Анализ аппаратных и программных средств для реализации проекта.
2. Анализ подобных проектов.
3. Формирование технического задания.
4. Проектирование и разработка базы данных.
5. Разработка сайта.
6. Тестирование всех частей проекта.
7. Документирование процесса разработки и результатов.
8. Формирование отчета и презентации о проделанной работы.

Если данный перечень задач будет корректно выполнен, в итоге получится работающий сервис, который сможет выполнять задуманные функции.

**1.3 Научная новизна**

Данная тема не нова, но ранее все подобные ресурсы были рассчитаны на профессиональную разработку коммерческих проектов. Такая концепция не совсем подходит для реалий ВУЗа. Мой сайт имеет ряд особенностей, которыми не смогут похвастаться подобные сервисы.

1. Главная цель разработок в университете — это знания и опыт, а не материальные ценности. На сайте есть несколько вариантов оплаты: деньги, какие-либо вещи, или отсутствие оплаты.
2. Так как сервис запускается в рамках одного ВУЗа, студенты всегда могут встретиться друг с другом для передачи вознаграждения.
3. Разнообразие работ. В нашем вузе огромное множество различных направлений подготовки. Каждый найдет для себя проект по душе.
4. Сервис работает на безвозмездной основе. Для размещения и поиска объявлений не взымается плата. Пользователь может лишь поддержать проект с помощью пожертвования.
5. Высокий уровень качества. Благодаря простоте сайта, он работает быстро и стабильно. Это повышает удобство пользования сервисом.
6. Сайт отличается от других сервисов тем, что пользователь с помощью сервиса ищет коллег, которые помогут ему с разработкой проекта, а не передает проект целиком.

Работы над улучшением сервиса будут вестись и далее, чтобы обеспечить максимальное удобство использования веб-сервиса. Таким образом, можно сделать вывод, что данная тема является довольно развитой, но остается еще большое множество пространства для креатива и чего-то нового.

**1.4 Теоретическая и практическая значимость работы**

Теоретическая значимость работы заключается в том, что она представляет собой исследование сферы фриланса и интеграцию данной концепции в реалии обучения в ВУЗе. Также, значимым является создание сайта для этой области и его корректная настройка. Работа содержит теоретический анализ, сравнительный анализ и обзор существующих сайтов фриланса, описание особенностей работы и процесса создания сайта. Также, значимым является опыт, который был получен при проектировании, создании, отладке и документировании проекта. Это отличная возможность начать развитие в сфере web-разработки.

Практическая значимость заключается в создании ресурса, который позволит студентам КФУ из разных направлений подготовки работать вместе над проектами. Такой сайт может помочь учащимся, которые хотят получить опыт работы в команде, создать портфолио для успешного прохождения собеседований на работу, повысить качество своих проектов, находить новые возможности для роста. Разработка в различных сферах позволит расширить кругозор студента, поможет раскрыть таланты, будет мотивировать учащегося узнавать что-то новое, стремиться к большему.

**1.5 Методы исследования**

Исследование темы включает в себя следующие методы:

1. Анализ литературы – изучение различных источников, как печатных, так и электронных, по данной теме. Включает в себя обзоры сайтов и отзывов пользователей, а также анализ статей и книг по web-разработке.
2. Сравнительный анализ включает в себя сравнение различных сайтов фриланса на основе определенных критериев.
3. Анализ технологий поможет разобраться, какие из них необходимы и какую пользу они смогут принести проекту. Включает в себя прочтение статей и документации по данным технологиям.

Каждый из методов поможет улучшить и разнообразить сайт. Сервис лишится некоторых недостатков заранее, благодаря методам исследования, представленных выше.

**1.6 Целевая аудитория**

Целевой аудиторией сервиса являются первую очередь учащиеся начальных курсов бакалавриата КФУ. Данный сайт поможет им быстрее начать разработку, повысить качество разрабатываемых проектов, найти надежных друзей и возможно получить дополнительные балы от преподавателей, если разработка пересекается с темами курсов. Студенты смогут пополнять свои портфолио с ранних курсов, что повысит их опыт и поможет в будущем с меньшими усилиями устроиться на работу.

**Глава 2.  
Инструментальные средства разработки**

**2.1 Описание проекта**

Мой проект создан для студентов КФУ и призван облегчить процесс создания проектов в первую очередь для студентов начальных курсов. Проект представляет собой сайт с объявлениями. Каждый пользователь может выложить объявление, где содержится название проекта, описание, вознаграждение, а также требования к знаниям и навыкам претендента в форме тегов. Пользователь сможет написать о своих навыках и опыте в отдельном окне в профиле. Необходимо реализовать алгоритм фильтрования заявок и резюме. Для каждого пользователя будут показаны только те заявки, которые подходят ему, опираясь на теги в его профиле, которые он выбрал при регистрации.

В качестве вспомогательного элемента используется Telegram, данный сервис выступает в качестве чата для сайта. Пользователи смогут обсуждать проект именно там.

**2.2 Обзор технологий**

Для создания такого проекта, как сайт необходимо немалое количество технологий. Обзор технологий поможет структурировать информацию о них, а также, разобраться, какие из них действительно нужны.

**2.2.1 Среда разработки**

Visual Studio Code (VS Code) – это бесплатный редактор кода с обширным функционалом. Он доступен для многих операционных систем и поддерживает большинство популярных языков программирования, благодаря чему является универсальным инструментом, который подойдет для разработки frontend и backend частей.

Основные преимущества VS Code включают:

* Быстрый редактор с подсветкой синтаксиса.
* Поддержка Git.
* Расширяемость с помощью большого количества доступных расширений и плагинов.

**2.2.2 Frontend**

HTML (Hypertext Markup Language) — это язык разметки гипертекстовых документов.[1] Он нужен, чтобы отображать в браузере специальным образом отформатированный документ с множеством вложенных элементов: заголовками, абзацами, списками, гиперссылками, медиа источниками, расположением изображений, видео и аудио. HTML – это основа любого веб-сервиса. HTML является фундаментом, который содержит структуру страницы, блоки других языков.

CSS (Cascading Style Sheets) — это язык таблиц стилей, используемый для настройки внешнего вида веб-страницы, написанной на HTML.[2]

CSS определяет различные аспекты стиля веб-страницы, такие как цвет, шрифты, отступы, размеры и позиционирование элементов, а также анимацию и эффекты перехода. Также данный язык имеет специальные селекторы, которые позволяют настраивать стили для конкретных элементов, например, селекторы по атрибуту, по классу или по идентификатору.

JavaScript (JS) — это высокоуровневый язык программирования, который широко используется для создания динамических интерактивных веб-сайтов и приложений. [3] Он работает в браузере и может использоваться для добавления динамического поведения на веб-страницах, а также для создания серверных приложений и мобильных приложений.

JavaScript часто используется с другими языками, такими как HTML и CSS, и может использоваться для создания интерактивных пользовательских интерфейсов, динамических функций, управления поведением и многого другого.

**2.2.3 Backend**

Node.js – это серверная платформа, основанная на движке V8 JavaScript от Google. Она позволяет использовать JavaScript для создания быстрого и масштабируемого веб-приложения на стороне сервера. [5,7] Node.js позволяет разработчикам использовать JavaScript не только на клиентской стороне, но и на серверной, что делает его уникальным инструментом для full stack веб-приложений на JS. Node.js основывается на механизме событийной модели, что позволяет иметь высокую производительность и возможность взаимодействия с большим количеством клиентов одновременно.

MySQL — это реляционная система управления базами данных (СУБД), которая распространяется как свободное программное обеспечение.[4] Является одной из наиболее популярных, так как отличается гибкостью, легкостью, удобством в использовании. Также, присутствует большое количество плагинов, которые расширяют функционал БД. В данном случае, необходим, чтобы хранить информацию о пользователях и заказах.

JSON (JavaScript Object Notation) используется для передачи данных. JSON является форматом данных, основанным на тексте, который может быть легко прочитан и записан как человеком, так и компьютером.[6]

Использование JSON на сайте позволит обмениваться данными между клиентским и серверным приложениями. JSON-объекты используются для передачи структурированных данных между сервером и клиентом.

**Глава 3.  
Разработка концептуальной модели сервиса**

**3.1 Frontend сайта**

Frontend – это часть веб-разработки, которая отвечает за создание пользовательского интерфейса на сайте. Для данной части, были выбраны такие технологии, как: HTML, CSS, JavaScript. Целью frontend разработки является создание удобного и привлекательного пользовательского интерфейса, который был бы понятен пользователю с любым уровнем владения ПК.

**3.1.1 Верхняя информационная панель страницы**

Необходимо разработать удобную верхнюю часть страницы (header), которая будет обеспечивать комфортную навигацию по сайту, в данной части страницы будут размещены: логотип, название сайта, кнопки перемещения между главной страницей, личным кабинетом, вкладкой “Информация” и вкладкой “Разместить объявление”. Также, в шапке должны содержаться кнопки для регистрации и входа в аккаунт. Данный элемент будет содержаться в каждой странице на сайте.



Рисунок 3.1. Схема верхней информационной панели (header).

**3.1.2 Нижняя информационная панель страницы**

Необходимо разработать удобную нижнюю часть страницы (footer), в которой будут продублированы все навигационные кнопки на сайте, контакты, также будет указано название компании, разработавшей сайт. Данный элемент будет содержаться в каждой странице на сайте.

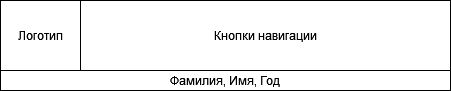


Рисунок 3.2. Схема нижней информационной панели (footer).

**3.1.3 Главная страница**

На главной странице содержатся карточки с заказами, которые подходят пользователю на основании его навыков, введенные при регистрации в виде тегов.

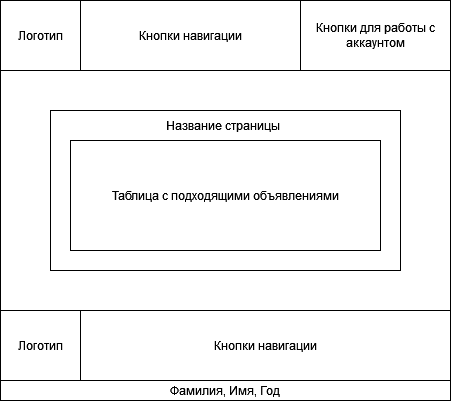


Рисунок 3.3. Схема главной страницы.

**3.1.4 Страница “Информация”**

Страница “Информация” должна содержать информацию о сайте, где будет описано, что данный проект создан автором в рамках ВКР.

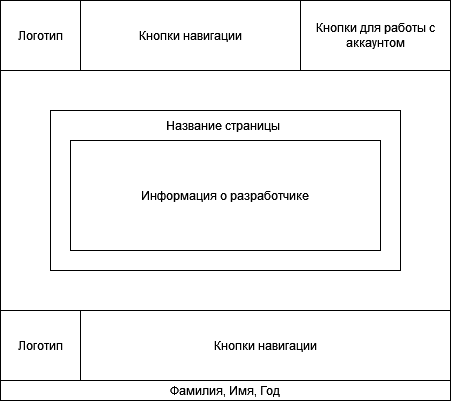


Рисунок 3.4. Схема страницы “Информация”.

**3.1.5 Страница регистрации**

Страница регистрации, должна содержать поля для ввода имени, фамилии, адреса Telegram, пароля. Также, на данной странице должен содержаться элемент для ввода навыков пользователя.

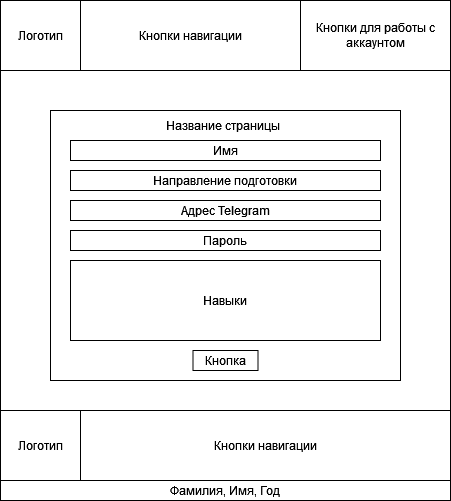


Рисунок 3.5. Схема страницы регистрации.

**3.1.6 Страница входа**

Страница входа должна содержать поля для ввода Telegram-аккаунта и пароля.

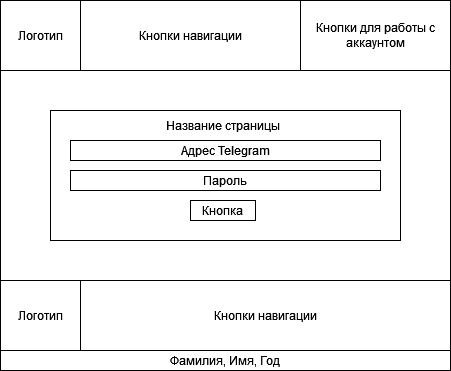


Рисунок 3.6. Схема страницы авторизации.

**3.1.7 Страница “Размещение объявления”**

Данная страница нужна для того, чтобы разместить новое объявление и содержит в себе поля для ввода названия проекта, описания проекта, сроков (если они существуют), вознаграждения, а также для необходимых навыков.



Рисунок 3.7. Схема страницы регистрации объявлении.

**3.2 Backend сайта**

**3.2.1 База данных**

Необходимо спроектировать и разработать базу данных, в таблицах которой будет содержаться информация о пользователях и заказах.

* Первая таблица будет содержать в себе информацию о пользователе: имя, курс, адрес telegram, пароль.
* Вторая таблица должна содержать имя пользователя и навыки, которые были введены при регистрации.
* Третья таблица будет содержать информацию о заказах: автор, название, описание, необходимые навыки.

**3.2.2 Навигация**

Необходимо настроить навигацию на сайте, чтобы все кнопки вели на необходимые страницы.

**3.2.3 Регистрация и вход**

При регистрации пользователя должно происходить сравнение введенных данных с уже имеющимися, чтобы исключить повторную регистрацию, после чего необходимо создать новую строку в базе данных с информацией, которая была введена.

При входе в аккаунт необходимо проверять БД с уже зарегистрированными пользователями на совпадение с введенными данными.

**3.2.4 Размещение объявления**

При создании объявления, должна создаваться новая строка в соответствующей таблице базы данных.

**3.2.5 Алгоритм фильтрования объявлений по тегам**

Необходимо сравнивать теги заказа с тегами пользователя и на основе совпадений, предоставлять карточки с заказами на основе сравнения.

**Глава 4.  
Реализация сервиса на базе разработанной модели**

**4.1 Реализация frontend**

Для frontend разработки были выбраны такие языки, как HTML, CSS и JavaScript. Для более простого представления информации, программный код будет разбит на фрагменты и представлен раздельно с комментариями.

**4.1.1.** **Верхняя информационная панель страницы**

На рисунке представлен внешний вид верхней информационной панели

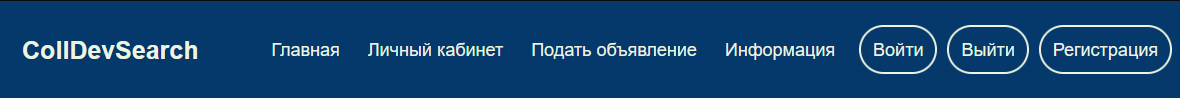


Рисунок 4.1. Внешний вид верхней информационной панели (header).

Программная реализация представленной панели построена на базе стандартных элементов языка разметки html5.



Рисунок 4.2. Программный код верхней информационной панели (header).

**4.1.2 Нижняя информационная панель страницы**

На рисунке представлен внешний вид нижней информационной панели.

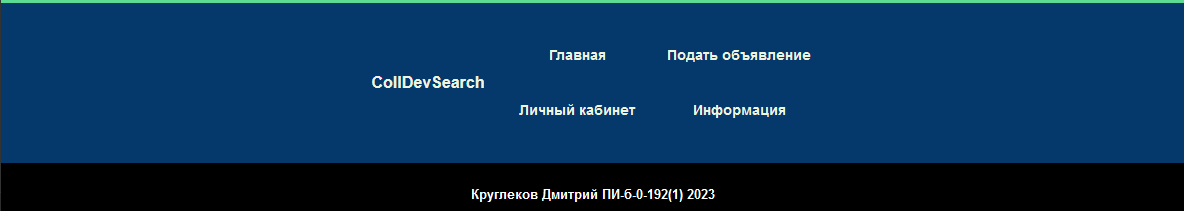


Рисунок 4.3. Внешний вид нижней информационной панели (footer).

Программная реализация представленной панели также построена на базе стандартных элементов языка разметки html5.



Рисунок 4.4. Программный код нижней информационной панели (footer).

**4.1.3 Главная страница**

На главной странице представлены все элементы необходимые пользователю для быстрого доступа к основному функционалу сервиса.



Рисунок 4.5. Внешний вид главной страницы.



Рисунок 4.6. Программный код главной страницы.

**4.1.4 Страница “Информация”**

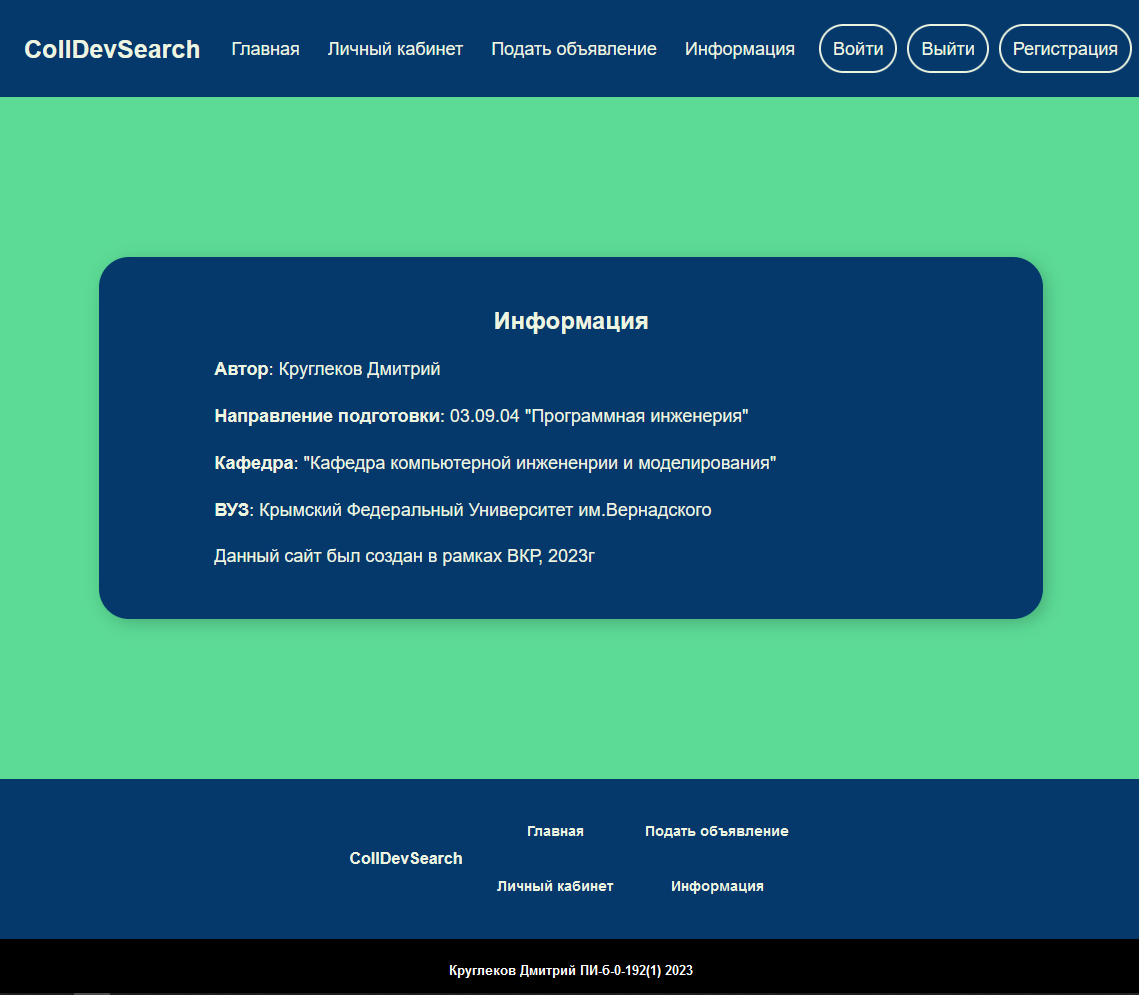


Рисунок 4.7. Внешний вид страницы “Информация”.

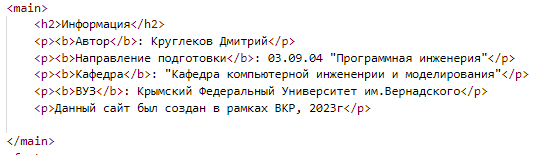


Рисунок 4.8. Программный код страницы “Информация”.

**4.1.5. Страница регистрации аккаунта**

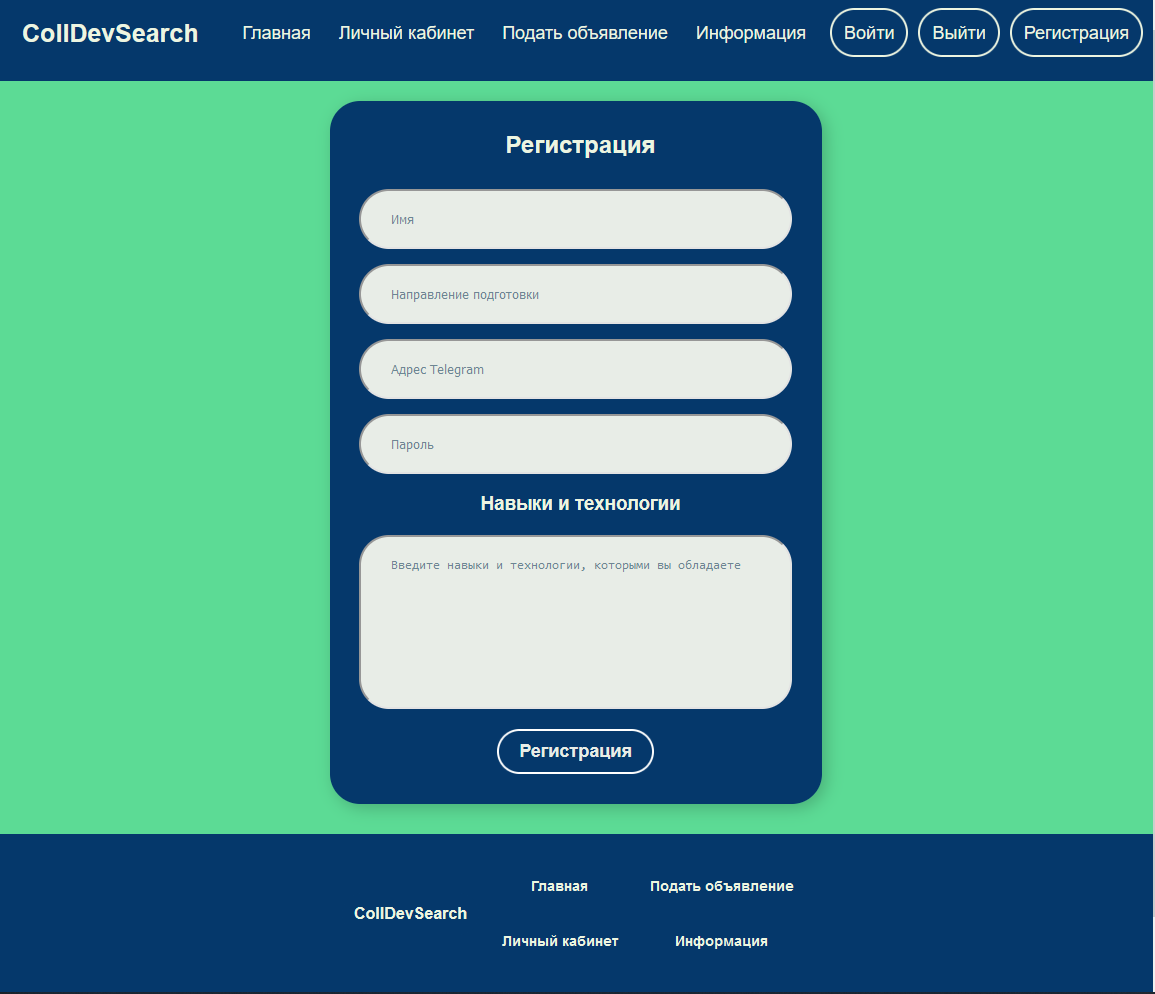


Рисунок 4.9. Внешний вид страницы регистрации.

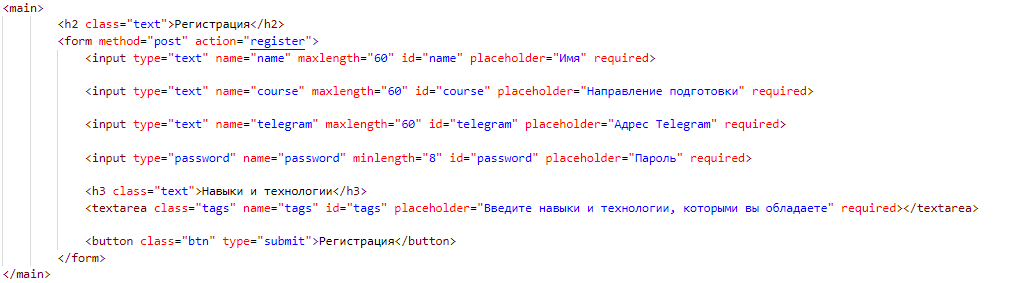


Рисунок 4.10. Программный код страницы регистрации.

**4.1.6. Страница авторизации**

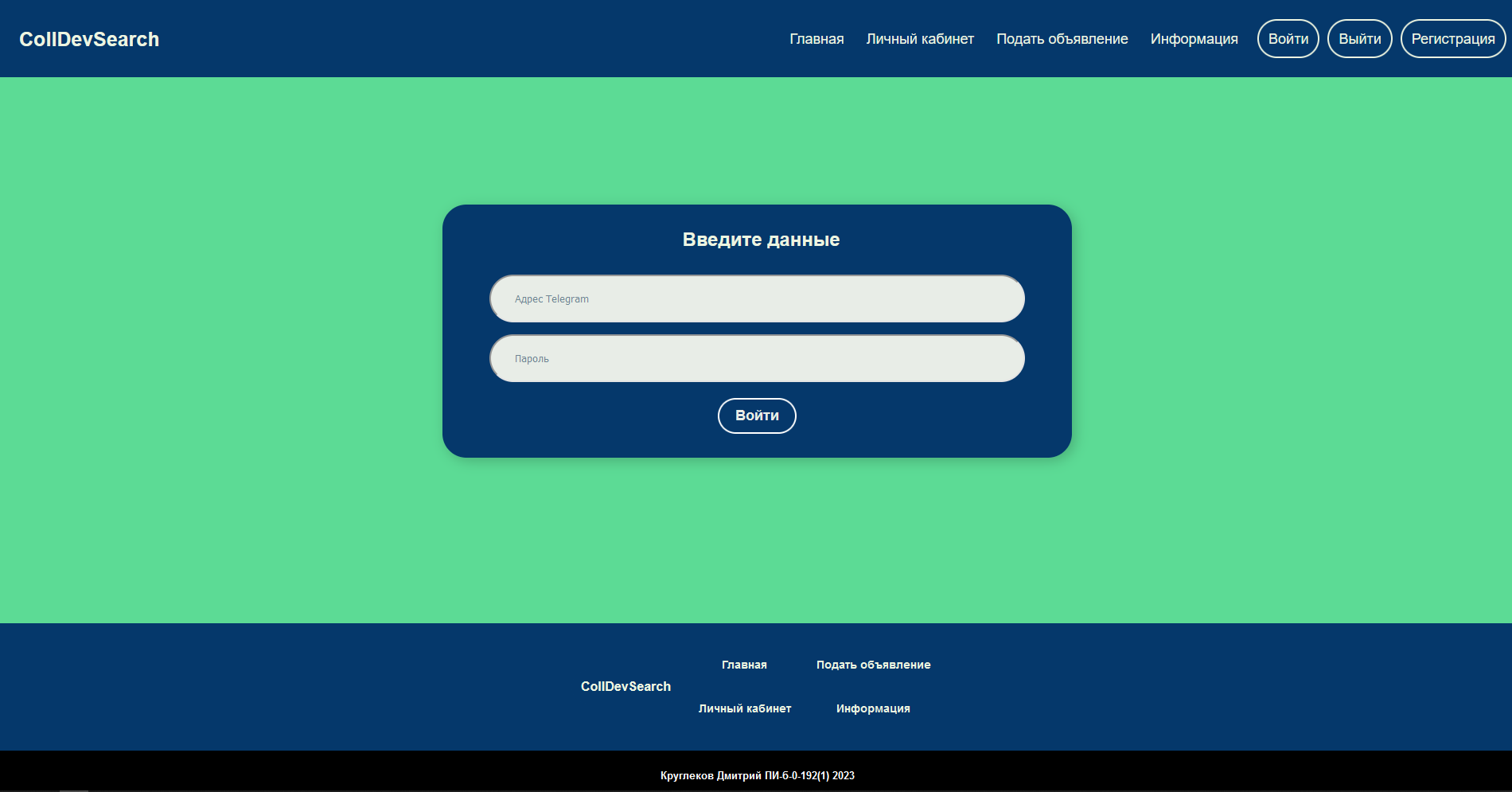


Рисунок 4.11. Внешний вид страницы авторизации.

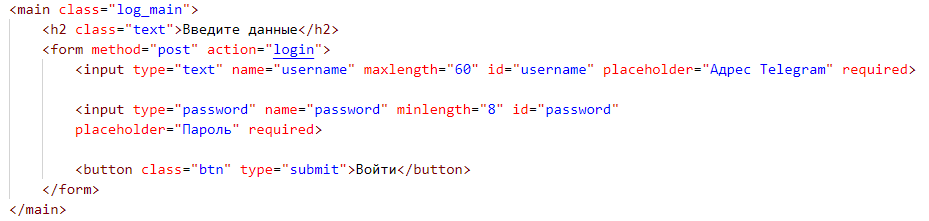


Рисунок 4.12. Программный код страницы авторизации.

**4.1.7. Страница размещения объявления**

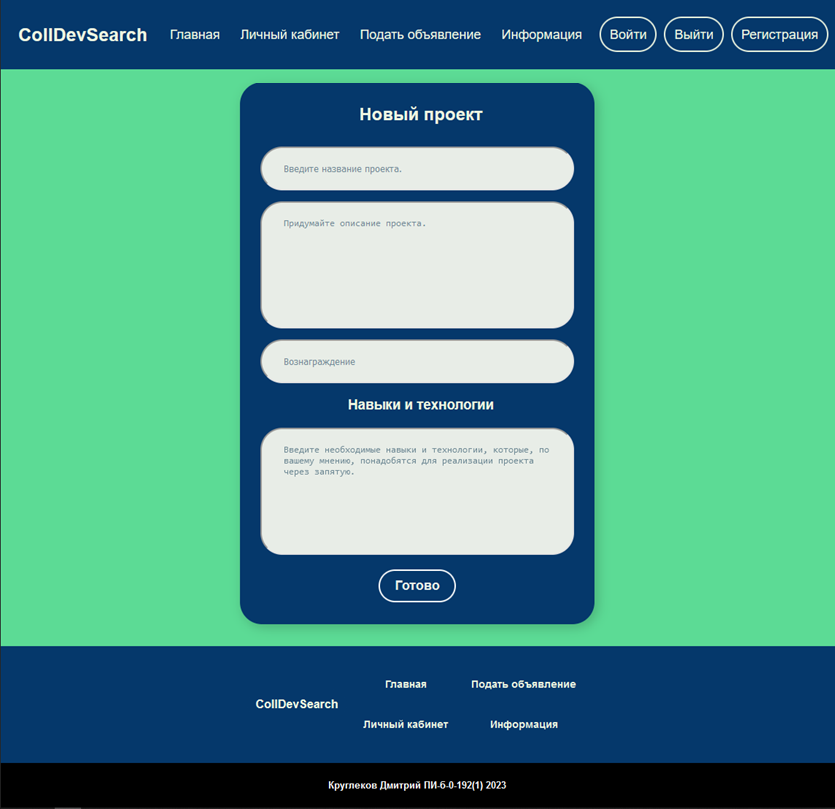
****

Рисунок 4.13. Внешний вид страницы подачи объявления.

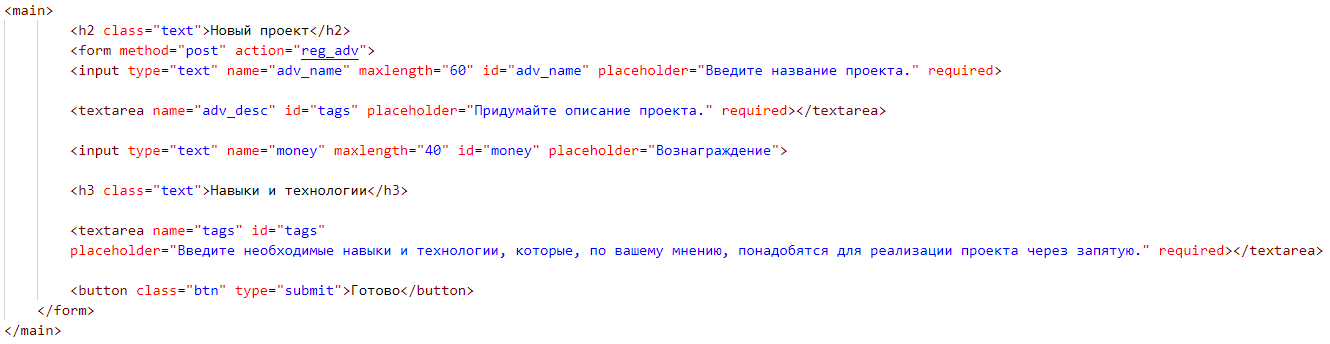


Рисунок 4.14. Программный код страницы подачи объявления.

**4.1.8. Страница “Личный кабинет**

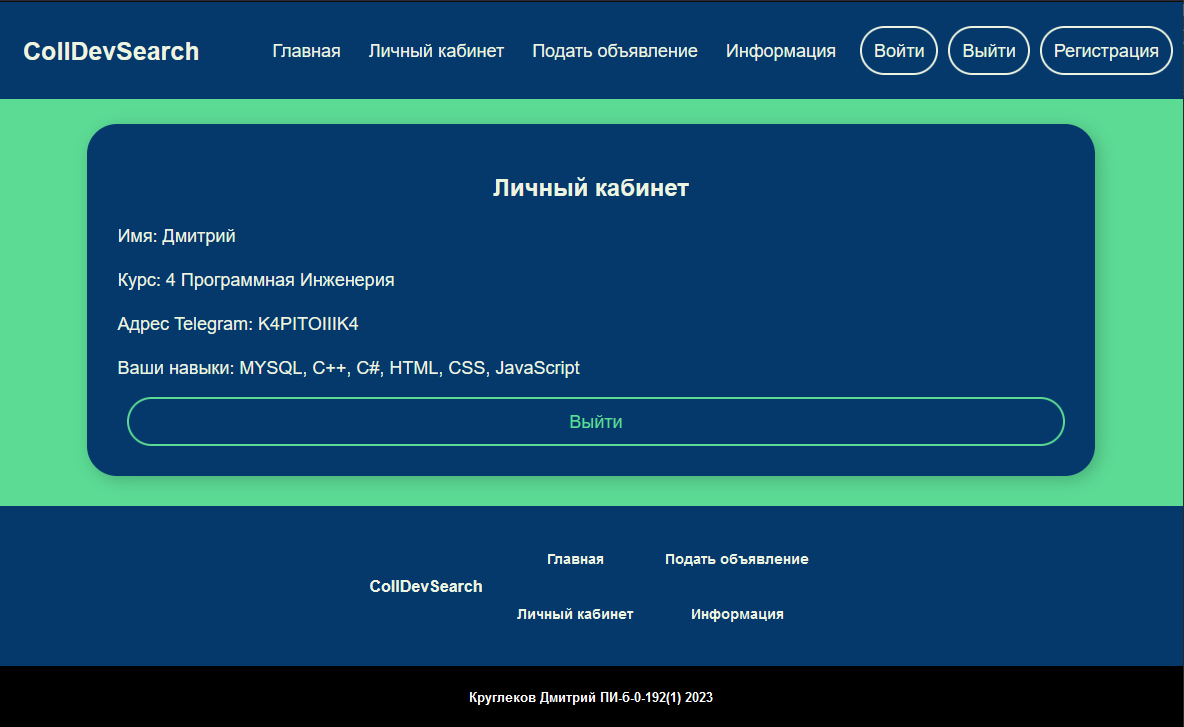


Рисунок 4.15. Внешний вид страницы “Личный кабинет”.

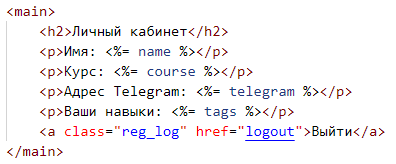


Рисунок 4.16. Программный код страницы “Личный кабинет”.

**4.2 Реализация backend**

Для backend разработки были выбраны такие языки, как JavaScript с библиотекой Node, а также JSON. Для построения базы данных был выбран MySQL. Для более простого представления информации, программный код будет разбит на фрагменты и представлен раздельно с комментариями.

**4.2.1 Реализация базы данных**



Рисунок 4.17. Список таблиц, задействованных в работе.

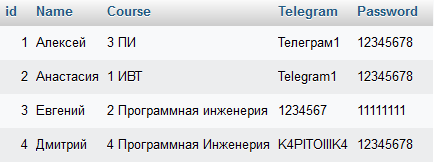




Рисунок 4.18. Таблица “users” Рисунок 4.19. Таблица “tags”



Рисунок 4.20. Таблица “job”

**4.2.2 Реализация алгоритма фильтрования**

Рисунок 4.21. Программный код алгоритма фильтрования объявлений.

**4.2.3 Реализация алгоритма регистрации пользователя**



Рисунок 4.22. Программный код алгоритма регистрации пользователя.

**4.2.4 Реализация алгоритма входа в аккаунт**

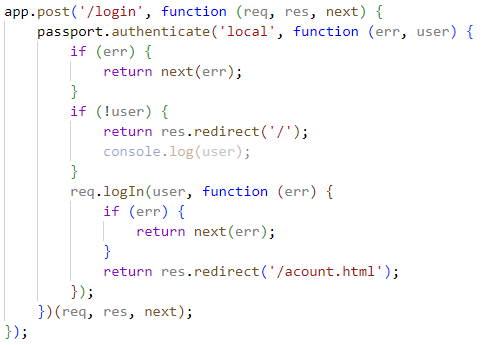


Рисунок 4.23. Программный код алгоритма авторизации.

**4.2.5 Реализация алгоритма подачи объявления**



Рисунок 4.24. Программный код алгоритма подачи объявления.

**Глава 5.  
Тестирование сервиса**

Тестирование всех компонентов сервиса проводились вручную с использованием тест кейсов. Ручной тип тестирования выбран в качестве наиболее приближенного к реальному использованию сервиса. Заранее были подготовлены тест кейсы, которые позволят проверить весь задуманный функционал.

Тест кейсы были представлены в виде таблицы, где в первом столбце сформированы действия, которые необходимо выполнить, во втором столбце ожидаемая реакция системы на данное действие, в третьем столбце тестировщик должен написать реальную реакцию системы на действие, в 4 столбце он должен дать вердикт, прошел ли тест данный модуль.

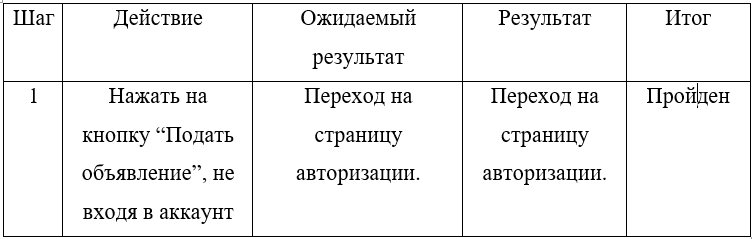


Рисунок 5.1. Тест-кейс №1 тестирование алгоритма проверки авторизации.

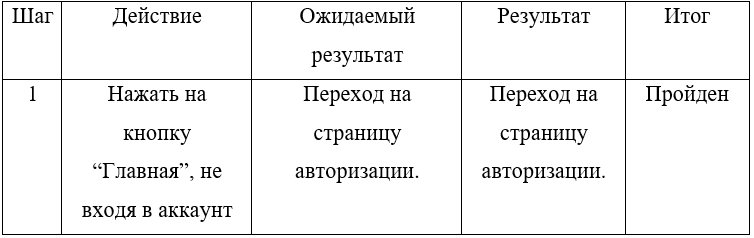


Рисунок 5.2. Тест-кейс №2 тестирование алгоритма проверки авторизации.



Рисунок 5.3. Тест-кейс №3 тестирование алгоритма регистрации.

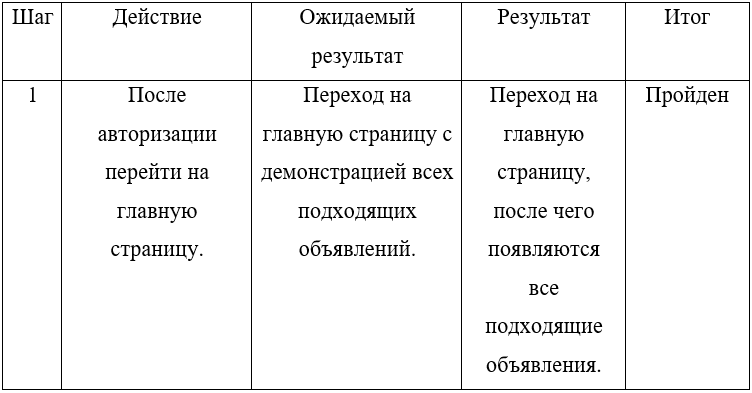


Рисунок 5.4. Тест-кейс №4 тестирование алгоритма вывода данных таблицы.

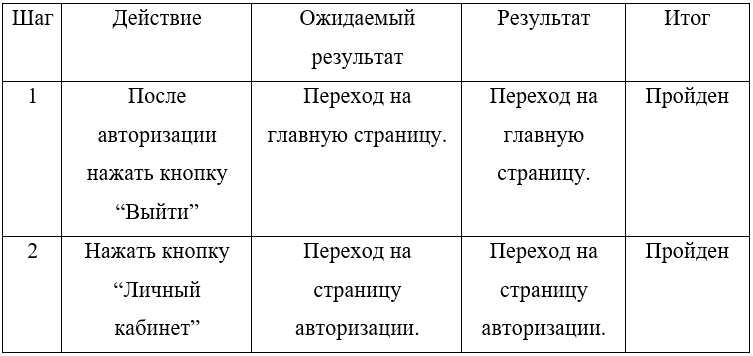


Рисунок 5.5. Тест-кейс №5 тестирование алгоритма выхода из аккаунта.

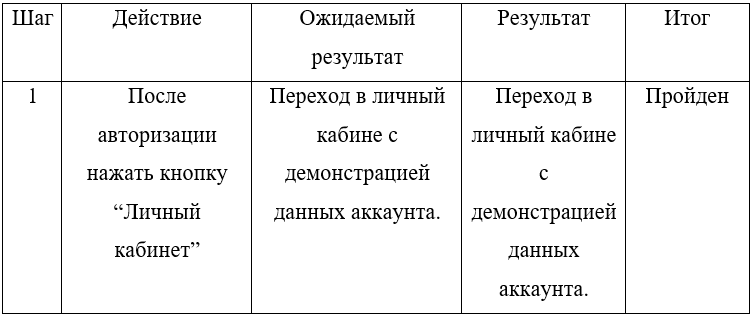


Рисунок 5.6. Тест-кейс №6 тестирование алгоритма вывода данных из таблицы.

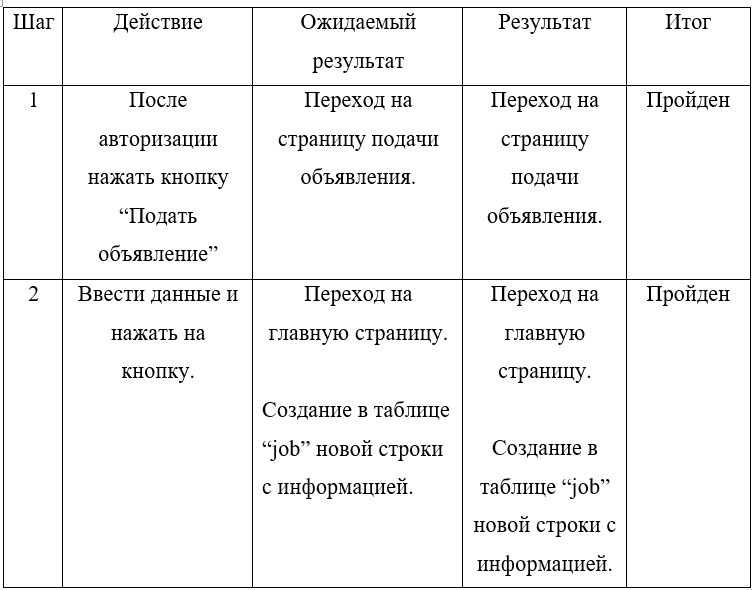


Рисунок 5.7. Тест-кейс №7 тестирование алгоритма подачи объявления.

# **Заключение**

После тщательного анализа технологий, аналогичных сервисов, опроса пользователей и долгого и кропотливого процесса разработки, можно сделать вывод, что был создан сервис с удобным интерфейсом, поддерживающий весь необходимый функционал, который поможет студентам найти себе партнера для реализации самых сложных проектов. Данная система имеет неоспоримые плюсы:

* Благодаря тому, что каждый будет выполнять определенную роль в разработке, у учащегося появится возможность заранее решить, в каком направлении ИТ он желает развиваться. Такой тип разработки приблизит студента к реалиям настоящей работы в компании.
* Благодаря множеству предложений, учащийся сможет дополнительно узнавать что-то новое в различных сферах, которые ему интересны.
* Совместная разработка – это возможность найти надежных партнёров, с которыми в будущем можно продолжить работу.

Создание сайта – это возможность развития в web разработке полного стека. Было изучено много статей и литературы, исправлено множество ошибок, некоторые навыки были восстановлены, некоторые получены с нуля. Мной было изучено множество технологий, которые несомненно пригодятся в будущем. Данный проект отлично подойдет для пополнения портфолио.

Проект имеет перспективы для развития, разработка продолжится, что позволит сделать сервис более стабильным и удобным для пользователя. Существует большое количество идей с новыми функциями, которые необходимо реализовать. Благодаря данным функциям, сервис станет более интересным и разнообразным, что позволит привлечь больше пользователей.

В заключении, с уверенностью можно сказать, что в процессе написания ВКР были выполнены все поставленные задачи, благодаря чему, можно с уверенностью сказать, что цель дипломной работы была достигнута.

**Список используемых источников**

1. HTML // SkillFactory URL: blog.skillfactory.ru
2. CSS // SkillFactory URL: blog.skillfactory.ru
3. JavaScript // SkillFactory URL: blog.skillfactory.ru
4. MySQL // SkillFactory URL: blog.skillfactory.ru
5. Node // SkillFactory URL: blog.skillfactory.ru
6. JSON // SkillFactory URL: blog.skillfactory.ru
7. 15 библиотек JavaScript, о которых стоит знать // Tech URL: techrocks.ru
8. Холланд Минник JavaScript для чайников
9. Поль Дюбуа MySQL. Сборник рецептов. - 3-е издание
10. The State of Node.js & JavaScript for Backend Development // Snipcart URL: snipcart.com
11. IDE // SkillFactory URL: blog.skillfactory.ru
12. Хедер // SkillFactory URL: blog.skillfactory.ru