Министерство образования Республики Беларусь

Учреждение образования

БелорусскиЙ государственный университет

информатики и радиоэлектроники

Факультет информационных технологий и управления

Кафедра информационных технологий автоматизированных систем

Дисциплина: Интернет технологии и веб-программирование

|  |
| --- |
|  |
|  |
|  |

Пояснительная записка

к курсовому проекту

на тему

**онлайн-платформа для поиска работы и размещения вакансий**

БГУИР КП 1-53 01 02 01 002 ПЗ

|  |  |  |
| --- | --- | --- |
| Студент группы 120601 |  | В. А. Антоненко |
| Руководитель |  | Н. В. Хаджинова |

Минск 2024

Учреждение образования

«Белорусский государственный университет информатики   
и радиоэлектроники»

Факультет информационных технологий и управления

УТВЕРЖДАЮ

Заведующий кафедрой

*––––––––––––––––––––––––*

(подпись)

–––––––––––––––2024г.

ЗАДАНИЕ

по курсовому проектированию

Студенту    *Янущику Григорию Сергеевичу––––––––––––––––––*

1. Тема проекта   Объектно-ориентированный анализ и проектирование программного обеспечения. Система книжного магазина.

2. Срок сдачи студентом законченного проекта– ––––*20 мая 2023 г.–––*

3. Исходные данные к проекту *Книжный магазин предлагает возможность приобретения своих товаров через Интернет. Клиент может выбрать книги на web-странице магазина. Книги разделяются по жанрам, авторам. Также нужную книгу можно найти по ее названию. Книги, которые клиент желает приобрести, могут быть добавлены в корзину. В корзине можно сформировать заказ и добавить в него те товары, которые нужно. При формировании заказа клиент выбирает вариант доставки, дату доставки и способ оплаты. По окончании работы с заказом, он помечается в системе как выполненный. Заказы хранятся в системе в течение 15 месяцев с момента создания для составления годовых и квартальных отчетов, после чего автоматически удаляются.*

*Также существует возможность зарегистрироваться в системе. Зарегистрированные пользователи имеют дополнительные функции, такие как: добавить товар в избранное, оставить комментарий к товару, отменить заказ, отредактировать профиль. Неавторизованные пользователи могут отменить заказ только связавшись с менеджером по*

4. Содержание расчетно-пояснительной записки (перечень вопросов, которые подлежат разработке):

*Введение*

1. *Анализ технического задания на курсовое проектирование*
2. *Проектирование системы складского учета*
3. *Программная реализация системы складского учета*

*Заключение*

*Список использованных источников*

*Приложения*

*Ведомость курсового проекта*

5. Перечень графического материала (с точным обозначением обязательных чертежей и графиков)

1. *Диаграмма вариантов использования (ПЛ, формат А1)*
2. *Диаграмма классов (ПЛ, формат А1)*

6. Консультант по проекту (с обозначением разделов проекта)  *Т. С. Боброва–*

7. Дата выдачи задания –––– –*22 февраля 2023 г.––––––––––––––––––––   –*

8. Календарный график работы над проектом на весь период проектирования (с обозначением сроков выполнения и трудоемкости отдельных этапов):

*раздел 1 к 15.03 – 15 %; ––––––––––––––––– ––––––––––––––––        –*

*раздел 2 к 15.04 – 50 %; –– –––––––––––––– ––––––––––––––––––       ––––*

*раздел 3 к 15.05 – 15 %; ––––––––––––––––––––– –––––––––––––––––*

*оформление пояснительной записки и графического   
материала к 20. 05 – 90 %*

*Защита курсового проекта с* ***23.05*** *по* ***05.06.2025****г.––––––––––––––––––––*

Руководитель*– –––––––– В. И. Ярмолик*

(подпись)

Задание принял к исполнению *–––––––\_\_\_\_\_\_\_\_\_\_\_\_ Г. С. Янущик*

(дата и подпись студента)

**СОДЕРЖАНИЕ**

[Введение 5](#_Toc184543119)

[1 Анализ предметной области 6](#_Toc184543120)

[1.1 Описание предметной области 6](#_Toc184543121)

[1.2 Постановка задачи 6](#_Toc184543122)

[2 Проектирование архитектуры системы 8](#_Toc184543123)

[2.1 Концептуальная модель системы 8](#_Toc184543124)

[2.3 Алгоритмическое обеспечение 11](#_Toc184543125)

[3 Реализация веб-приложения 13](#_Toc184543126)

[3.1 Техническое обеспечение 13](#_Toc184543127)

[3.2 Программное обеспечение 15](#_Toc184543128)

[3.3 Руководство пользователя 17](#_Toc184543129)

[Заключение 29](#_Toc184543130)

[Список использованных источников 30](#_Toc184543131)

[Приложение А (обязательное) Программный код функции приложения 31](#_Toc184543132)

ВВЕДЕНИЕ

В современном мире процессы автоматизации играют ключевую роль в развитии всех сфер человеческой деятельности. Прогресс в технологиях, начиная от искусственного интеллекта и машинного обучения до облачных вычислений и больших данных, открывает новые горизонты для оптимизации работы, повышения производительности и улучшения качества жизни. Автоматизация позволяет сократить время на выполнение рутинных и повторяющихся задач, минимизировать ошибки, связанные с человеческим фактором, и сосредоточить усилия на более творческих и аналитических аспектах деятельности.

Бурное развитие информационных технологий привело к их активному использованию в различных областях, включая сферу поиска работы. В последние годы компьютерные технологии существенно изменили процесс поиска работы через интернет. Современные платформы для поиска работы базируются на новых методах публикации вакансий, что позволяет пользователям быстро находить интересующие их предложения в разнообразном цифровом пространстве. Электронные платформы и информационные технологии стали неотъемлемой частью таких сервисов.

Эти сервисы значительно упрощают процесс поиска работы как для работодателей, так и для соискателей. Они расширяют рынки труда и позволяют потенциальным работникам находить подходящие вакансии в любой точке мира в любое удобное для них время. Продавцы услуг на рынке труда, ориентированные на интернет-сервисы, получают значительные преимущества, избегая посредников и имея доступ к широкому кругу потенциальных кандидатов. Это делает подобные сервисы более привлекательными по сравнению с классическими методами поиска работы.

Целью курсового проекта является разработка веб-приложения, предназначенного для автоматизации процесса поиска работы. Разработка системы автоматизации поиска работы требует комплексного подхода, включающего анализ потребностей пользователей, проектирование архитектуры системы, выбор технологий для реализации, а также тестирование и поддержку продукта.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ
   1. Описание предметной области

Интернет-сервис по поиску работы – это онлайн-платформа, предоставляющая пользователю доступ к рынку труда через веб-сервер. Электронная площадка для поиска работы, также называемая интернет-сервисом, позволяет множеству работодателей публиковать вакансии и соискателям находить подходящие вакансии.

Технический аспект интернет-сервиса по поиску работы подобен онлайн-витрине. Эта виртуальная площадка предоставляет интерфейс к базе данных с вакансиями (в виде каталога или списка), позволяя размещать вакансии, откликаться на них, сравнивать предложения и оставлять заявки на работу.

Ранее люди тратили много времени на поиск работы через газетные объявления или личные контакты. Однако с развитием интернет-технологий все изменилось. Теперь автоматизированные площадки для поиска работы объединяют в себе возможность размещения вакансий любым пользователем и доступ к этим объявлениям для множества соискателей.

Сегодня сервисы по поиску работы ориентированы на удобство пользователей. Период, когда поиск работы был связан с просмотром бумажных объявлений, ушел в прошлое. Современный поиск работы в базе данных позволяет сократить время на поиск и получить максимально полную информацию о вакансиях, а использование веб-технологий делает этот процесс еще более эффективным и удобным для всех участников.

Система «Веб-сервис для поиска работы и размещения вакансий» предоставляет эффективный сервис поиска вакансий и соискателей. Более удобное обслуживание в сфере поиска работы не только обеспечивает комфорт для пользователей, но и существенно упрощает труд работодателей и соискателей, повышает эффективность поиска и обработки информации с использованием современных инструментов.

* 1. Постановка задачи

Задачей данного курсового проекта является спроектировать и создать веб-сервис портала для поиска работы, а также спроектировать и разработать интерфейс работы с ним.

Необходимо описать архитектуру приложения использую диаграммы различных методологий.

Система должна предоставить пользователям эффективный и комфортный сервис по поиску работы.

Требуется обеспечить работу четырех групп пользователей: гостя, авторизованного пользователя, компании и администратора, которые должны быть наделены различными правами.

Гость должен иметь возможность просматривать доступные вакансии, искать необходимые вакансии по параметрам, подавать заявление на определенную должность.

Авторизованный пользователь должен иметь личный кабинет, где он смог бы увидеть и изменить свои данные и прикрепленное резюме, с которым работодатель может ознакомиться, страницу отправленных заявок и их статус, почтовый ящик, который выполняет роль связующего звена между кандидатом и работодателем, настройки аккаунта, позволяющие сменить пароль или удалить аккаунт.

Компания должна иметь доступ к статистике собственных вакансий, возможность просматривать и изменять общедоступную информацию о компании, блок создания вакансий, список текущих вакансий и список откликов на эти вакансии, почтовый ящик, который выполняет роль связующего звена между кандидатом и работодателем, настройки аккаунта, позволяющие сменить пароль или удалить аккаунт, сводку о всех кандидатах с возможностью загрузки резюме каждого.

Администратор должен иметь право видеть статистику по сервису, все объявления и по желанию удалять пользователей, компании и объявления, которые не проходят по определенным параметрам. Только после принятия заявки компании на вступление ей будет предоставлен доступ к пользованию платформой.

Приложение должно обеспечивать безопасность данных пользователей. Никто кроме самих пользователей не должен изменять их данные или получать доступ к ним.

1. ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ СИСТЕМЫ
   1. Концептуальная модель системы

В системе портала для поиска работы можно выделить следующих действующих лиц:

– гость;

– пользователь;

– компания;

– администратор.

Гости – это клиенты, не прошедшие регистрацию. Пользователи – это клиенты данной системы, которые выполнят различные операции с системой и своим аккаунтом. Компания – это аккаунт компании, которая зарегистрирована на портале. Администратор – это особый пользователь, который выполняет управление системой, следит за выполнением пользователями ее правил, может создавать отчеты на основе своих действий и отслеживать состояние системы в виде графиков и диаграмм.

Исходя из потребностей действующих лиц, выделяются следующие варианты использования для гостя:

– регистрация: позволяет гостю зарегистрироваться в системе;

– авторизация: дает возможность зайти в собственный профиль в системе;

– просмотр вакансий: позволяет просматривать вакансии;

В соответствии с потребностями пользователя выделяются следующие варианты использования для пользователя:

– регистрация: позволяет гостю зарегистрироваться в системе;

– авторизация: дает возможность зайти в собственный профиль в системе;

– выход из учетной записи: позволяет выйти из профиля;

– отправка писем: дает возможность отправлять письма;

– восстановление пароля: позволяет восстановить забытый пароль;

– изменение данных: позволяет изменять личные данные;

– просмотр заявок: позволяет просматривать заявки;

В соответствии с потребностями компании выделяются следующие варианты использования для пользователя:

– регистрация: позволяет гостю зарегистрироваться в системе;

– авторизация: дает возможность зайти в собственный профиль в системе;

– выход из учетной записи: позволяет выйти из профиля;

– отправка писем: дает возможность отправлять письма;

– восстановление пароля: позволяет восстановить забытый пароль;

– изменение данных: позволяет изменять личные данные;

– просмотр заявок: позволяет просматривать заявки;

– просмотр статистики вакансий: позволяет просматривать статистику вакансий;

– добавление вакансии: дает возможность добавлять вакансии;

– просмотр данных кандидатов: позволяет просматривать данные кандидатов;

– просмотр вакансий компании: дает возможность просматривать вакансии;

В соответствии с потребностями администратора выделены следующие варианты использования:

– авторизация: позволяет администратору зайти в панель управления;

– выход из учетной записи: выход из панели управления;

– просмотр статистики сайта: позволяет просматривать статистику сайта;

– управление учетной записью: позволяет удалять, активировать, просматривать учетные записи;

– управление вакансиями: дает возможность удалять либо просматривать вакансии;

Диаграмма вариантов использования для системы портала для поиска работы представлена на рисунке 2.1.

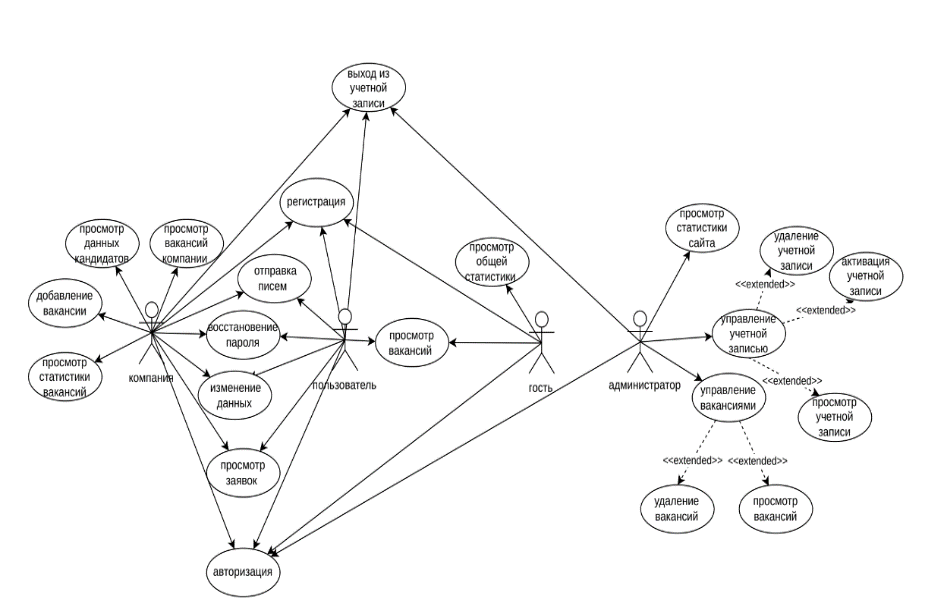


Рисунок 2.1 – Диаграмма вариантов использования

Для отображения объектной модели системы используется диаграмма классов.

Диаграмма классов – структурная [диаграмма](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_(UML)) языка моделирования [*UML*](https://ru.wikipedia.org/wiki/UML), демонстрирующая общую структуру иерархии [классов](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) системы, их коопераций, [атрибутов](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D0%B5_%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B0) (полей), [методов](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_(%D1%8F%D0%B7%D1%8B%D0%BA%D0%B8_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)), интерфейсов и взаимосвязей (отношений) между ними

Диаграмма классов представлена на рисунке 2.2.

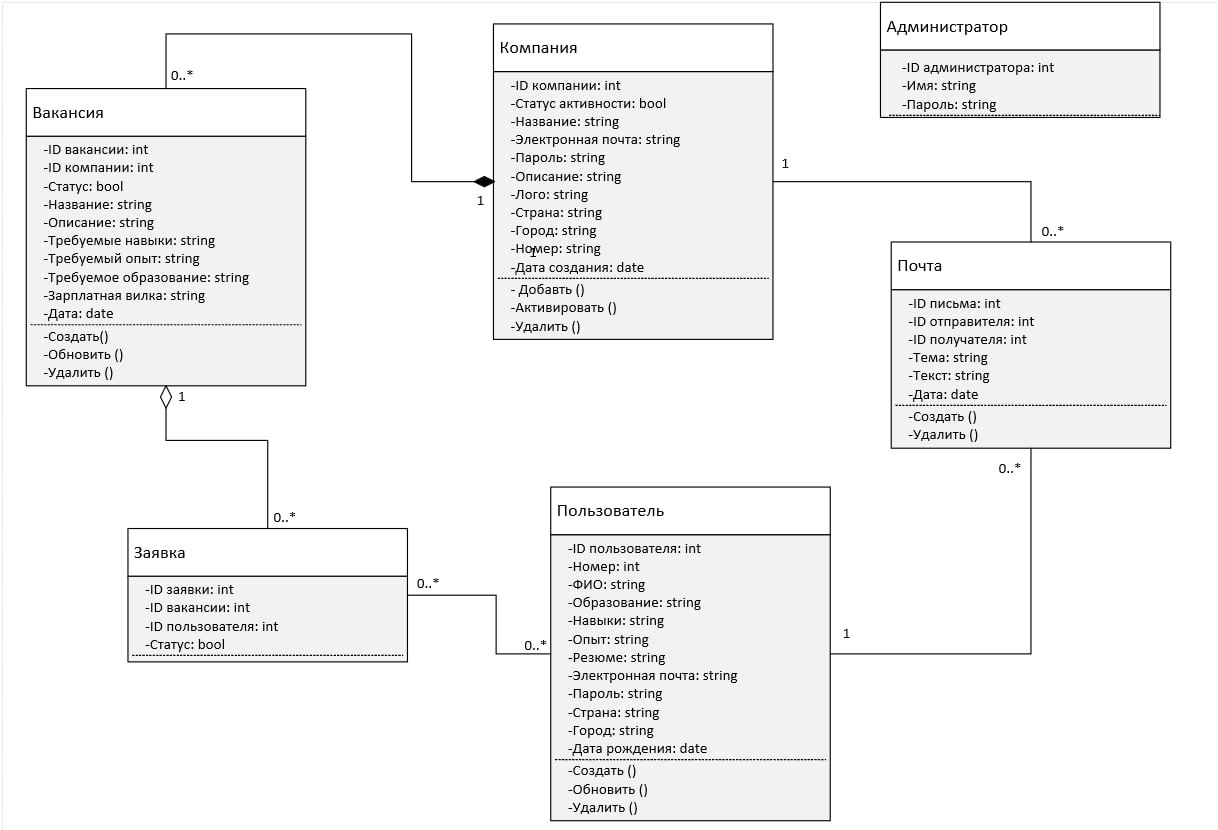


Рисунок 2.2 – Диаграмма классов

На диаграмме классов выделены шесть классов, которые соответствуют сущностям системы. Каждый класс имеет свои поля и методы. Между классами существуют различные виды отношений. На диаграмме представлены два из всех видов: агрегация и композиция.

На диаграмме классов представлены следующие классы:

– Вакансия;

– Компания;

– Администратор;

– Почта;

– Пользователь;

– Заявка.

Каждый класс имеет свои поля и методы. Поля представлены на рисунке 2.4 для каждого класса. Поле имеет свое название и тип данных, которое оно хранит.

У классов Вакансия, Почта, Пользователь есть метод Создать(). Данный метод служит для занесения информации в поля класса.

У классов Вакансия, Компания, Почта, Пользователь есть метод Удалить(), который служит для удаления информации из полей класса.

Класс Компания имеет методы Активировать() и Добавить(). Метод Добавить() добавляет информацию в поля класса, а Активировать() активирует данные.

У классов Вакансия и Пользователь есть метод Обновить(), который служит для обновления информации в полях класса.

2.3 Алгоритмическое обеспечение

Опишем алгоритмическое обеспечение с помощью диаграмм *UML* состояния и деятельности.

Диаграмма состояний описывает состояния и условия перехода между ними для определенного объекта в системе. Чаще всего диаграммы состояний используются для описания поведения отдельных экземпляров классов (объектов), но они также могут быть применены для спецификации функциональности других компонентов моделей, таких как варианты использования, актеры, подсистемы, операции и методы. Рассмотрим такую диаграмму для компании в системе.

Компания может находтся в следующих состояних:

– компания зарегестрирована;

– компания не верифицирована;

– компания верифицирована;

– компания заморожена;

Изначально все компании находятся в начальном своем состянии. Если администратор активировал профиль, компания переходит во второе состояние. Если администратор не активировал профиль, компания переходит в третье состояние. Из этого состояния компания может перейти во второе состояние, если администратор активирует профиль. Если администратор блокирует профиль, компания переходит в четвертое состояние. Четвертое состояние песни является конечным ее состоянием [6].

Диграмма состояний представлена на рисунке 2.3.

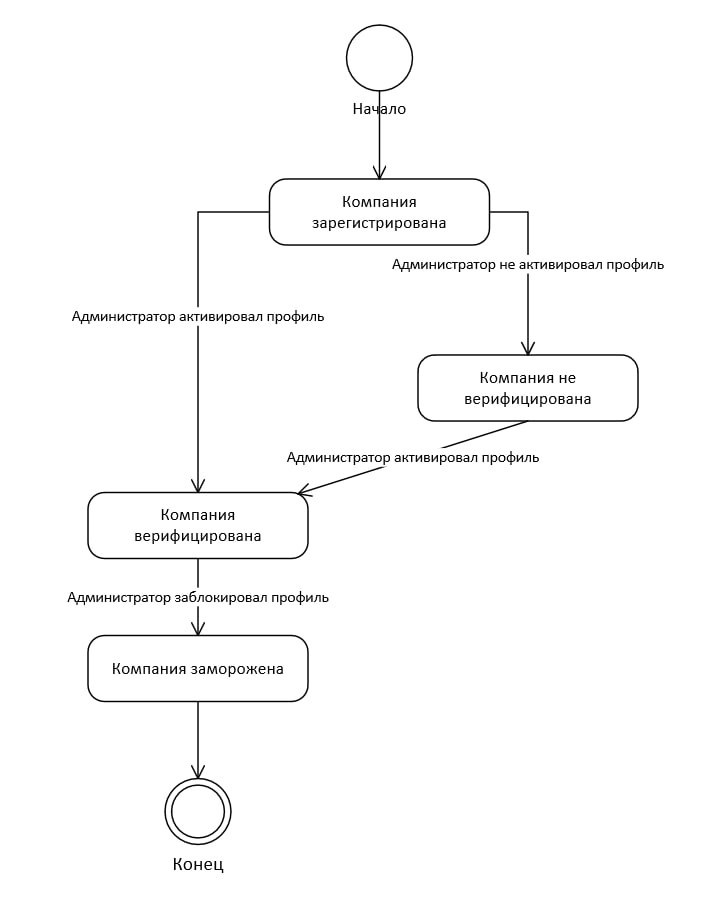


Рисунок 2.3 – Диаграмма состояний

1. РЕАЛИЗАЦИЯ ВЕБ-ПРИЛОЖЕНИЯ
   1. Техническое обеспечение

Рассмотрим выбор языка программирования клиентской части приложения.

*Node.js* — это среда, которая позволяет использовать *JavaScript* как для бэкенд, так и для фронтенд разработки, а также для решения проблем совместимости. *Node.js* также можно определить как язык сценариев на стороне сервера. Он был запущен в 2009 году и постепенно набирает популярность.

Преимущества *Node.js*:

– возможность применять один язык на клиенте и сервере. Если программист прокачался в *JavaScript*, ему будет легче выучить “надстройку”, чем кардинально отличающуюся технологию. Разрабатывался специально для *web*. Свободно взаимодействует с наиболее популярными базами данных, помогает получить низкоуровневый доступ (*http*, *udp*, *https*, *tcp*);

– постоянное развитие экосистемы. Количество готовых модулей и внешних библиотек беспрерывно растет, в чем способствует использование npm. Движок *V8*. Он экономно расходует память, неплохо оптимизирован, дает функционал по профилированию процессора и памяти.

Недостатки *Node.js*:

– *Node.js* может быть менее эффективным при обработке *CPU*-интенсивных задач, так как он работает в одном потоке. Однако существуют методы для обработки таких задач с использованием фоновых процессов или делегирования работы на другие языки или сервисы;

– приложения, основанные на *Node.js*, могут быть сложнее в разработке и поддержке из-за асинхронной природы языка и необходимости управления обратными вызовами (*callbacks*) или использования синтаксических конструкций, таких как промисы или асинхронные функции.

*C#* – это язык с *C*-подобным синтаксисом. Здесь он близок в этом отношении к *C++* и *Java*.

Будучи объектно-ориентированным языком, он много перенял у *Java* и С++. Объектно-ориентированный подход позволяет строить с помощью *C#* крупные, но в то же время гибкие, масштабируемые и расширяемые приложения.

Преимущества *C#*:

– поддержка подавляющего большинства продуктов *Microsoft*;

– автоматическая «сборка мусора»: это значит, что в большинстве случаев не придётся заботиться об освобождении памяти; общеязыковая среда *CLR* сама вызовет сборщик мусора и очистит память;

– низкий порог вхождения.

Недостатки *C*#:

– приоритетная ориентированность на платформу *Windows*;

– язык бесплатен только для небольших фирм, индивидуальных программистов, стартапов и учащихся, а крупной компании покупка лицензионной версии этого языка обойдётся в круглую сумму.

*PHP* – язык программирования, исполняемый на стороне веб-сервера, спроектированный в качестве инструмента создания динамических и интерактивных веб-сайтов.

Этот язык оказался достаточно гибким и мощным, поэтому приобрёл большую популярность и используется в проектах любого масштаба: от простого блога до крупнейших веб-приложений в Интернете.

Преимущества *PHP*:

– *PHP* является свободным и открытым программным обеспечением, доступным для всех. Благодаря этому, разработчики могут использовать *PHP* для создания приложений без необходимости платить за лицензии или ограничений;

– легок в освоении на всех этапах;

–поддерживается большим сообществом пользователей и разработчиков;

– имеет развитую поддержку баз данных;

– имеется огромное количество библиотек и расширений языка;

–богатая функциональность *PHP* предлагает множество встроенных функций и библиотек, которые облегчают разработку веб-приложений. Они включают функции для работы с файлами, строками, сетью, отправки электронной почты, создания изображений и многое другое;

– предлагает нативные средства организации веб-сессий, программный интерфейс расширений;

– *PHP* является кросс-платформенным языком, что означает, что приложения, написанные на PHP, могут работать на различных операционных системах, таких как *Windows*, *macOS* и *Linux*, без необходимости внесения существенных изменений;

– *PHP* может интегрироваться с другими технологиями, такими как *HTML*, *CSS*, *JavaScript*, *XML* и многое другое. Это позволяет разработчикам создавать полнофункциональные веб-приложения, которые взаимодействуют с различными компонентами и сервисами;

– может быть развёрнут почти на любом сервере;

Учитывая возможности *PHP*, а также собственный опыт разработки, для реализации функционала данного проекта был выбран язык программирования *PHP*. Исходя из требований проекта, он отлично подходит для поставленной задачи.

* 1. Программное обеспечение

Приведем программный код основных функций приложения, где видна реализация работы с базой данных с кратким пояснением.

Функция создания компании представлена в приложении А к пояснительной записке.

Функция добавления пользователя:

*//sql new registration insert query*

*$sql = "INSERT INTO users(firstname, lastname, email, password, address, city, state, contactno, qualification, stream, passingyear, dob, age, designation, resume, hash, aboutme, skills) VALUES ('$firstname', '$lastname', '$email', '$password', '$address', '$city', '$state', '$contactno', '$qualification', '$stream', '$passingyear', '$dob', '$age', '$designation', '$file', '$hash', '$aboutme', '$skills')";*

*if($conn->query($sql)===TRUE) {*

*// //If data inserted successfully then Set some session variables for easy reference and redirect to login*

*$\_SESSION['registerCompleted'] = true;*

*header("Location: login-candidates.php");*

*exit();*

*} else {*

*//If data failed to insert then show that error. Note: This condition should not come unless we as a developer make mistake or someone tries to hack their way in and mess up :D*

*echo "Error " . $sql . "<br>" . $conn->error;*

*}*

*} else {*

*//if email found in database then show email already exists error.*

*$\_SESSION['registerError'] = true;*

*header("Location: register-candidates.php");*

*exit();*

*}*

*//Close database connection. Not compulsory but good practice.*

*$conn->close();*

*} else {*

*//redirect them back to register page if they didn't click register button*

*header("Location: register-candidates.php");*

*exit();*

*}*

В данной функции в базу данных добавляется информация о пользователе. Если добавление прошло успешно, то данные записываются в поля структуры *Use*r и передаются на сервер. В противном случае выводится ошибка.

Функция добавления вакансии:

*<?php*

*//To Handle Session Variables on This Page*

*session\_start();*

*if(empty($\_SESSION['id\_company'])) {*

*header("Location: ../index.php");*

*exit();*

*}*

*//Including Database Connection From db.php file to avoid rewriting in all files*

*require\_once("../db.php")*

*//if user Actually clicked Add Post Button*

*if(isset($\_POST)) {*

*// New way using prepared statements. This is safe from SQL INJECTION. Should consider to update to this method when many people are using this method.*

*$stmt = $conn->prepare("INSERT INTO job\_post(id\_company, jobtitle, description, minimumsalary, maximumsalary, experience, qualification) VALUES (?,?, ?, ?, ?, ?, ?)");*

*$stmt->bind\_param("issssss", $\_SESSION['id\_company'], $jobtitle, $description, $minimumsalary, $maximumsalary, $experience, $qualification);*

*$jobtitle = mysqli\_real\_escape\_string($conn, $\_POST['jobtitle']);*

*$description = mysqli\_real\_escape\_string($conn, $\_POST['description']);*

*$minimumsalary = mysqli\_real\_escape\_string($conn, $\_POST['minimumsalary']);*

*$maximumsalary = mysqli\_real\_escape\_string($conn, $\_POST['maximumsalary']);*

*$experience = mysqli\_real\_escape\_string($conn, $\_POST['experience']);*

*$qualification = mysqli\_real\_escape\_string($conn, $\_POST['qualification']);*

*$skills = mysqli\_real\_escape\_string($conn, $\_POST['skills']);*

*if($stmt->execute()) {*

*//If data Inserted successfully then redirect to dashboard*

*$\_SESSION['jobPostSuccess'] = true;*

*header("Location: index.php");*

*exit();*

*} else {*

*echo "Error ";*

*}*

*$stmt->close();*

*$conn->close();*

*} else {*

*header("Location: create-job-post.php");*

*exit();*

*}*

В начале данной функции происходит проверка на подключение к базе данных. Далее происходит проверка на корректность данных. Если все верно, то происходит добавление вакансии в базу данных и в интерфейс приложения.

Функция удаления компании:

<?php

session\_start();

if(empty($\_SESSION['id\_admin'])) {

header("Location: index.php");

exit();

}

require\_once("../db.php");

if(isset($\_GET)) {

//Delete Company using id and redirect

$sql = "DELETE FROM company WHERE id\_company='$\_GET[id]'";

if($conn->query($sql)) {

header("Location: companies.php");

exit();

} else {

echo "Error";

}

}

В данной функции происходит запрос в базу данных на выборку компаний по *id*. Если запрос прошел некорректно, то возвращается ошибка. Если все прошло успешно, то компания удаляется.

* 1. Руководство пользователя

При входе в систему пользователь попадает на главную страницу. На данной странице находятся: меню, доступные к просмотру объявления, кнопка поиска. Вид главной страницы приложения представлен на рисунке 3.1.

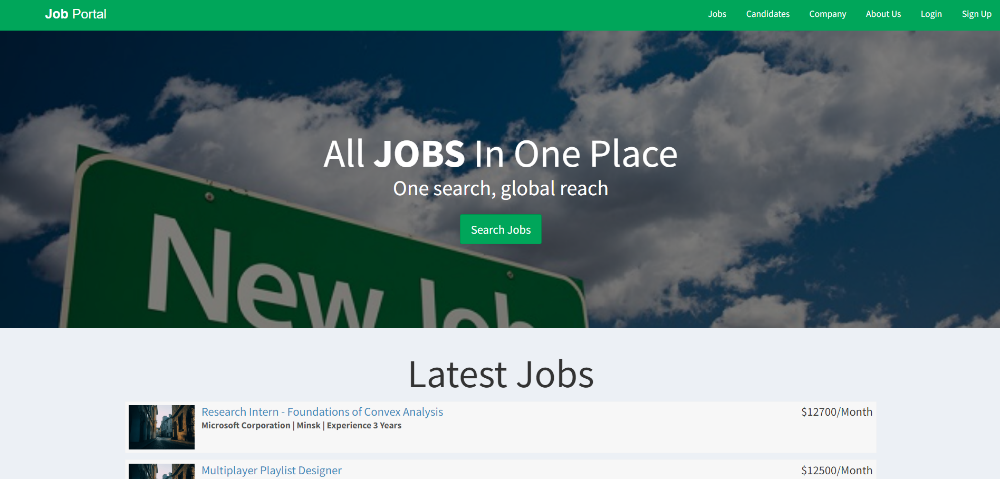


Рисунок 3.1 – Главная страница

У гостя есть возможность просмотреть недавно загруженные работы. Страница, содержащая недавно загруженные работы, представлена на рисунке 3.2.

Рисунок 3.2 – Последние вакансии

На главной странице также есть статистика сайта. Внешний вид статистики показан на рисунке 3.3.

Рисунок 3.3 – Статистика

Гостю предоставляется возможность пройти регистрацию на сайте. Можно пройти регистрацию как за пользователя, так и за компанию. Внешний вид окна регистрации представлен на рисунке 3.4.

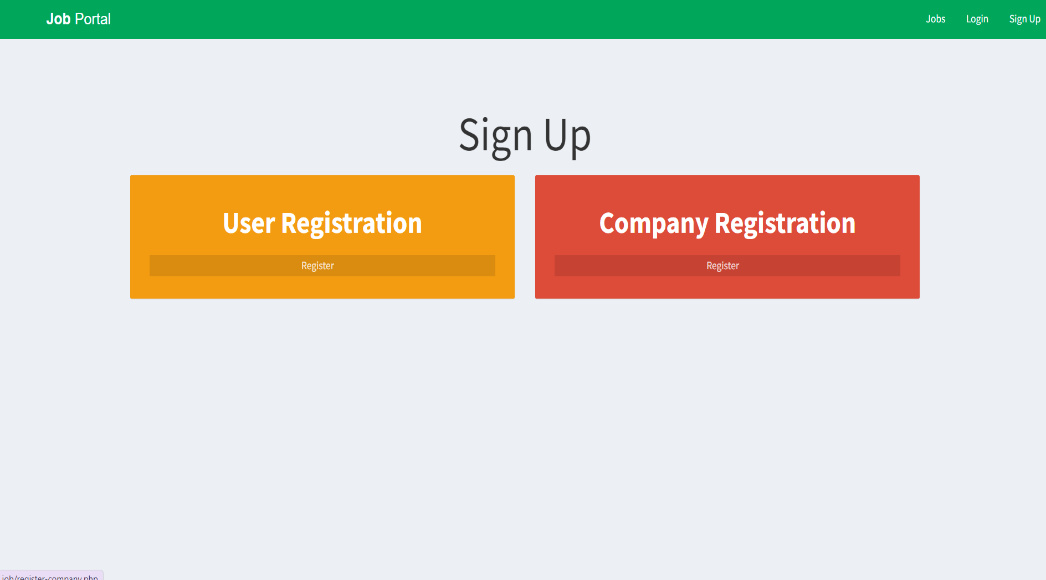


Рисунок 3.4 – Регистрация на сайте

Также гостю предоставляется возможность войти в сайт. Можно войти как за пользователя, так и за компанию. Внешний вид окна входа представлен на рисунке 3.5.

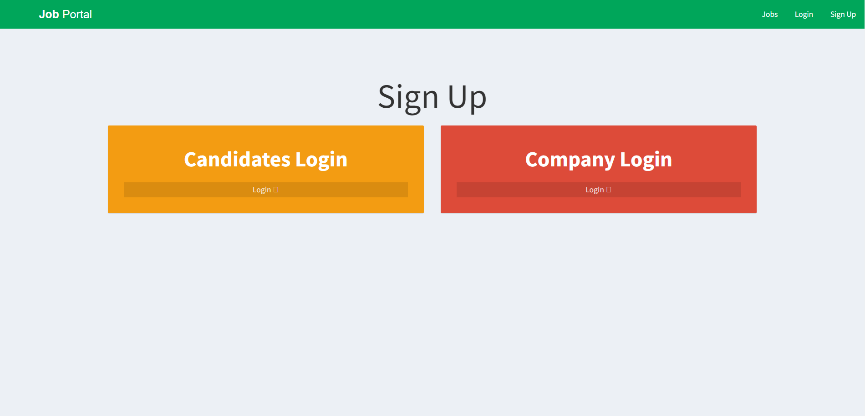


Рисунок 3.5 – Вход в сайт

Страница входа представлена на рисунке 3.6.

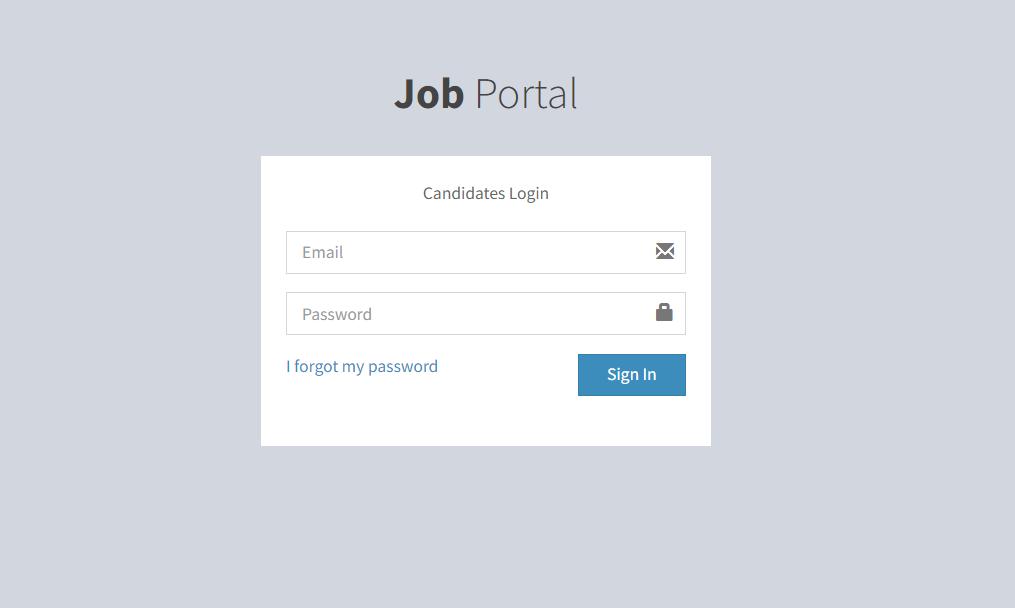


Рисунок 3.6 – Страница входа

Пользователь имеет возможность восстановить пароль. Страница смены пароля представлена на рисунке 3.7.

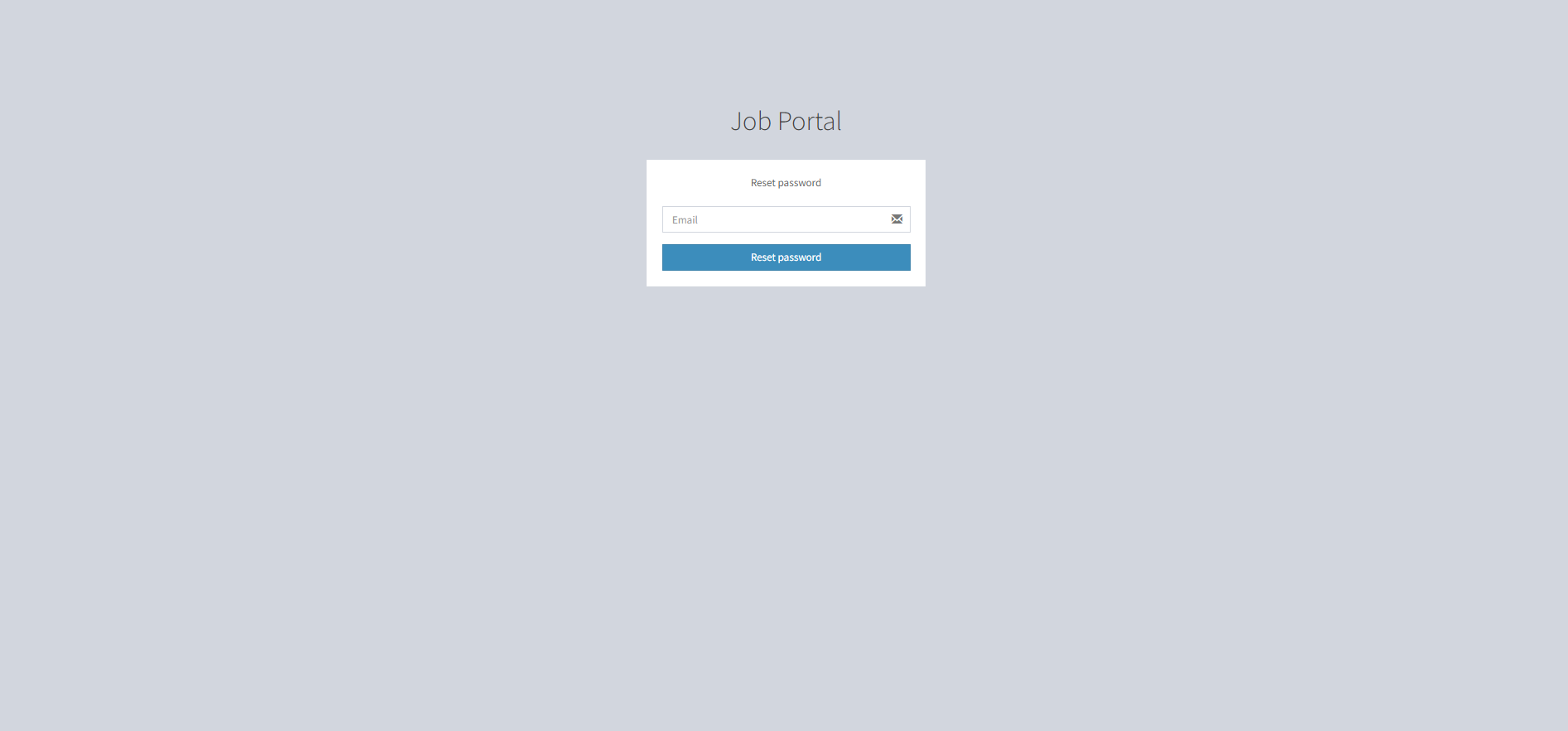


Рисунок 3.7 – Страница смены пароля

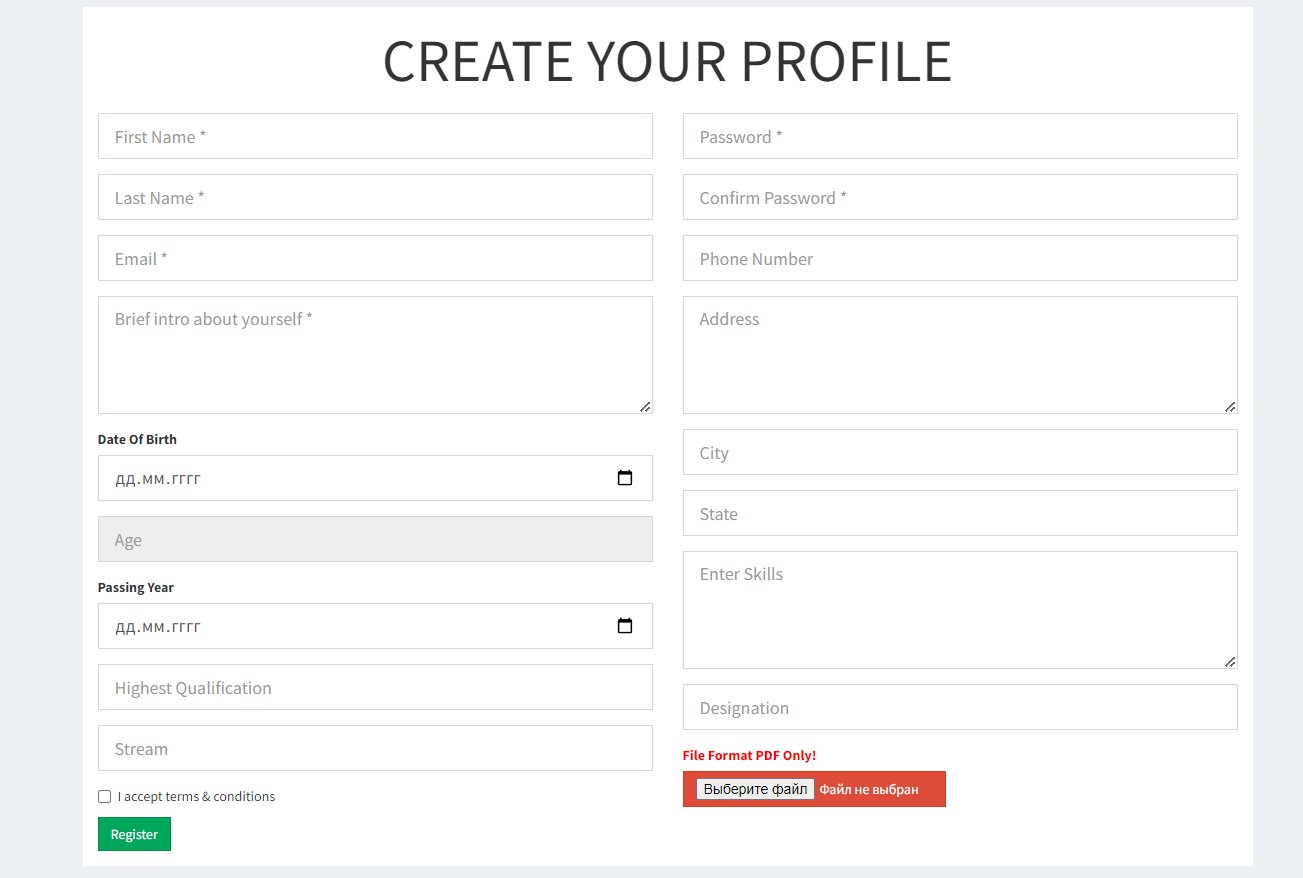
После регистрации пользователь имеет возможность заполнить свою анкету. Страница представлена на рисунке 3.8.

Рисунок 3.8 – Страница заполнения анкеты пользователя

После регистрации компании можно также заполнить анкету для компании. Страница с данным разделом представлена на рисунке 3.9.

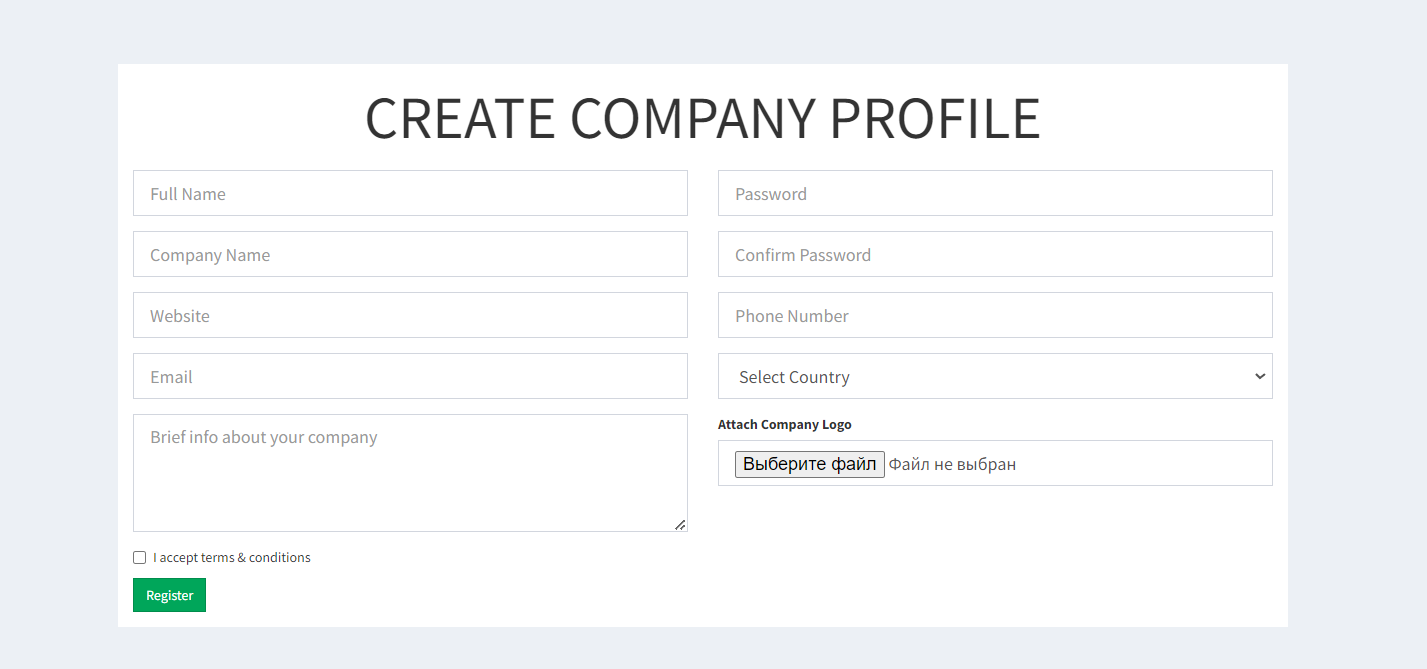


Рисунок 3.9 – Страница заполнения анкеты компании

Авторизованный пользователь имеет возможность зайти в личный кабинет. В личном кабинете пользователь может изменить информацию о себе. Внешний вид личного кабинета представлен на рисунке 3.10.

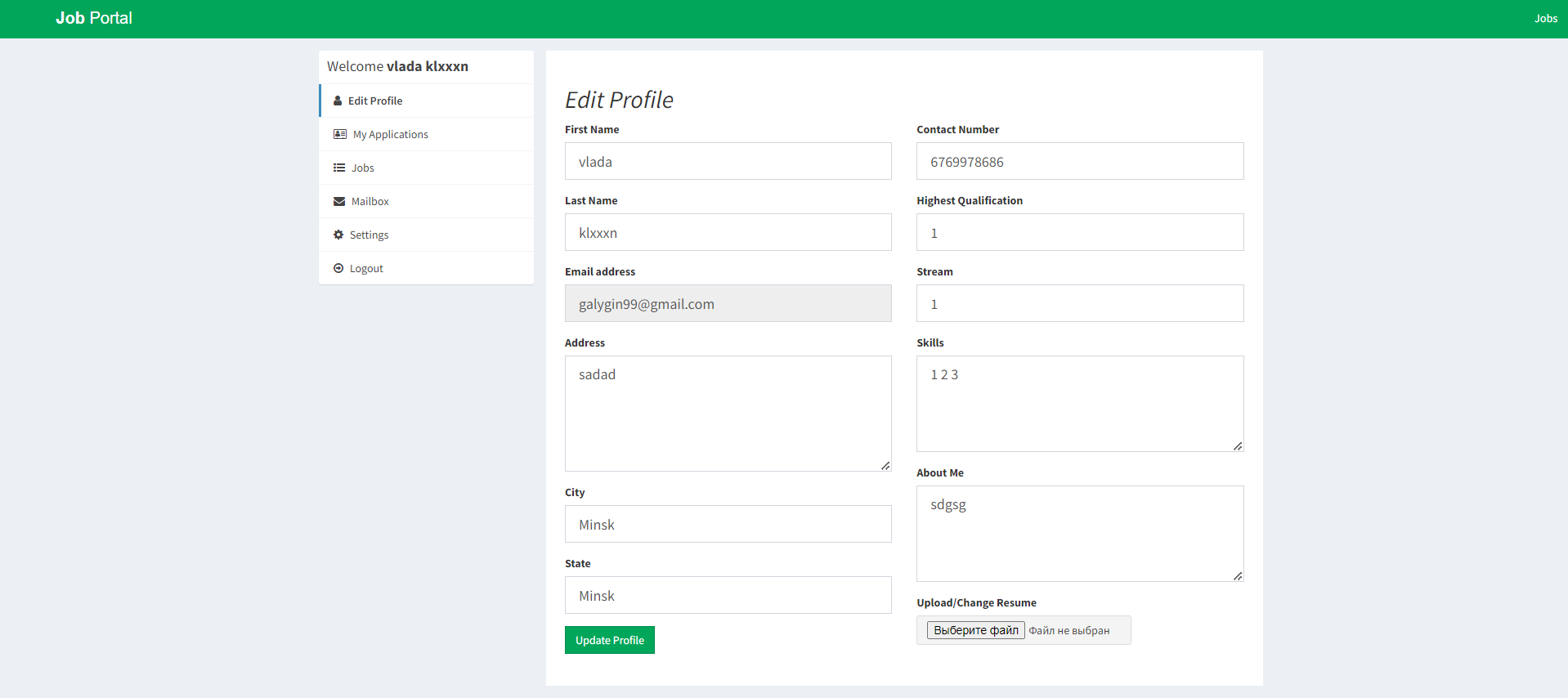


Рисунок 3.10 – Раздел с добавлением нового объявления

Также в личном кабинете пользователь может наблюдать отклики на работу. Это показано на рисунке 3.11.

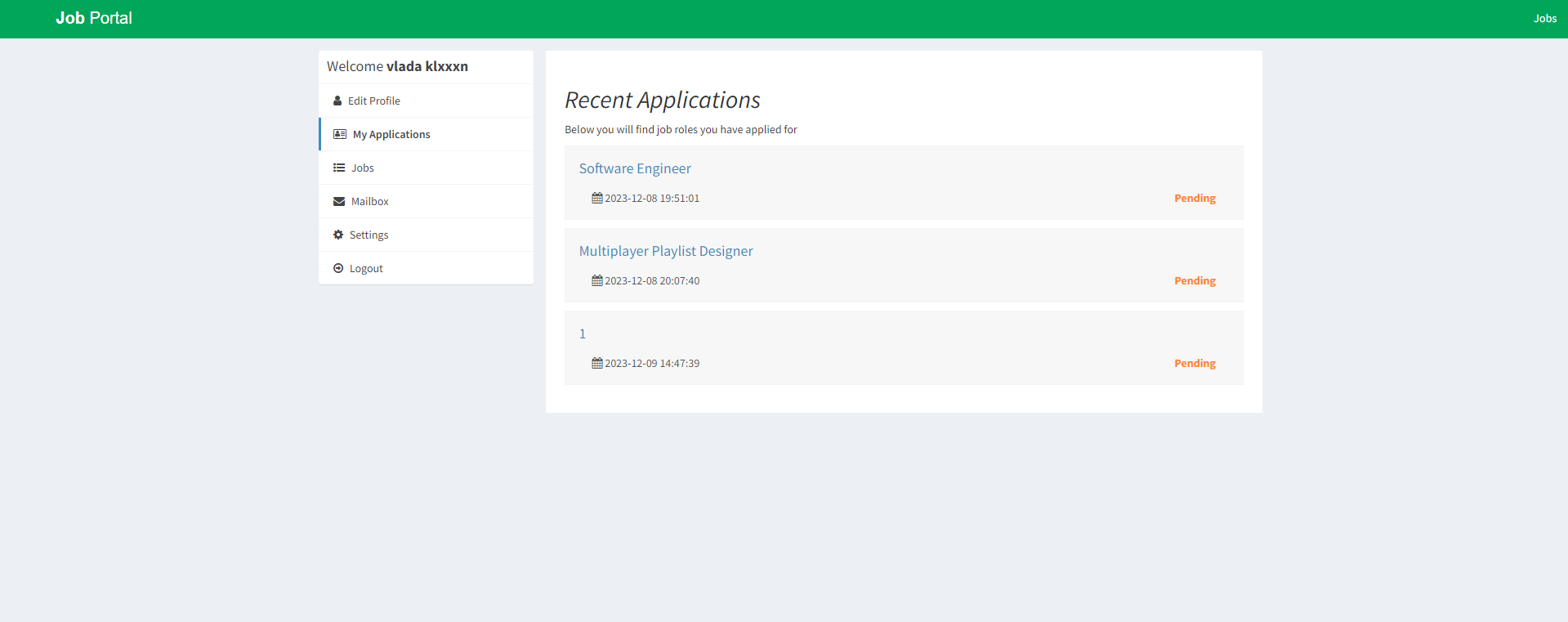


Рисунок 3.11 – Раздел откликов на вакансии

Из личного кабинета пользователь может зайти на доску вакансий и подать заявку на выбранную должность. Это показано на рисунке 3.12.

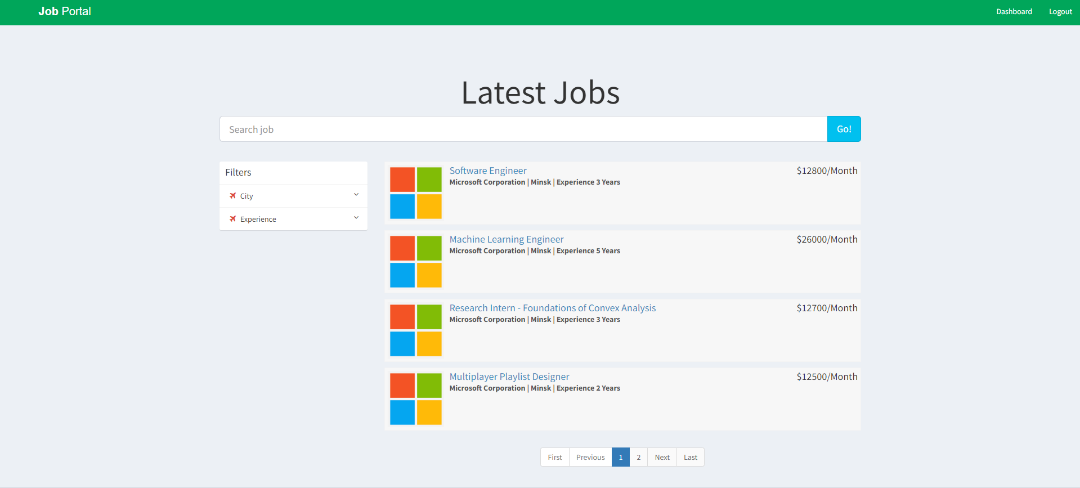


Рисунок 3.12 – Раздел с размещенными вакансиями

Из личного кабинета пользователь может зайти на страницу почты и прочитать, написать сообщение компании, которая уже утвердила кандидата на одну из должностей. Это показано на рисунке 3.13.

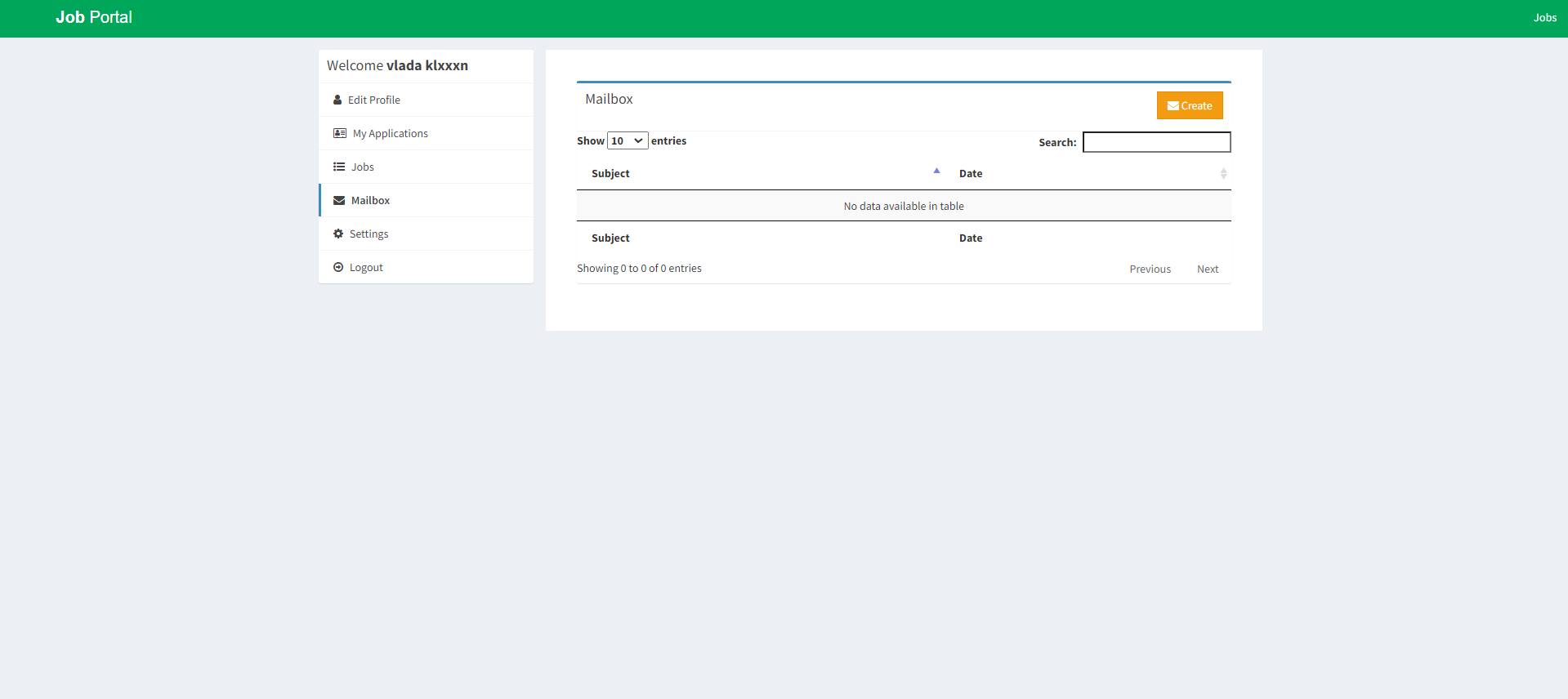


Рисунок 3.13 – Страница почты

Авторизованная компания имеет возможность зайти в личный кабинет. В личном кабинете компания может увидеть общую статистику по откликам. Внешний вид личного кабинета представлен на рисунке 3.14.

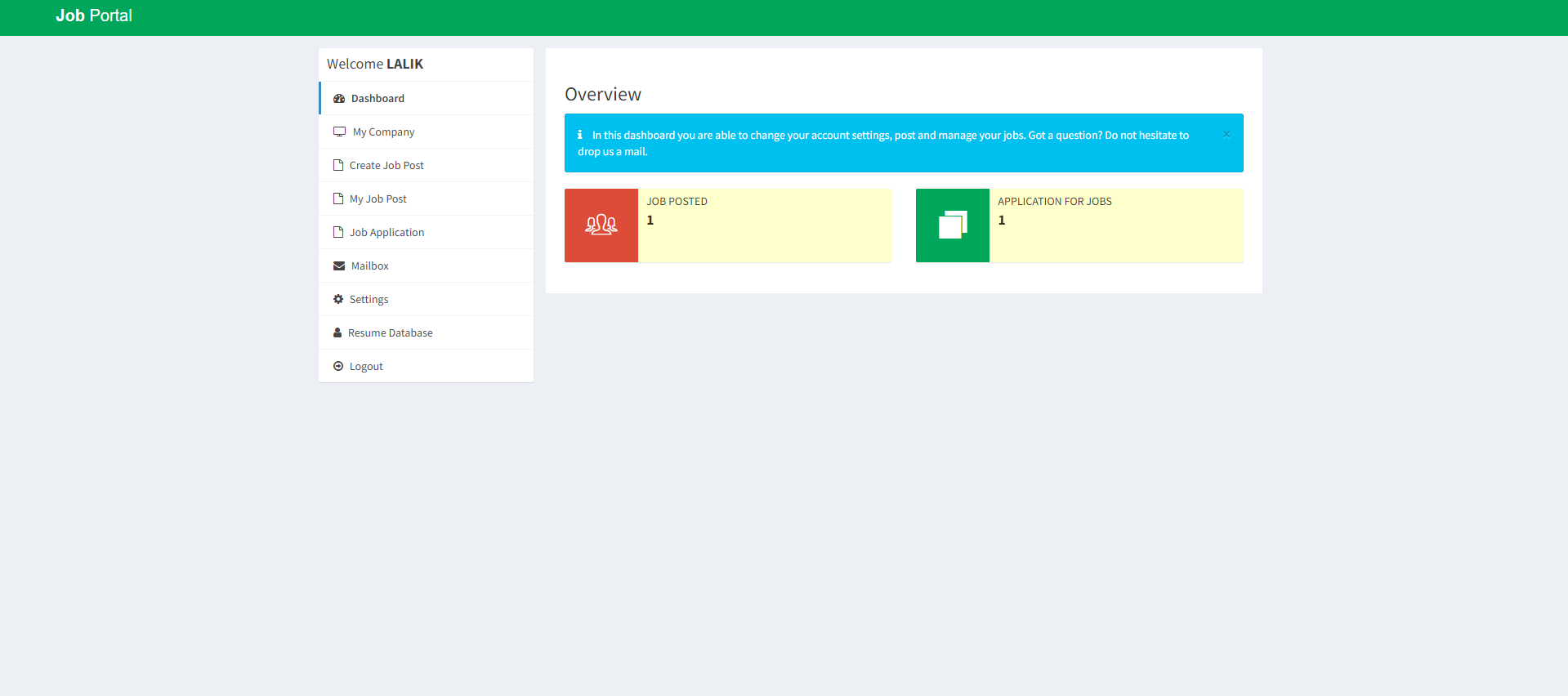


Рисунок 3.14 – Страница личного кабинета

Авторизованная компания имеет возможность зайти в личный кабинет. В личном кабинете компания может изменить информацию о себе. Внешний вид личного кабинета представлен на рисунке 3.15.

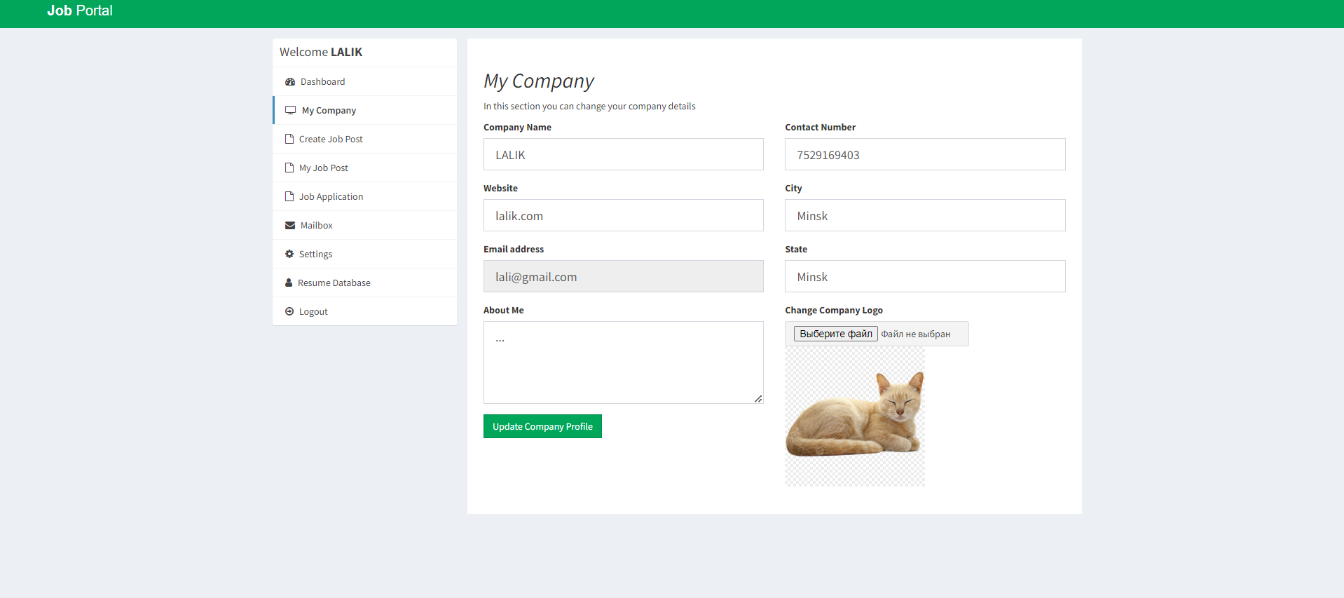


Рисунок 3.15 – Страница изменения личного кабинета

На рисунке 3.16 представлена форма создания новой вакансии.

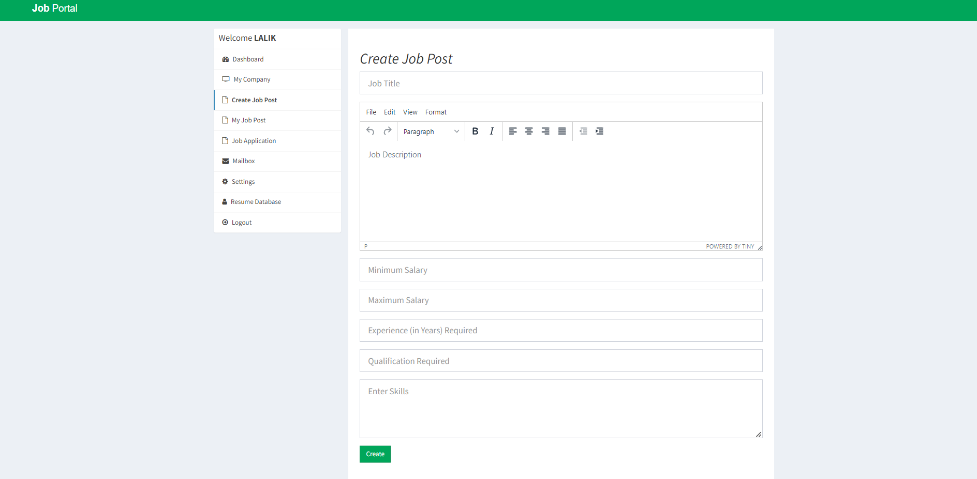


Рисунок 3.16 – Раздел создания новой вакансии

На рисунке 3.17 страница, на которой перечислены все отклики на вакансии и меню с ответом на заявку.

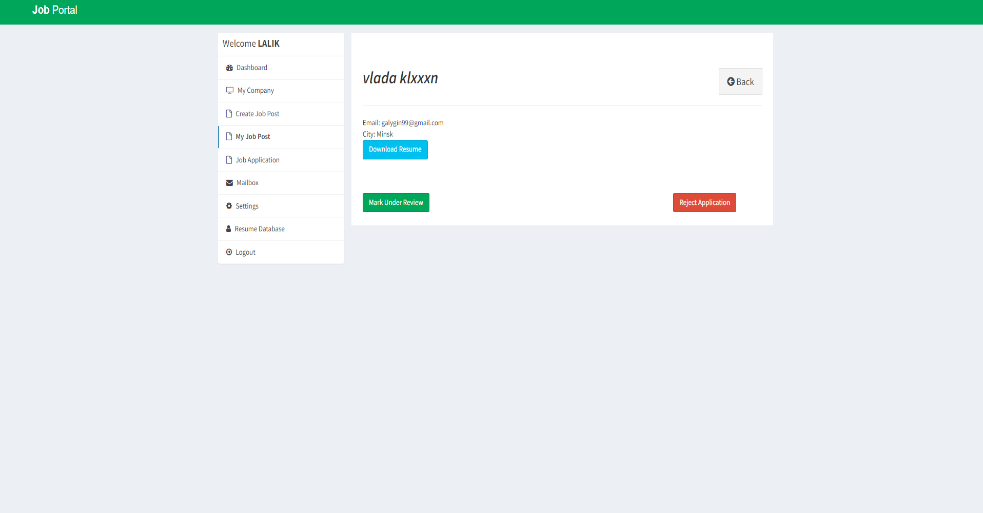


Рисунок 3.17 – Отклики на вакансии

На рисунке 3.18 представлена страница почты.

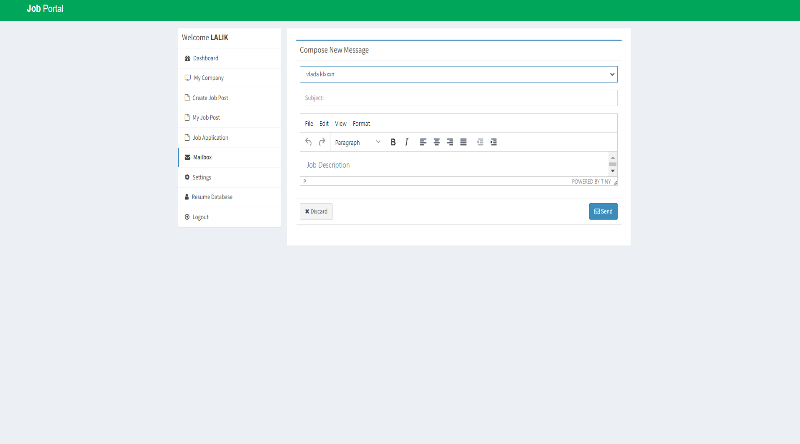


Рисунок 3.18 – Страница почты

В личном кабинете пользователь может изменить пароль. Это показано на рисунке 3.19.

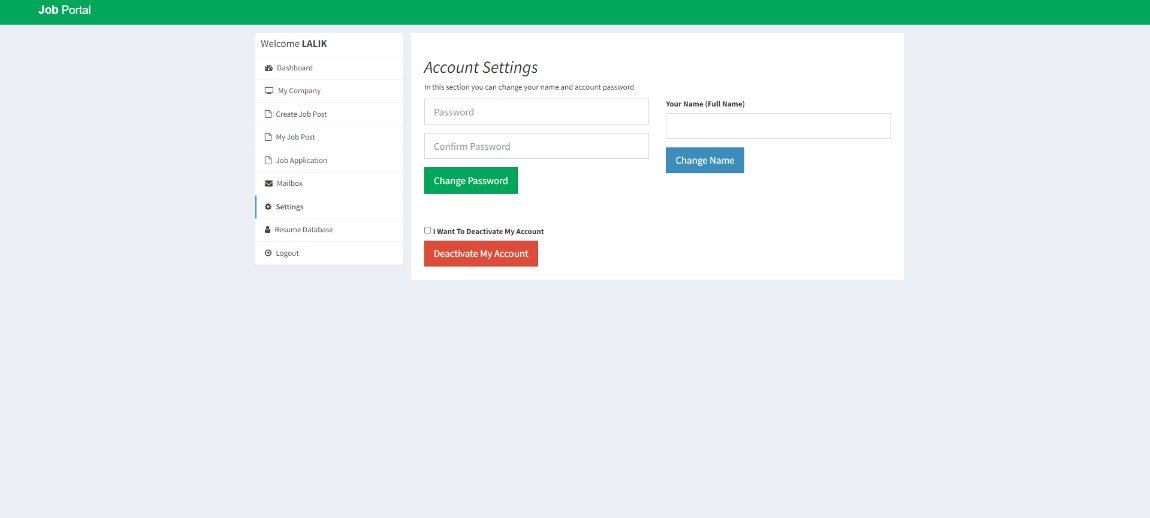


Рисунок 3.20 – Раздел с изменением пароля и деактивацией

На рисунке 3.21 представлена страница всех кандидатов с возможностью скачать резюме каждого.

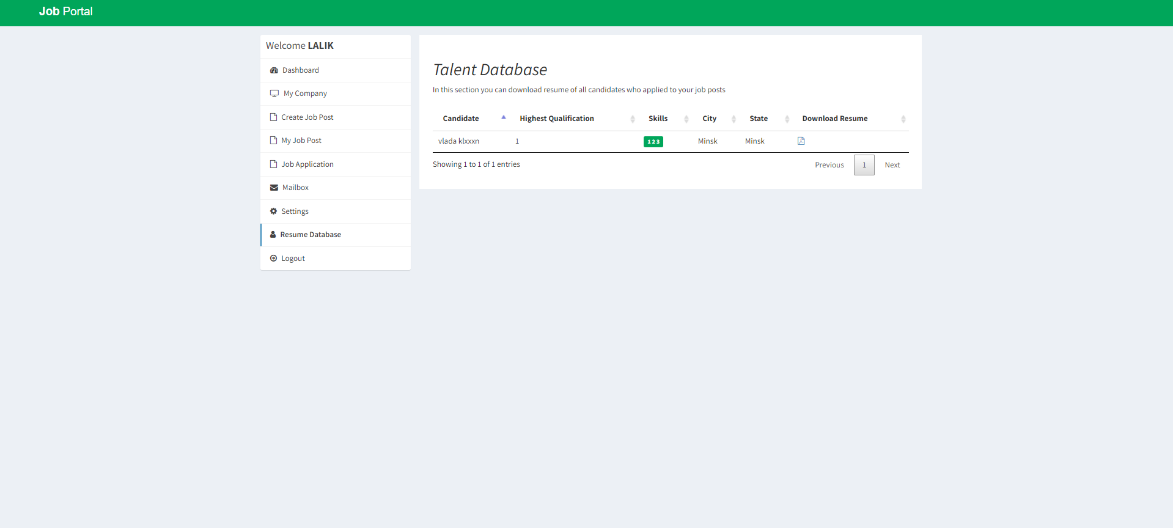


Рисунок 3.21 – Раздел с базой данных кандидатов

После авторизации пользователя с ролью администратора, он получает возможность для удаления автомобилей в интерфейсе главной страницы при нажатии на кнопку «*Delete*» в карточке объявления, рисунок 3.22. У записей, которые можно удалить, доступны кнопки удаления, то есть они красного цвета. Чтобы удалить запись, необходимо в соответствующей ей месте нажать кнопку удаления.

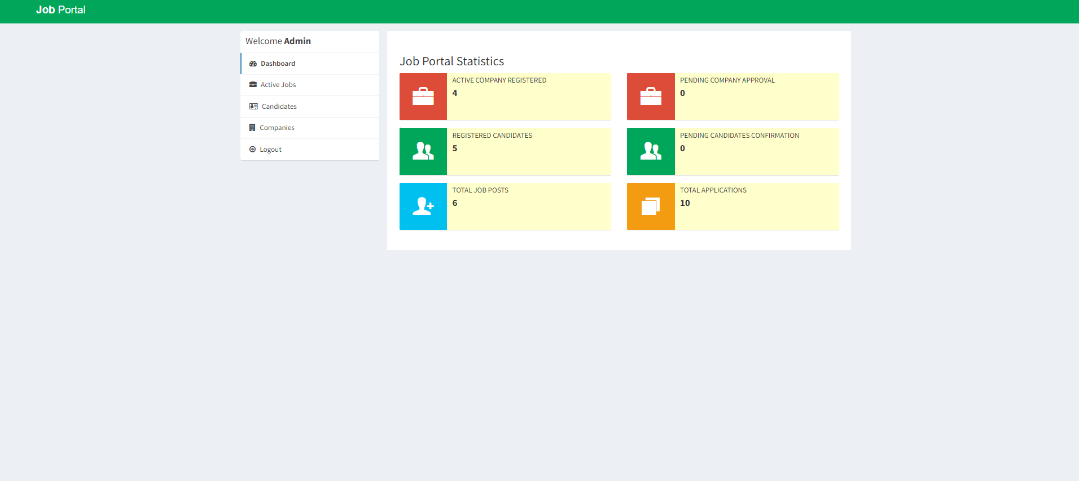


Рисунок 3.22 – Главная страница администратора

Администратор имеет возможность управлять вакансиями на специальной странице. Также в панели администратора предусмотрена страница для управления кандидатами. Страница управления вакансиями отображена на рисунке 3.23, а страница для управления кандидатами отображена на рисунке 3.24.

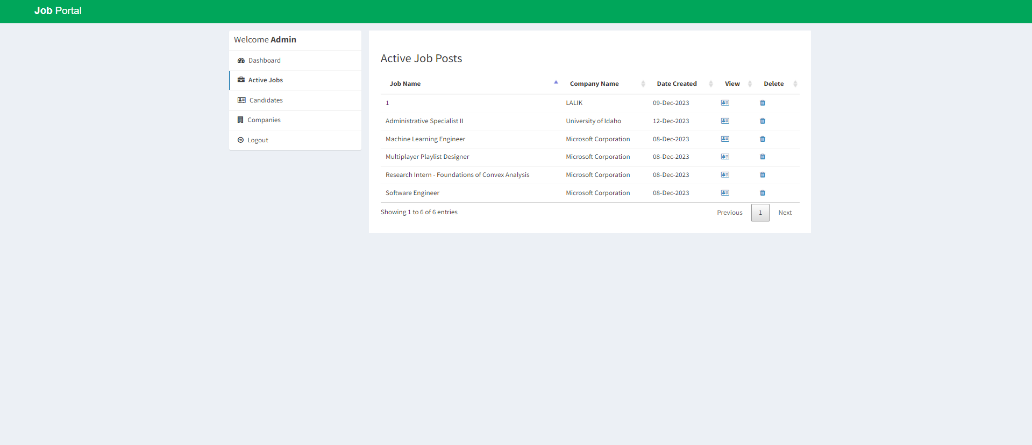


Рисунок 3.23 – Страница управления вакансиями

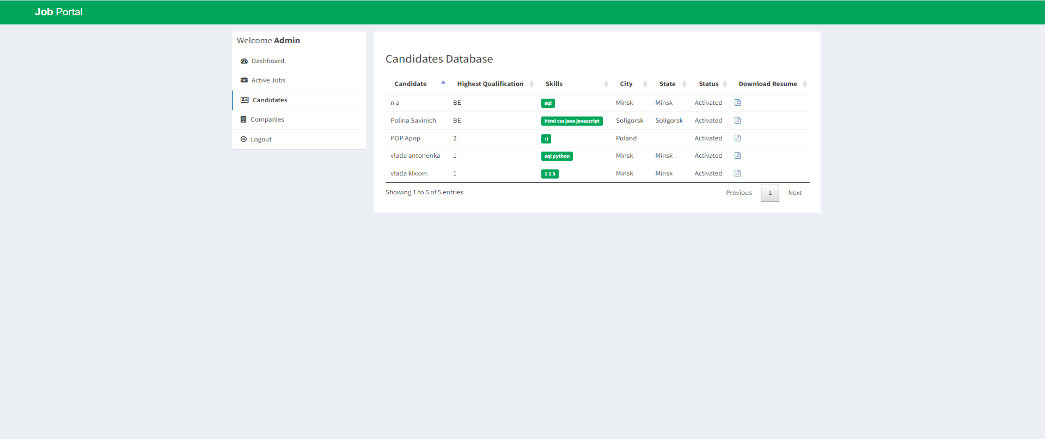


Рисунок 3.24 – Страница управления кандидатами

В панели администратора также реализована страница управления компаниями, на которой администратор может видеть всю информацию о компаниях, удалять компании или менять их статус. Страница управления компаниями изображена на рисунке 3.25.

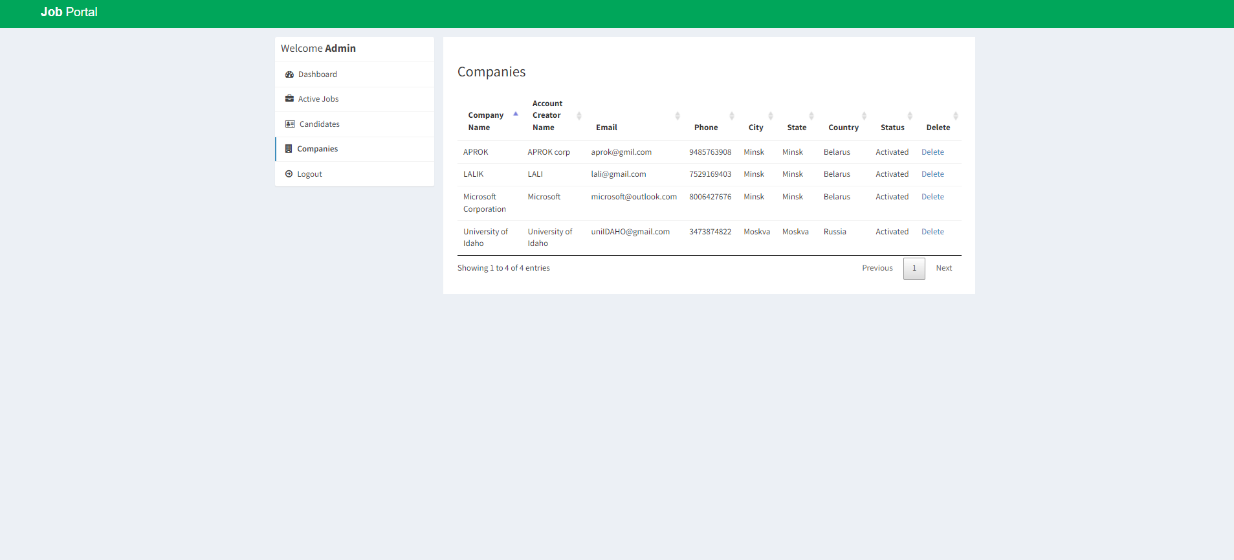


Рисунок 3.25 – Страница управления компаниями

ЗАКЛЮЧЕНИЕ

Основной целью проекта являлись проектирование и реализация системы с уникальным графическим интерфейсом в виде веб-приложения, для комфортного использования с устройств, имеющих выход в интернет. Одним из важных аспектов при создании системы является создание интуитивно понятного пользовательского интерфейса, который позволяет пользователям легко взаимодействовать с порталом для поиска работы.

В результате выполнения курсового проекта разработано пользовательское веб-приложения «Портал для поиска работы», предназначенное для поиска подходящих кандидатов на должности.

Для решения задачи проведен анализ существующих аналогов, описан функционал системы, доступный каждой группе пользователей и разработаны *UML*-диаграммы вариантов использования.

Разработанный сервис представляет собой клиент-серверное приложение. В системе есть четыре группы пользователей: гость, пользователь, компания и администратор, которые наделены различными правами. Пользователю, в зависимости от его роли, предоставляется определенный набор функций и элементов интерфейса, позволяющий осуществлять определенные действия в системе.

В данном курсовом проекте описаны основные этапы процесса разработки, включая проектирование интерфейса, определение функциональности, выбор технологий и инструментов, создание базы данных, а также реализация клиентской части с использованием *HTML, CSS, JavaScript*.

Результатом данного проекта стало создание веб-приложения, обеспечивающего удобный интерфейс для взаимодействия с разработанной базой данных. Применение современных веб-технологий и методов разработки обеспечивает высокую производительность, отзывчивость и удобство использования конечными пользователями.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] Орлов, С. А. Технологии разработки программного обеспечения: Разработка сложных программных систем: учебник / С. А. Орлов. – 3-е изд.– СПб.: Питер, 2004. – 527 с.

[2] Секреты *HTML CSS* [Электронный ресурс]. ̶ Режим доступа: [*https://skysmart.ru/articles/programming/secrety-HTML-CSS*](https://skysmart.ru/articles/programming/secrety-HTML-CSS)

[3] Рамбо, Дж. *UML* 2.0. Объектно-ориентированное моделирование и разработка / Дж. Рамбо, М. Блаха. – 2-е изд.– СПб.: Питер, 2007.— 544 с.

[4] Диго, С. М. Базы данных: проектирование и использование: учебник / С. М. Диго. – М.: Финансы и статистика, 2005.— 592 с.

[5] Ульман, Л. MySQL / Л. Ульман. – М.: ДМК Пресс; СПб.: Питер, 2004. – 352 с.

[6] Что такое нормализация данных? [Электронный ресурс]. ̶ Режим доступа: [*https://appmaster.io/ru/blog/chto-takoe-normalizatsiia-dannykh/*](https://appmaster.io/ru/blog/chto-takoe-normalizatsiia-dannykh/)

[7] Кузин, А. В. Базы данных: учеб. пособие / А. В. Кузин, С. В. Левонисова. – М.: Академия, 2005. – 320 с.

[8] СТП 01–2024. Стандарт предприятия. Дипломные проекты (работы). Общие требования. - Минск: БГУИР, 2024. – 178 с.

ПРИЛОЖЕНИЕ А  
(обязательное)  
Программный код функции приложения

*<?php*

*<?php*

*require 'vendor/autoload.php';*

*require\_once("db.php");*

*use PHPMailer\PHPMailer\PHPMailer;*

*use PHPMailer\PHPMailer\Exception;*

*if (isset($\_POST['email'])) {*

*$email = mysqli\_real\_escape\_string($conn, $\_POST['email']);*

*$temporaryPassword = generateTemporaryPassword();*

*$encryptedPassword = base64\_encode(strrev(md5($temporaryPassword)));*

*$sql = "UPDATE users SET password='$encryptedPassword' WHERE email='$email'";*

*if ($conn->query($sql) === true) {*

*$phpmailer = new PHPMailer();*

*try {*

*$phpmailer->isSMTP();*

*$phpmailer->Host = 'sandbox.smtp.mailtrap.io';*

*$phpmailer->SMTPAuth = true;*

*$phpmailer->Port = 2525;*

*$phpmailer->Username = '36600c33a8f396';*

*$phpmailer->Password = '5b74fef922554f';*

*$phpmailer->setFrom('vladislava2003love@gmail.com', 'Sender Name');*

*$phpmailer->addAddress('vlada@gmail.com','v');*

*$phpmailer->Subject = 'Password Recovery';*

*$phpmailer->Body = 'Your temporary password: ' . $temporaryPassword;*

*if ($phpmailer->send()) {*

*echo "Temporary password sent successfully.";*

*} else {*

*echo "Error sending email: " . $phpmailer->ErrorInfo;*

*}*

*} catch (Exception $e) {*

*echo 'Error sending email: ', $e->getMessage();*

*}*

*} else {*

*echo $conn->error;*

*}*

*}*

*header("Location: index.php");*

*$conn->close();*

*function generateTemporaryPassword() {*

*$length = 8;*

*$characters='0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';*

*$temporaryPassword = '';*

*for ($i = 0; $i < $length; $i++) {*

*$index = rand(0, strlen($characters) - 1);*

*$temporaryPassword .= $characters[$index];*

*}*

*return $temporaryPassword;*

*}*

*?>*