**Введение**

На курсовой проект поставлена задача разработки системы «Умный логистический трекинг».

Цель данного проекта – создание приложения, позволяющего пользователям управлять учетными записями и работать с данными об отгрузках. Приложение предоставляет базовые инструменты для учета и управления логистическими операциями, обеспечивая удобный интерфейс для работы с системой.

Далее приведено краткое описание разделов пояснительной записки.

Первый раздел «Анализ задачи» включает в себя исследование предметной области, определение её организационно-экономической сущности. В подразделе «Инструменты разработки» описана среда разработки, использованная для создания проекта, и установлены минимальные и оптимальные требования к аппаратным характеристикам для его правильного функционирования. Далее приведена диаграмма вариантов использования, отражающая взаимодействие пользователя с системой. Важным аспектом является модель данных, которая описывает структуру и взаимосвязи данных в системе, обеспечивая эффективное хранение и обработку информации.

Во втором разделе «Проектирование задачи» организуются данные и процессы. Приводится описание внешнего пользовательского интерфейса, а также разграничение прав доступа. Составляются тесты на использование.

Раздел «Реализация» описывает структуру программного продукта и описание функций пользователя и элементов управления, за которыми они закреплены.

Четвёртый раздел – «Тестирование». Данный раздел описывает тестирование программного продукта, список ошибок, выявленных при тестировании, а также содержит вывод о результатах тестирования.

В разделе «Применение» описано назначение, область применения и среда функционирования программного обеспечения. Включены перечень сообщений системы, а также краткое руководство по эксплуатации.

«Заключение» содержит краткую формулировку задачи, результаты проделанной работы, описание использованных методов и средств, а также оценку степени автоматизации процессов на различных этапах разработки.

В разделе «Список использованных источников» приведён список источников, использованных при разработке проекта.

В приложении к пояснительной записке представлены: диаграмма вариантов использования и листинг программы.

**1. Анализ предметной области и формулировка требований к программе**

**1.1 Описание предметной области**

Темой данного курсового проекта является «Умный логистический трекинг». Данный программный продукт представляет собой веб-приложение для управления учетными записями пользователей и обработки данных о грузоперевозках. Система предназначена для логистических компаний и владельцев бизнеса, которым необходимо контролировать процессы доставки и управлять данными о перевозках в удобном интерфейсе.

Основные функции логистических платформ включают учет отгрузок, управление заказами, отслеживание статусов доставки и анализ логистических данных. Система предоставляет пользователям возможность эффективно работать с данными о перевозках, управлять маршрутами и контролировать процесс логистики на всех этапах.

Программный продукт «Умный логистический трекинг» ориентирован на автоматизацию процессов учета отгрузок и городских доставок, что позволит сократить временные затраты и минимизировать ошибки при оформлении и обработке данных. Важно обеспечить удобный и понятный интерфейс, чтобы пользователи могли легко взаимодействовать с системой и находить нужную информацию.

Система позволяет пользователям регистрироваться, управлять учетной записью, редактировать данные профиля и удалять аккаунт. В приложении реализован функционал создания, редактирования и удаления записей об отгрузках. Аналогичные возможности предусмотрены для учета и управления городскими доставками. Встроенные инструменты поиска и фильтрации позволяют пользователям быстро находить необходимую информацию по отгрузкам и доставкам, используя параметры, такие как дата, статус или место назначения.

Программный продукт поддерживает работу с документацией, включая возможность выгружать транспортные накладные (ТТН) для учета перевозок. Дополнительно реализован функционал формирования диаграмм, позволяющий визуализировать данные о логистических процессах, что упрощает их анализ и способствует повышению эффективности работы.

Таким образом, «Умный логистический трекинг» представляет собой систему для упрощенного управления логистическими процессами, обеспечивающую удобную работу с учетными записями, отгрузками и аналитическими данными.

**1.2 Диаграмма вариантов использования**

**1.2 Диаграмма вариантов использования**

Диаграмма вариантов использования (Use Case Diagram) представляет взаимодействия между пользователями (акторами) и системой, демонстрируя основные функции, которые система выполняет.

Основные элементы диаграммы:

Актёры (Actors) – пользователи системы, взаимодействующие с её функциональностью

Администратор – единственный, кто имеет доступ ко всем функциям системы, включая управление пользователями, отслеживание отгрузок, формирование аналитики и контроль данных

Работник склада – передаёт информацию о поступлении и отправке товара, но не имеет доступа к системе

Водитель – выполняет доставку грузов, однако также не взаимодействует с системой напрямую

Графическое обозначение актёров представлено на Рисунке 1

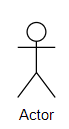


Рисунок 1 – Графическое обозначение актера

Варианты использования (Use Cases) – ключевые функции системы, доступные только администратору

Вход в систему – авторизация в системе

Управление пользователями – добавление, редактирование и удаление аккаунтов работников склада и водителей

Создание отгрузок – внесение данных о грузах, маршрутах и времени отправки

Редактирование и удаление отгрузок – изменение информации о доставке

Создание и управление городскими доставками – ведение учёта мелких перевозок внутри города

Поиск и фильтрация отгрузок – быстрый доступ к нужной информации

Формирование и выгрузка ТТН (транспортных накладных)

Создание аналитических диаграмм – визуализация статистики перевозок

Графическое обозначение варианта использования представлено на Рисунке 2



Рисунок 2 – Графическое обозначение варианта использования

Связи (Associations) – показывают, какие функции доступны администратору. Поскольку работник склада и водитель не взаимодействуют с системой напрямую, связи присутствуют только между администратором и функциональными возможностями системы

Графическое обозначение связи представлено на Рисунке 3

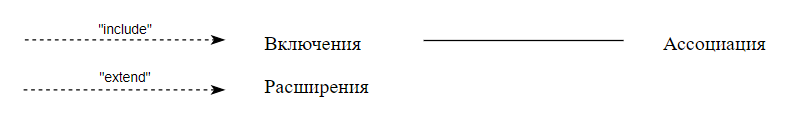


Рисунок 3 – Графическое обозначение связи

Границы системы (System Boundaries) – определяют, что входит в функциональность приложения. В систему включены управление аккаунтами, отслеживание отгрузок и аналитика перевозок, но не операционная деятельность складов и водителей

Диаграмма вариантов использования представлена в Приложении А

**1.3 Модель данных**

Цель моделирования данных состоит в определении концептуальной схемы совокупности данных в форме одной или нескольких локальных моделей, которые относительно легко могут быть отображены в любую форму представления данных.

Первый шаг моделирования – извлечение информации и выделение сущностей. Сущность – это реальный объект, имеющий существенное значение для рассматриваемой предметной области, информация о котором подлежит хранению в проекте.



Рисунок 4 – Графическое изображение сущности

Применимо к разрабатываемой задаче можно выделить следующие сущности:

Таблица 1 – Описание сущностей

|  |  |
| --- | --- |
| Сущность | Описание |
| Пользователь | Содержит информацию о пользователях системы, включая уникальный идентификатор, статус (например, обычный пользователь), личные данные (имя), логин, зашифрованный пароль, дату регистрации и путь к изображению аватара. |
| Отгрузка | Содержит информацию о логистических отгрузках, включая уникальный идентификатор отгрузки, название отгрузки, ID автора отгрузки (ссылается на пользователя), ID водителя (ссылается на сотрудника), пункты назначения и отправления, транспортное средство (номер автомобиля), и дату отгрузки. |
| Сотрудник | Содержит информацию о сотрудниках компании, включая уникальный идентификатор, имя, должность, дату приема на работу, логин, зашифрованный пароль, аватар и связку с пользователем через логин (может быть связан с пользователем через его статус или систему прав доступа). Также в таблице отображаются разные должности сотрудников, такие как водители, кладовщики, и другие роли. |

Следующим шагом в моделировании данных является идентификация связей. Связь – это ассоциация между двумя сущностями, значимая для рассматриваемой предметной области. Степень связи и обязательность графически изображаются так, как показано на рисунке 3.

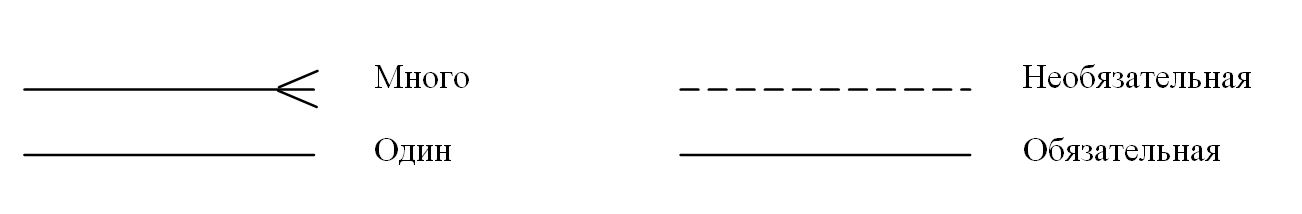


Рисунок 5 – Графическое изображение связи и обязательности

Логические зависимости (связи), которые прослеживаются между объектами нашей предметной области, будут как обязательными, так и необязательными. Причем, по характеру объединения данных установленные связи относятся к виду и «Один – ко – многим». Связи вида «Один – к – одному» и «Многие – ко – многим» в рассматриваемой модели не используется.

Ознакомиться со связями в нашей модели данных можно в таблице 2.

Таблица 2 – Сущности и их связи

|  |  |  |
| --- | --- | --- |
| Сущность | Связи | Сущность |
| Пользователь | Один пользователь может содержать несколько отгрузок | Отгрузка |
| Пользователь | Один пользователь может быть автором нескольких отгрузок | Отгрузка |
| Пользователь | Один пользователь может быть связан с несколькими сотрудниками (через отгрузки) | Сотрудник |
| Отгрузка | Одна отгрузка может иметь несколько сотрудников (водителей) | Сотрудник |
| Отгрузка | Одна отгрузка может содержать один или несколько водителей | Сотрудник |
| Сотрудник | Один сотрудник может быть связан с несколькими отгрузками | Отгрузка |

Продолжение таблицы (Таблица 2)

|  |  |  |
| --- | --- | --- |
| Сотрудник | Один сотрудник может иметь несколько позиций (например, водителя, кладовщика) | Позиция |
| Позиция | Одна позиция может принадлежать нескольким сотрудникам | Сотрудник |
| Отгрузка | Каждая отгрузка имеет пункт отправления и пункт назначения | Точка |
| Точка | Точка отправления может быть связана с несколькими отгрузками | Отгрузка |
| Точка | Точка назначения может быть связана с несколькими отгрузками | Отгрузка |

Последним шагом моделирования является идентификация атрибутов.

Атрибут – любая характеристика сущности, значимая для рассматриваемой предметной области и предназначенная для квалификации, идентификации, классификации, количественной характеристики или выражения состояния сущности. Атрибуты сущностей могут быть двух типов: обязательный (\*) и необязательный (°).

Таблица 3 – Описание атрибутов сущностей

|  |  |  |
| --- | --- | --- |
| Сущность | Атрибут | Описание атрибута |
| Пользователь | **id** | Уникальный идентификатор пользователя |
| Пользователь | **статус** | Статус пользователя (например, 'user') |
| Пользователь | **имя** | Имя пользователя |
| Пользователь | **логин** | Уникальный логин пользователя |
| Пользователь | **пароль** | Пароль для доступа к аккаунту пользователя |
| Пользователь | **дата** | Дата регистрации пользователя |
| Пользователь | **аватар** | URL изображения (аватара) пользователя |
| Отгрузка | **id** | Уникальный идентификатор отгрузки |
| Отгрузка | **название\_отгрузки** | Название отгрузки (например, ООО Минск) |
| Отгрузка | **автор\_отгрузки** | Идентификатор пользователя, создавшего отгрузку |
| Отгрузка | **id\_водителя** | Идентификатор водителя (сотрудника) |
| Отгрузка | **пункт\_назначения** | Пункт назначения отгрузки |
| Отгрузка | **пункт\_отправления** | Пункт отправления отгрузки |
| Транспортное средство | **Номер (Number)** | Номер транспортного средства |
| Транспортное средство | PK id ТС (Transport ID) | Уникальный идентификатор транспортного средства |

Продолжение таблицы (Таблица 3)

|  |  |  |
| --- | --- | --- |
| Сотрудник | **PK id** | Уникальный идентификатор сотрудника |
| Сотрудник | **Имя** | Имя сотрудника |
| Сотрудник | **id Должность (Position ID)** | Идентификатор должности сотрудника |
| Сотрудник | **Логин** | Уникальный логин сотрудника |
| Сотрудник | **Пароль** | Пароль для доступа к аккаунту сотрудника |
| Сотрудник | **PK id** | Уникальный идентификатор сотрудника |
| Должность | **Водитель (Driver)** | Должность "Водитель" |
| Должность | **Кладовщик (Storekeeper)** | Должность "Кладовщик" |
| Отгрузка | **PK id** | Уникальный идентификатор отгрузки |
| Отгрузка | **Перевозчик (Carrier)** | Название перевозчика |
| Отгрузка | **Водитель (Driver)** | Водитель отгрузки |
| Отгрузка | **Точка отправления (Departure Point)** | Место отправления |
| Отгрузка | **Точка назначения (Destination Point)** | Место назначения |
| Отгрузка | **id Транспортное средство (Transport ID)** | Идентификатор транспортного средства |
| Отгрузка | **Старший отгрузки (Senior Shipment)** | Ответственный за отгрузку |
| Отгрузка | **Дата (Date)** | Дата отгрузки |
| Отгрузка | **PK id** | Уникальный идентификатор отгрузки |

Продолжение таблицы (Таблица 3)

|  |  |  |
| --- | --- | --- |
| Админ | **PK id** | Уникальный идентификатор администратора |
| Админ | **Имя (Name)** | Имя администратора |
| Админ | **Логин (Login)** | Уникальный логин администратора |
| Админ | **Пароль (Password)** | Пароль для доступа к аккаунту администратора |
| Админ | **Фото (Photo)** | URL изображения администратора |
| Админ | **PK id** | Уникальный идентификатор администратора |

В итоге трех этапов моделирования получим модель данных, представленную в графической части проекта.

**1.4 Инструменты разработки**

Для разработки проекта использовались следующие инструменты и технологии:

-Flask – лёгкий и гибкий фреймворк для создания веб-приложений на языке Python. Flask отличается минималистичным подходом и предоставляет разработчикам гибкость в выборе компонентов, что делает его идеальным выбором для создания веб-приложений с простым и понятным API.

-PyCharm – среда разработки для Python, обеспечивающая мощные функции для написания, отладки и тестирования кода. PyCharm предоставляет все необходимые инструменты для работы с проектами на Flask и интеграцию с различными системами управления базами данных.

-PostgreSQL (PSQL) – система управления базами данных с открытым исходным кодом. PostgreSQL является мощной и стабильной СУБД, идеально подходящей для работы с приложениями, использующими сложные запросы и большие объемы данных. В проекте использовалась PostgreSQL для хранения данных и взаимодействия с веб-приложением.

Для разработки использовались также другие инструменты и программы:

-Google Chrome – браузер, который использовался для поиска информации, картинок и других ресурсов для теоретической части приложения, а также для тестирования работы веб-приложения в реальном времени.

-Microsoft Word 2010 – для написания пояснительных записок и создания HTML-страниц.

-Microsoft PowerPoint 2010 – для создания отчётных презентаций.

-Adobe Photoshop CC 2019 – для редактирования и создания изображений, которые использовались в проекте.

-Draw.io – для создания диаграмм и схем, таких как диаграммы вариантов использования, структурные схемы и другие графические элементы.

-Разработка велась на ноутбуке Lenovo с такими характеристиками:

-Процессор: Intel Core i5

-Оперативная память: 16 ГБ

-Жёсткий диск: 512 ГБ HDD

-Операционная система: Windows 10

-Минимальные системные требования для разработки:

-Процессор: Intel Core i5

-Оперативная память: 4 ГБ

-Жёсткий диск: 512 ГБ HDD

-Операционная система: Windows 10

Таким образом, для разработки и тестирования веб-приложения с использованием Flask и PostgreSQL не требуется высоких аппаратных ресурсов, что позволяет разрабатывать и работать с проектом даже на устройствах с минимальными требованиями.

**1.5 Требования к приложению**

Средства защиты:

-Аутентификация пользователей: Для повышения безопасности аккаунтов в системе реализована многофакторная аутентификация (например, с использованием одноразовых кодов, отправляемых на электронную почту или телефон). Это позволяет гарантировать, что только авторизованные пользователи могут получить доступ к функционалу приложения.

-Шифрование данных: Все передаваемые данные, такие как информация о транспорте, отгрузках и пользователях, защищены с использованием современных протоколов шифрования, таких как HTTPS. Это позволяет обеспечить безопасность информации на всех этапах ее передачи. Пароли и чувствительные данные также хранятся в зашифрованном виде.

-Контроль доступа: В зависимости от роли пользователя (например, водитель, работник склада, администратор) осуществляется ограничение доступа к различным функциям приложения. Например, администратор может управлять всеми данными и функциями системы, в то время как водитель и работник склада имеют доступ только к своим рабочим задачам, таким как отслеживание маршрутов и создание отгрузок.

Накладываемые ограничения:

-Ограничения по функционалу: Некоторые функции в системе могут быть ограничены в зависимости от роли пользователя. Например, водители и складские работники могут только просматривать свои текущие отгрузки и маршруты, но не могут изменять или удалять информацию, в то время как администраторы могут управлять всеми данными системы.

-Права пользователей: Для работы в системе необходимы определенные права, которые регулируются на основе ролей пользователей (например, администраторы могут редактировать данные, а водители только просматривать их). Это помогает обеспечивать безопасность и предотвращать несанкционированные изменения данных.

-Политика безопасности: Все пользователи обязаны следовать политикам безопасности, которые регулируют допустимость использования определенных функций системы. Например, только авторизованные сотрудники могут иметь доступ к данным о клиентах или конфиденциальной информации.

Требования к интерфейсу:

-Удобство и простота использования: Интерфейс системы должен быть интуитивно понятным и легким для восприятия. Пользователи, включая водителей и сотрудников склада, смогут быстро ориентироваться в функционале системы, благодаря понятной навигации и упрощенному взаимодействию с интерфейсом.

-Адаптивный дизайн: Система должна поддерживать корректное отображение на различных устройствах (ПК, планшеты, мобильные телефоны), с учетом особенностей экранов разных размеров. Это позволит пользователям работать с системой как в офисе, так и в дороге.

Доступность: Интерфейс должен быть доступен для людей с ограниченными возможностями. Например, для людей с нарушением зрения могут быть предусмотрены текстовые подсказки и возможность увеличения шрифта.

Эстетика и визуальная привлекательность: Дизайн системы должен быть современным, с использованием стандартов UI/UX для корпоративных приложений. Он должен быть не только функциональным, но и привлекательным, что способствует повышению пользовательского опыта и улучшает восприятие приложения в целом.

**2 Проектирование задачи**

**2.1 Организация данных**

В данной работе база данных (БД) организована в системе управления базами данных (СУБД) PostgreSQL. Она обеспечивает надежное хранение и управление данными, поддерживает сложные запросы и транзакции, а также обеспечивает высокую производительность и масштабируемость. PostgreSQL идеально подходит для работы с большими объемами данных и сложными структурами. В таблице 4 представлено описание таблиц БД.

Таблица 4 — Описание таблиц БД

|  |  |  |  |
| --- | --- | --- | --- |
| Таблица | Поле | Описание | Тип данных |
| User | id | Уникальный идентификатор пользователя | Integer (Primary Key) |
| User | status | Статус пользователя (например, "user") | String(10) |
| User | name | Имя пользователя | String(50) |
| User | login | Уникальный логин для входа | String(50) (Unique) |
| User | password | Пароль пользователя | String(150) |
| User | date | Дата и время регистрации пользователя | DateTime |

Продолжение таблицы (Таблица 4)

|  |  |  |  |
| --- | --- | --- | --- |
| User | avatar | URL изображения аватара пользователя | String(200) |
| Shipping | id | Уникальный идентификатор отгрузки | Integer (Primary Key) |
| Shipping | shipping\_name | Название отгрузки | String(30) |
| Shipping | author\_shipping\_name | Идентификатор пользователя, создавшего отгрузку | Integer (Foreign Key) |
| Shipping | driver\_id | Идентификатор водителя, связанный с отгрузкой | Integer (Foreign Key) |
| Shipping | destination\_point | Пункт назначения отгрузки | String(30) |
| Shipping | starting\_point | Пункт отправления отгрузки | String(30) |
| Shipping | vehicle | Тип транспортного средства | String(30) |
| Shipping | date | Дата и время отгрузки | DateTime |

Продолжение таблицы (Таблица 4)

|  |  |  |  |
| --- | --- | --- | --- |
| Employee | id | Уникальный идентификатор сотрудника | Integer (Primary Key) |
| Employee | name | Имя сотрудника | String(100) |
| Employee | position | Должность сотрудника | String(100) |
| Employee | date\_hired | Дата и время приема на работу сотрудника | DateTime |
| Employee | login | Уникальный логин для входа | String(50) (Unique) |
| Employee | password | Пароль для доступа к аккаунту сотрудника | String(200) |
| Employee | avatar | URL изображения аватара сотрудника | String(120) |

**2.2 Процессы**

В данной системе пользователи могут взаимодействовать с приложением через различные действия, такие как создание и управление профилями, создание и управление отгрузками, а также управление сотрудниками. В зависимости от роли пользователя (пользователь, администратор, сотрудник), система предоставляет разные функции для выполнения.

Варианты использования:

Регистрация пользователя: Пользователь создает новый аккаунт.

Реализация: Маршрут user/register создает нового пользователя с помощью формы регистрации.

Вход в систему: Пользователь аутентифицируется в системе.

Реализация: Маршрут user/login проверяет данные пользователя и осуществляет вход.

Просмотр профиля: Пользователь просматривает свой профиль или профиль другого пользователя.

Реализация: Маршрут user/profile позволяет пользователю просматривать свои данные, а user/all\_profile — просматривать данные других пользователей.

Редактирование профиля: Пользователь изменяет свои личные данные.

Реализация: Маршрут user/update\_user\_profile позволяет пользователю редактировать свои данные.

Удаление пользователя: Пользователь или администратор может удалить свой аккаунт или аккаунт другого пользователя.

Реализация: Маршрут user/<int:id>/delete\_user удаляет пользователя.

Поиск пользователей: Пользователи могут искать других пользователей по имени или логину.

Реализация: Маршрут user/search позволяет искать пользователей по введенному запросу.

Просмотр всех отгрузок: Пользователь может просматривать все отгрузки, отсортированные по дате.

Реализация: Маршрут shipping.all выводит список всех отгрузок.

Создание отгрузки: Пользователь может создать новую отгрузку, указав все необходимые данные.

Реализация: Маршрут shipping/create позволяет создать новую отгрузку.

Редактирование отгрузки: Пользователь может редактировать информацию о своей отгрузке.

Реализация: Маршрут shipping/<int:id>/update позволяет редактировать информацию о выбранной отгрузке.

Удаление отгрузки: Пользователь может удалить свою отгрузку.

Реализация: Маршрут shipping/<int:id>/delete удаляет выбранную отгрузку.

Фильтрация отгрузок: Пользователи могут фильтровать отгрузки по различным критериям, таким как дата или водитель.

Реализация: Маршрут shipping/filter применяет фильтры к отгрузкам.

Поиск отгрузок: Пользователи могут искать отгрузки по различным параметрам.

Реализация: Маршрут shipping/search позволяет искать отгрузки по нескольким параметрам (например, название отгрузки, водитель и т.д.).

Управление сотрудниками: Администратор может управлять учетными записями сотрудников.

Реализация: Маршруты, такие как employee/register\_employee, employee/update\_employee\_profile, и employee/delete\_employee, позволяют добавлять, редактировать и удалять сотрудников.

Регистрация сотрудника: Администратор регистрирует новых сотрудников, назначая их должности.

Реализация: Маршрут employee/register\_employee позволяет создавать новый профиль сотрудника.

Редактирование профиля сотрудника: Администратор или сам сотрудник может изменить данные профиля.

Реализация: Маршрут employee/update\_employee\_profile позволяет редактировать данные сотрудника.

Удаление сотрудника: Администратор может удалить учетную запись сотрудника.

Реализация: Маршрут employee/<int:id>/delete удаляет сотрудника.

**2.3 Описание внешнего пользовательского интерфейса**

Пользовательский интерфейс будет ориентирован на минимализм и простоту. Главные элементы дизайна включают:

Легкость в навигации.

Структура с четким разделением по ролям (пользователь, администратор).

Формы и таблицы для ввода и отображения данных о пользователях, отгрузках и сотрудниках.

Удобные всплывающие сообщения (flash-сообщения) для уведомлений о результате действия.

**2.4 Средства защиты и разграничения доступа к данным**

Средства защиты включают в себя аутентификацию с помощью Flask-Login и шифрование паролей с помощью bcrypt. Все формы требуют валидации данных, а доступ к чувствительным разделам (например, редактирование или удаление данных) ограничен правами пользователя или администратора.

Пользователь: Может взаимодействовать с основными функциями системы (регистрация, вход, создание/редактирование профиля).

Администратор: Имеет расширенные права для управления отгрузками и сотрудниками.

Сотрудник: Может редактировать свой профиль и работать с отгрузками.

**2.5 Тесты на использование**

Процесс тестирования будет включать проверку всех функциональных элементов, таких как регистрация, аутентификация, создание и редактирование данных. После реализации функций будут проведены тесты на функциональность (например, создание отгрузки, регистрация сотрудника). Тестирование будет гарантировать правильную работу всех функций, а также выполнение бизнес-логики и авторизации.

**3 Реализация**

Логическая организация определяет структуру данных и их взаимосвязи, тогда как физическая организация описывает, как данные хранятся на носителях. Оба аспекта важны для эффективного управления информацией. Описание методов представлено в таблице 6.

Таблица 6 — Описание методов

|  |  |  |  |
| --- | --- | --- | --- |
| Функция | Элемент управления | Модель | Метод |
| Авторизация | <%= f.submit "Войти" %> | User | after\_sign\_in\_path\_for |
| Регистрация | <%= f.submit "Зарегистрироваться" %> | User | after\_sign\_up\_path\_for |
| Редактировать профиль | <%= link\_to 'Редактировать профиль', edit\_page\_path(current\_user) %> | User | update |
| Создание отгрузки | <%= link\_to 'Создать отгрузку', new\_shipping\_path(current\_user.id) %> | Shipping | create |
| Редактирование отгрузки | <%= link\_to 'Редактировать отгрузку', edit\_shipping\_path(shipping.id) %> | Shipping | update |
| Удаление отгрузки | <%= button\_to 'Удалить отгрузку', shipping, method: :delete, data: { confirm: 'Вы уверены, что хотите удалить эту отгрузку?' } %> | Shipping | destroy |
| Создание сотрудника | <%= link\_to 'Создать сотрудника', new\_employee\_path(current\_user.id) %> | Employee | create |
| Редактирование сотрудника | <%= link\_to 'Редактировать сотрудника', edit\_employee\_path(employee.id) %> | Employee | update |
| Удаление сотрудника | <%= button\_to 'Удалить сотрудника', employee, method: :delete, data: { confirm: 'Вы уверены, что хотите удалить этого сотрудника?' } %> | Employee | destroy |
| Просмотр списка сотрудников | <%= link\_to 'Просмотреть сотрудников', employees\_path %> | Employee | index |
| Создание пользователя | <%= link\_to 'Создать пользователя', new\_user\_path %> | User | create |
| Редактирование пользователя | <%= link\_to 'Редактировать пользователя', edit\_user\_path(user.id) %> | User | update |

Продолжение таблицы (Таблица 6)

|  |  |  |  |
| --- | --- | --- | --- |
| Удаление пользователя | <%= button\_to 'Удалить пользователя', user, method: :delete, data: { confirm: 'Вы уверены, что хотите удалить этого пользователя?' } %> | User | destroy |
| Фильтрация отгрузок | <%= submit\_tag 'Фильтровать отгрузки' %> | Shipping | filter |
| Сортировка отгрузок | <%= submit\_tag 'Сортировать отгрузки' %> | Shipping | sort |
| Подписание заявки на отгрузку | <%= button\_to 'Подписаться на отгрузку', shipping\_subscribe\_path(shipping.id), method: :post %> | Shipping | subscribe |
| Отписка от отгрузки | <%= button\_to 'Отписаться от отгрузки', shipping\_unsubscribe\_path(shipping.id), method: :delete, data: { confirm: 'Вы уверены, что хотите отписаться?' } %> | Shipping | unsubscribe |
| Отправить заявку на создание отгрузки | <%= button\_to 'Отправить заявку на создание отгрузки', request\_shipping\_path %> | ShippingRequest | create |
| Отмена заявки на отгрузку | <%= button\_to 'Отменить заявку', cancel\_shipping\_request\_path(request), method: :delete %> | ShippingRequest | destroy |
| Просмотр профиля пользователя | <%= link\_to 'Посмотреть профиль', user\_profile\_path(current\_user) %> | User | show |

4. Тестирование

Тестирование является одним из ключевых этапов разработки программного продукта. Проведение тестов позволяет выявить ошибки и недочёты в работе системы, однако даже при тщательном тестировании нельзя гарантировать полное отсутствие ошибок в реальных условиях эксплуатации.

Основное внимание при тестировании было уделено корректности отображения страниц, работоспособности кнопок, выпадающих списков и других элементов интерфейса. Большая часть ошибок была обнаружена и устранена на этапе реализации проекта.

После завершения разработки проведено комплексное тестирование программного продукта. Тестирование выполнялось разработчиком — Лапутько Кириллом Сергеевичем. Оно проводилось на персональном компьютере с операционной системой Windows 10 с использованием локального сервера.

Расписание и продолжительность тестирования приведены в таблице 7.

Таблица 7 — Расписание проведения тестов

| ФИО | Дата | Описание | Длительность, ч |
| --- | --- | --- | --- |
| Лапутько К. С. | 01.03.2025 | Выполнение тест-кейсов | 1.5 |
| Лапутько К. С. | 02.03.2025 | Анализ результатов тестирования | 1.0 |
| Лапутько К. С. | 05.03.2025 | Повторное выполнение тест-кейсов | 1.5 |

После тестирования была проведена оценка количества и критичности выявленных ошибок. Результаты приведены в таблице 8.

Таблица 8 – Статистика по ошибкам

| Статус | Количество ошибок | Важность ошибки |
| --- | --- | --- |
|  | Критическая | Высокая |
| Проверено | 4 | 1 |
| Исправлено | 4 | 1 |

В ходе тестирования была выявлена ошибка при обновлении данных, а также некорректное отображение некоторых элементов интерфейса. Все найденные ошибки были исправлены.

На основании проведённого тестирования можно сделать вывод о стабильности работы программного продукта и его готовности к использованию.

**5. Применение**

**5.1 Назначение**

Цель данного проекта заключается в разработке веб-приложения, которое позволит пользователям эффективно управлять процессами, связанными с отгрузками товаров, сотрудниками и учетными записями пользователей. Платформа предназначена для предприятий, занимающихся логистикой, складским учетом и управлением персоналом.

Система предоставляет удобные инструменты для создания, редактирования и отслеживания отгрузок, а также управления сотрудниками и пользователями. Это позволит автоматизировать процессы, повысить прозрачность учета и улучшить контроль за операциями.

Пользователи смогут просматривать список доступных отгрузок, фильтровать их по параметрам, подписываться на уведомления, а также управлять сотрудниками, изменяя их данные или удаляя их из системы. Благодаря интуитивно понятному интерфейсу, работа с системой будет удобной и эффективной.

**5.2 Среда функционирования**

Система поддерживает работу на различных операционных системах, включая Windows, macOS, Linux, а также мобильные платформы Android и iOS.

Для работы с веб-приложением необходимо стабильное подключение к интернету и наличие установленного веб-браузера. Поддерживаются браузеры: Google Chrome, Mozilla Firefox, Safari, Microsoft Edge.

**5.3 Инсталляция**

Приложение запускается на локальном сервере и требует установки всех необходимых зависимостей.

**5.4 Сообщения системы**

Приложение выводит пользователям различные уведомления, упрощающие навигацию и работу с системой.

Сообщения при авторизации и регистрации

"Вы успешно вошли в систему!" – подтверждение успешного входа.

"Неверный логин или пароль." – ошибка при вводе учетных данных.

"Вы успешно зарегистрировались!" – подтверждение успешной регистрации.

"Пожалуйста, подтвердите свой адрес электронной почты." – запрос на подтверждение e-mail.

"Вы вышли из системы." – уведомление о выходе.

Сообщения при управлении отгрузками

"Отгрузка успешно создана!" – подтверждение создания отгрузки.

"Отгрузка успешно обновлена!" – подтверждение редактирования.

"Отгрузка успешно удалена!" – подтверждение удаления.

"Ошибка при создании отгрузки. Проверьте введенные данные." – ошибка при вводе данных.

"Вы подписались на уведомления об отгрузке." – подтверждение подписки.

"Вы отписались от уведомлений об отгрузке." – подтверждение отписки.

Сообщения при управлении сотрудниками

"Сотрудник успешно добавлен!" – подтверждение добавления нового сотрудника.

"Сотрудник успешно обновлён!" – подтверждение редактирования данных.

"Сотрудник успешно удалён!" – подтверждение удаления.

"Ошибка при обновлении сотрудника. Проверьте введенные данные." – ошибка при редактировании.

Сообщения при управлении пользователями

"Пользователь успешно добавлен!" – подтверждение регистрации нового пользователя.

"Пользователь успешно обновлён!" – подтверждение изменения данных.

"Пользователь успешно удалён!" – подтверждение удаления.

"Ошибка при изменении данных пользователя. Попробуйте снова." – ошибка при обновлении данных.

"Роль пользователя успешно изменена!" – подтверждение изменения прав доступа.

"Ошибка при изменении роли. Проверьте данные." – ошибка при изменении прав.

Сообщения при сбросе пароля

"Сброс пароля завершен!" – подтверждение успешного сброса.

"Пожалуйста, проверьте вашу почту для инструкций по сбросу пароля." – система отправила инструкции.

"Ошибка при сбросе пароля. Попробуйте снова." – ошибка в процессе сброса.

Сообщения для администратора

"Запись успешно создана!" – подтверждение добавления нового контента.

"Запись успешно обновлена!" – подтверждение редактирования контента.

"Запись успешно удалена!" – подтверждение удаления записи.

"Ошибка при создании записи. Проверьте данные." – ошибка при добавлении контента.

**5.5 Обучение**

При открытии сайта попадаем на домашнюю страницу, представленную на рисунке 5.

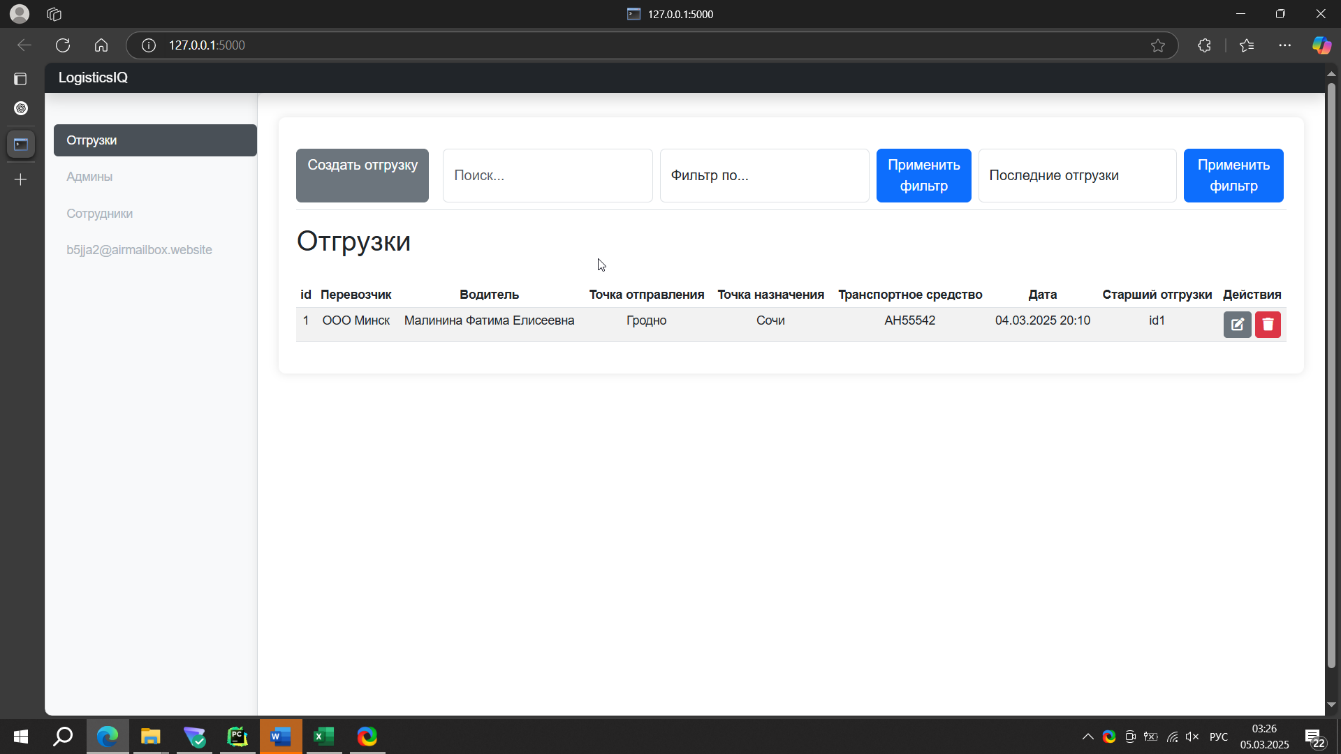


Рисунок 5 — Домашняя страница

На странице профиля, представленной на рисунке 6, можно просмотреть профиль.

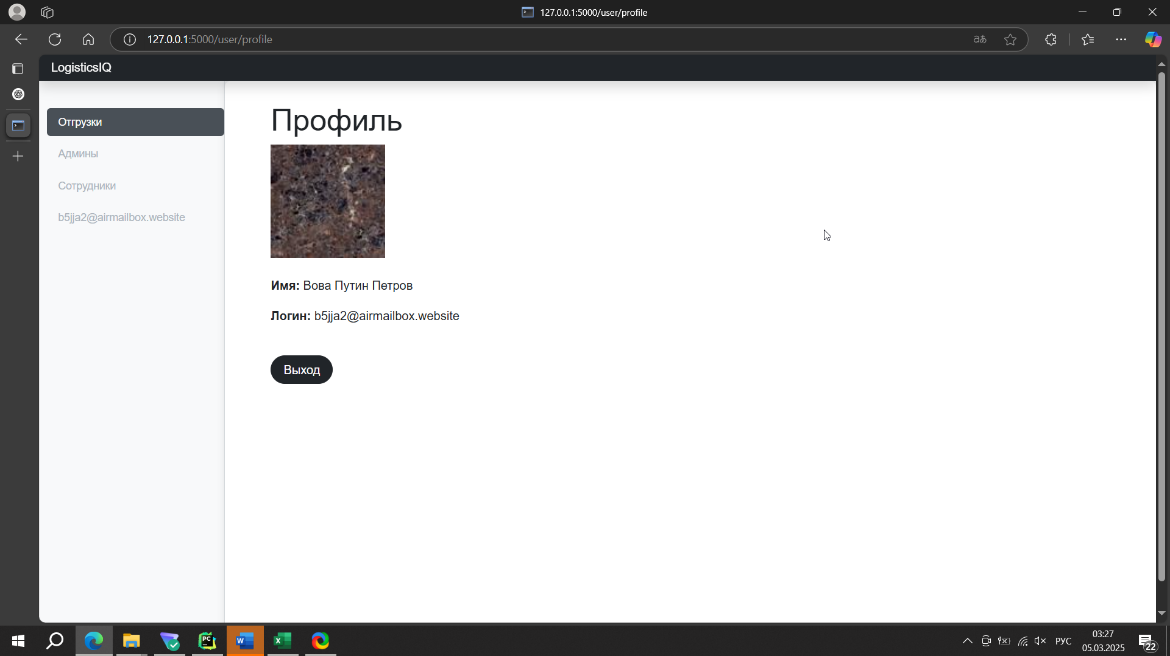


Рисунок 6 —Страница профиля пользователя

На рисунке 7 представлена страница, где можно просмотреть сотрудников.

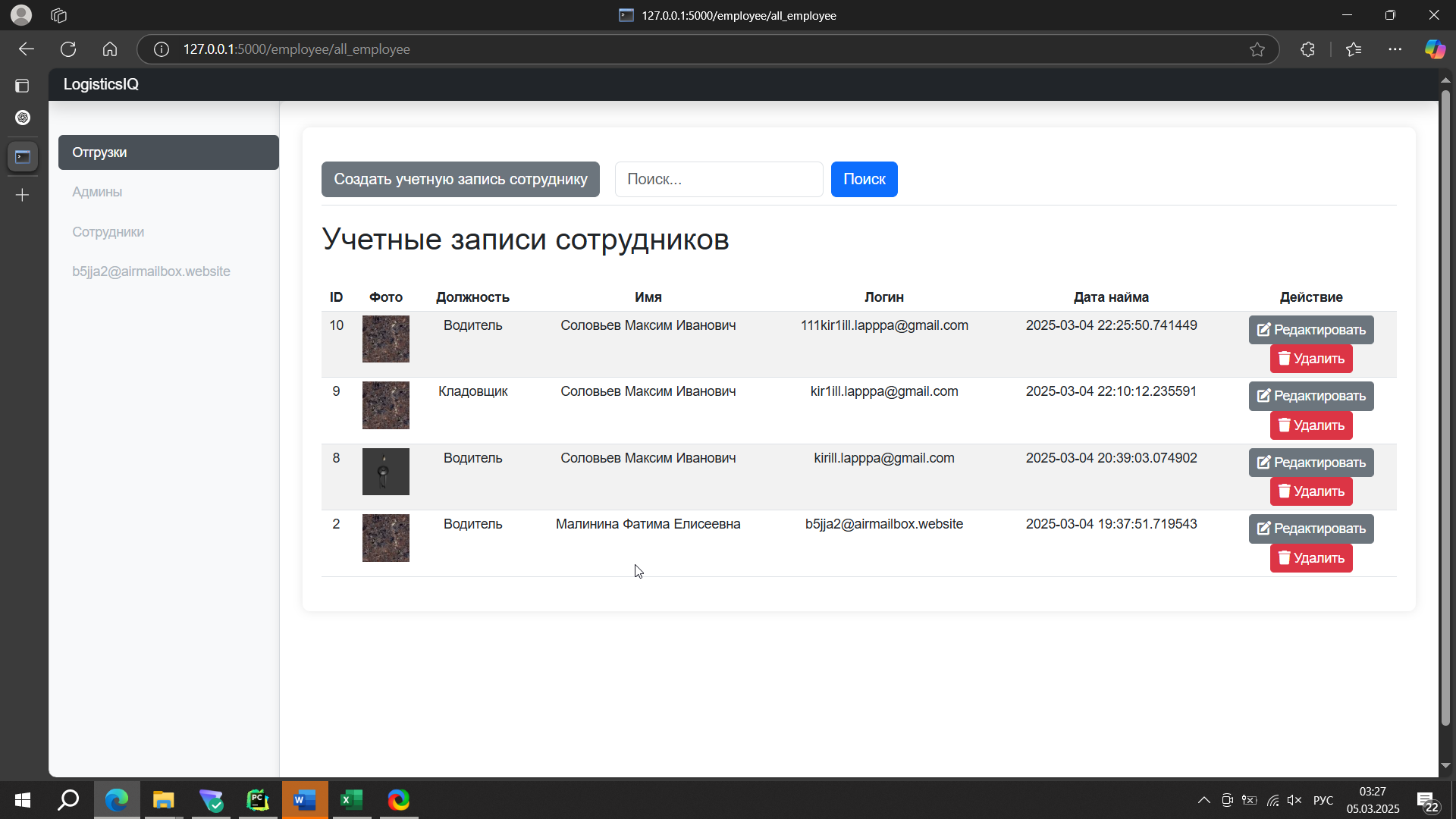


Рисунок 7 —Страница просмотра сотрудников

На рисунке 8 представлена страница просмотра админов

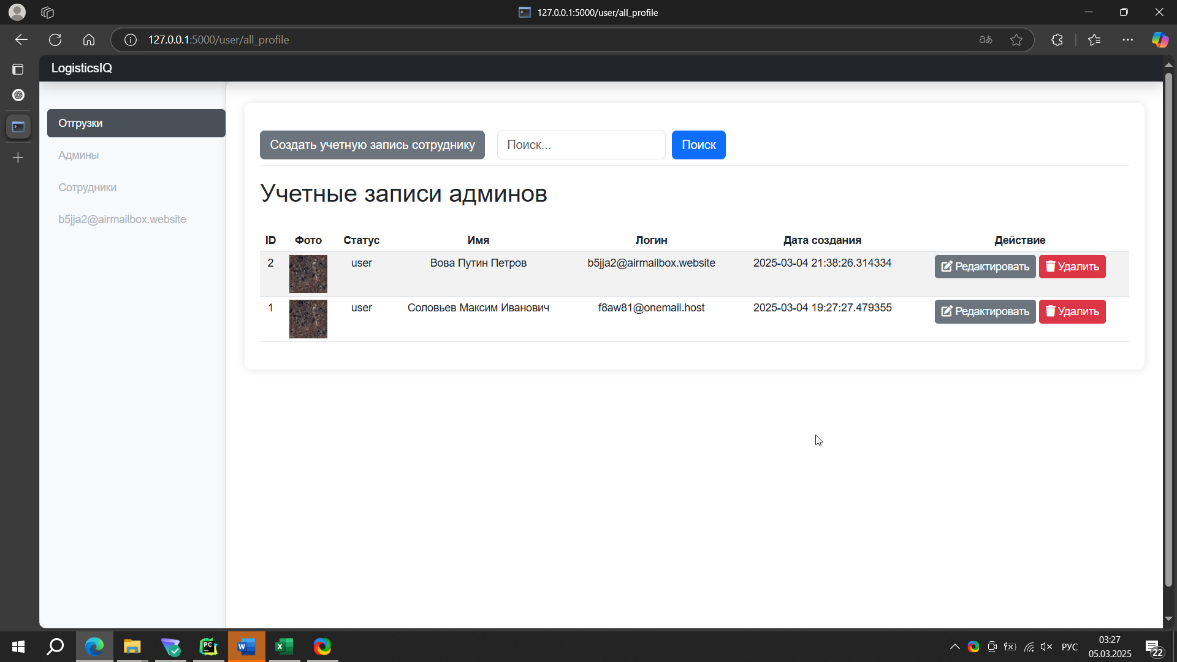


Рисунок 8 —Страница просмотра админов

**Заключение**

Целью проекта «Умный логистический трекинг» было создание системы для управления учетными записями и обработки данных об отгрузках, которая обеспечит пользователям удобный инструмент для эффективного учета и мониторинга логистических операций. В ходе разработки системы была реализована функциональность, охватывающая ключевые аспекты логистики, включая управление учетными записями, обработку и отслеживание данных об отгрузках, а также анализ и формирование отчетов.

Приложение успешно прошло тестирование, продемонстрировав свою стабильность и эффективность в различных сценариях использования. Разработанный интерфейс отличается интуитивной понятностью, что значительно упрощает работу с системой и позволяет пользователям сосредоточиться на выполнении своих задач, а не на технических аспектах работы программы.

В дальнейшем возможно расширение функционала приложения, включая интеграцию с внешними сервисами, улучшение аналитических инструментов и добавление новых возможностей для оптимизации логистических процессов. Использование современных технологий и подходов в разработке позволило создать надежную и масштабируемую систему, отвечающую всем требованиям пользователей.

Проект «Умный логистический трекинг» представляет собой полнофункциональное решение, которое значительно улучшает процессы управления логистикой, повышая эффективность работы компаний и обеспечивая высокий уровень удобства и надежности для конечных пользователей.

**Список использованных источников**

Список использованных источников:

Документация Flask [Электронный ресурс]. – Электронные данные. – Режим доступа: https://flask.palletsprojects.com/ – Дата доступа: 05.03.2025.

Flask Web Development: Developing Web Applications with Python / Miguel Grinberg. – 2-е издание. – O'Reilly Media, 2018. – 480 с.

Flask by Example [Электронный ресурс]. – Электронные данные. – Режим доступа: https://www.realpython.com/flask-by-example-part-1-project-setup/ – Дата доступа: 05.03.2025.

Flask – A Microframework for Python [Электронный ресурс]. – Электронные данные. – Режим доступа: https://www.tutorialspoint.com/flask/index.htm