

Министерство науки и высшего образования Российской Федерации
Муромский институт (филиал)
Федерального государственного бюджетного образовательного учреждения высшего
образования
«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»

Факультет _____ ИТР _____

Кафедра _____ ПИН _____

КУРСОВАЯ РАБОТА

По _____ Разработка корпоративных приложений _____

Тема _____ Веб-приложение «Доставка продуктов питания» _____

Руководитель

Кульков Я.Ю.

(фамилия, инициалы)

(подпись)

(дата)

Студент _____ ПИН - 121 _____

(группа)

Скотников Е.С.

(фамилия, инициалы)

(подпись)

(дата)

Муром 2024

В данной курсовой работе необходимо было спроектировать веб-приложение для сервиса доставки продуктов сервиса. В качестве средств разработки приложения был фреймворк Spring. Язык разработки: Java

In this course work, it was necessary to design a web application for a product delivery service. The Spring framework was used as an application development tool. Development language: Java

Содержание

Введение.....	6
1.Анализ технического задания.....	7
2.Разработка моделей данных.....	8
2.1. Концептуальная модель данных.....	8
2.2. Логическая модель данных	9
2.3. Физическая модель данных	10
3.Проектирование работы системы.....	11
4.Разработка и реализация системы	13
5.Тестирование системы.....	15
Заключение	17
Список литературы:	18
Приложение	19
Приложение 1. Модели данных.....	19
Приложение 2. Текст программы	21
Приложение 3. Снимки окон программы (скриншоты программы)	22

					МИВУ 09.03.04 - 0.016 ПЗ					
Изм.	Лист	№ докум.	Подпись	Дата	Веб-приложение «Доставка продуктов питания»			Лит.	Лист	Листов
Разраб.		Скотников Е.С.								
Провер.		Кульков Я.Ю.							4	27
Реценз.								МИ ВлГУ ПИН-121		
Н. Контр.										
Утверд.										

Введение

Современные технологии значительно упрощают процессы управления предприятиями, особенно в сфере доставки продуктов питания, где важна оперативность, точность и прозрачность. В связи с этим автоматизация таких процессов становится неотъемлемой частью эффективной работы компаний, занимающихся доставкой.

Данная работа посвящена разработке веб-приложения «Доставка продуктов питания», которое предназначено для автоматизации ключевых процессов предприятия. Основная цель разработки заключается в создании функционального и интуитивно понятного решения для администраторов и клиентов.

Приложение предоставляет возможность администраторам управлять каталогом продуктов с использованием CRUD-методов, формировать отчеты по продажам. Для клиентов система обеспечивает регистрацию, просмотр каталога, добавление товаров в корзину, оформление покупок и просмотр истории заказов.

Разработка ведется с использованием фреймворка Spring, что обеспечивает надежность, масштабируемость и гибкость системы. Программный продукт поможет повысить эффективность работы предприятия, улучшить клиентский опыт и упростить управление бизнес-процессами.

					МИВУ 09.03.04 – 0.016 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		6

1. Анализ технического задания

Основная цель заключается в создании функционального и удобного приложения, которое позволит администратору эффективно управлять продуктами, заказами, а также генерировать отчеты. Для клиентов предусмотрены возможности регистрации, просмотра каталога товаров, добавления их в корзину, оформления заказов и просмотра истории покупок, что обеспечивает удобство использования и повышает удовлетворенность пользователей.

Дополнительные требования включают авторизацию с различными ролями пользователей, адаптивный интерфейс, валидацию данных на клиентской и серверной сторонах, формирование отчетов в виде файлов и отправку уведомлений по электронной почте. Адаптивный интерфейс гарантирует корректное отображение приложения на различных устройствах, что важно для охвата широкой аудитории пользователей.

Особое внимание уделено системе учета остатков продуктов, что позволяет поддерживать актуальную информацию о наличии товаров и предотвращать ситуацию с отсутствием необходимых продуктов.

					МИВУ 09.03.04 – 0.016 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		7

2. Разработка моделей данных

2.1. Концептуальная модель данных

Концептуальная модель хранилища данных представляет собой описание главных (основных) сущностей и отношений между ними. Концептуальная модель является отражением предметных областей, в рамках которых планируется построение хранилища данных. При проектировании концептуальной модели структурируют данные и выявляют взаимосвязи между ними, без рассмотрения особенностей реализации и вопросов эффективности обработки. Поэтому концептуальная модель не является полностью подходящей для дальнейшей разработки; все таблицы должны быть нормализованы для реляционной базы данных. Составленная концептуальная модель представлена на рисунке 1 приложения 1.

					МИВУ 09.03.04 – 0.016 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		8

2.2. Логическая модель данных

Логическая модель расширяет концептуальную путем определения для сущностей их атрибутов, описаний и ограничений, уточняет состав сущностей и взаимосвязи между ними. Концептуальная модель изменяется так, чтобы она могла быть обеспечена конкретной моделью данных. В результате формируется логическая модель. Логическая модель отражает логические связи между элементами данных вне зависимости от их содержания и среды хранения. Логическая модель может быть реляционной, иерархической или сетевой.

В качестве способа организации информационной базы выбрана реляционная база данных. Именно такой способ хранения всех данных является наиболее подходящим для проектируемой информационной системы по следующим причинам:

Наглядность модели для пользователя: Все данные в реляционной модели представлены в табличной форме, что делает их легко читаемыми и понятными.

Независимость данных от программного продукта для их обработки: Реляционные базы данных позволяют изменять структуру данных без необходимости изменения приложений, которые с ними работают.

Реляционные базы данных являются наиболее распространенными среди разработчиков ПО: следовательно, использование этих баз позволит сэкономить время и бюджет на внедрение нового типа БД.

Основные сущности логической модели и их атрибуты:

CartItem: id, price, quantity, product_id, user_id.

OrderItem: id, price, quantity, order_table_id, product_history_id.

OrderTable: id, status, city, created_at, home, street, total_price, user_id

Product: id, description, image_data, name, price, quantity.

ProductHistory: id, description, image_data, name, price, quantity.

Users: id, email, role, username, password, phone_number.

					МИВУ 09.03.04 – 0.016 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		9

Составленная логическая модель представлена на рисунке 2 приложения 1.

2.3. Физическая модель данных

Физические модели данных служат для отображения моделей данных. Основными понятиями модели данных являются поле, логическая запись, логический файл. Слово "логический" введено, чтобы отличать понятия, относящиеся к логической модели данных, от понятий, относящихся к физической модели данных. Основными понятиями физической модели данных, используемыми для представления логической модели данных, являются поле, физическая запись, физический файл. В частности, логическая запись, состоящая из полей, может быть представлена в виде физической записи (из тех же полей), логический файл – в виде физического файла. Имена таблиц и колонок будут сгенерированы на основе сущностей и атрибутов логической модели, учитывая максимальную длину имени и другие синтаксические ограничения, накладываемые СУБД. Если в имени сущности или атрибута встречается пробел, он заменяется на символ «_».

Физическая модель описывает способ хранения данных в базе данных. В физической модели мы учитываем типы данных, индексы, ограничения целостности и другие технические детали.

					МИВУ 09.03.04 – 0.016 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		10

3. Проектирование работы системы

В данном разделе содержится информация о проектировании работы информационной системы для автоматизации работы сервиса доставки продуктов, разработанной в текущей курсовой работе.

В первую очередь было необходимо определить основные компоненты системы и их взаимодействие. При проектировании работы системы автовокзала нужно уделить особое внимание архитектуре приложения и его функциональности.

В соответствии с техническим заданием, система должна быть создана с использованием технологии Spring Framework, а в качестве СУБД выбрана PostgreSQL.

Основные компоненты системы включают клиентскую и серверную части, которые в данном случае объединены в одном проекте на Spring.

Клиентская часть состоит из формы регистрации пользователей, формы для клиентов, которая обеспечивает возможность просмотра каталога, сбора корзины, заказов и просмотра истории покупок, а также формы для администраторов, предоставляющей интерфейс для управления данными о продуктах, пользователях и статусах заказов.

Серверная часть включает контроллеры, которые обрабатывают запросы от клиентской части, выполняют CRUD-операции и генерируют отчеты, сервисы, реализующие бизнес-логику приложения, такую как управление данными и генерация отчетов, а также репозитории, которые взаимодействуют с базой данных для выполнения операций чтения и записи данных.

Первоначальным этапом проектирования является разработка моделей данных. Для учета клиентов, продуктов, заказов и корзины необходимо создать соответствующие сущности и связи между ними.

					МИВУ 09.03.04 – 0.016 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		11

Далее следует создание CRUD-операций для администратора и пользователя. Это позволит администратору управлять данными о товарах, пользователях и заказах, а клиенту — просматривать и собирать заказы.

В рамках создания CRUD-функционала важно предусмотреть возможности добавления, просмотра, обновления и удаления записей.

Для обеспечения безопасности данных пользователей необходимо реализовать механизм аутентификации и авторизации. Для реализации данного этапа можно использовать компонент Spring Security для разграничения доступа к функционалу приложения.

Пользователи будут иметь разные роли (администратор и пользователь), что определяет их доступ к функционалу системы. Система должна предоставлять возможность генерации сводных отчетов на основе данных из базы данных. Эти отчеты могут включать информацию о всех заказах.

Интерфейс приложения должен быть понятным и удобным для пользователей. Для этого рекомендуется использовать современные принципы дизайна пользовательского интерфейса и обеспечить его дружелюбность и удобство использования.

Система будет использовать базу данных для хранения всех необходимых данных. База данных должна содержать таблицы для учета корзины, заказов, пользователей и продуктов. Серверная часть будет обрабатывать запросы от клиентской части, выполнять CRUD-операции с базой данных и генерировать отчеты.

Проектирование работы веб-приложения доставки продуктов включает в себя проектирование базы данных, разработку пользовательского интерфейса, создание CRUD-операций для администратора и пользователя. Важно уделить внимание архитектуре приложения, безопасности данных и удобству использования для конечных пользователей.

4. Разработка и реализация системы

Разработка и реализация системы для автоматизации работы сервиса доставки включает несколько ключевых этапов, каждый из которых играет важную роль в создании функционального и надежного приложения. В данном разделе будет описан процесс разработки и реализации клиентской и серверной частей приложения, а также интеграция различных компонентов системы.

Разработка моделей данных. Первым шагом в разработке системы является создание моделей данных. Модели данных определяют структуру базы данных и взаимосвязи между различными сущностями. В нашем случае основные сущности включают пользователей (User), элементы корзины (CartItem), элементы заказа (OrderItem), продукты (Product), историю продуктов (ProductHistory) и заказы (OrderTable). Для каждой сущности были определены атрибуты, ограничения и связи с другими сущностями.

Проектирование архитектуры системы.

Архитектура системы была разработана с использованием фреймворка Spring, что обеспечивает модульность и масштабируемость приложения. Основные компоненты архитектуры включают:

Контроллеры (Controllers): Обработывают запросы от пользователей, взаимодействуют с сервисами для выполнения бизнес-логики и возвращают ответы клиенту.

Сервисы (Services): Реализуют бизнес-логику приложения, такую как управление данными, генерация отчетов и отправка уведомлений.

Репозитории (Repositories): Взаимодействуют с базой данных для выполнения операций чтения и записи данных.

Сущности (Pojo): Представляют модели данных, которые отображаются на таблицы в базе данных.

Реализация CRUD-операций. Для обеспечения полного функционала системы были реализованы CRUD-операции для всех основных сущностей. Это позволяет администраторам управлять данными о продуктах, заказах и пользователях, а клиентам — просматривать и собирать заказы. Например, для сущности OrderTable были реализованы методы для создания нового заказа, чтения информации о законченных заказах.

Реализация безопасности. Для обеспечения безопасности данных и защиты от несанкционированного доступа была реализована система аутентификации и авторизации с использованием Spring Security. Пользователи системы имеют разные роли (администратор и пользователь), что определяет их доступ к функционалу системы. Администраторы имеют доступ к управленческим функциям системы, в то время как клиенты имеют ограниченный доступ только к тем функциям, которые необходимы для сборки и просмотров собственных заказов.

Генерация отчетов.

Система предоставляет возможность генерации сводных отчетов на основе данных из базы данных. Эти отчеты могут включать информацию о всех заказах пользователей. Для генерации отчетов были реализованы специальные сервисы, которые извлекают данные из базы данных и формируют отчеты в удобном для пользователя формате.

5. Тестирование системы

Основная цель проведения тестирования состоит в проверке соответствия реализации системы требуемой функциональности. Во время тестирования каждый тест регистрируется и его результаты сравниваются с ожидаемым. Если ожидаемый результат не совпадает с фактическим, это отмечается в протоколе тестирования. В таблице 1 представлена методика проведения тестирования разработанного программного продукта.

Выполненное действие	Полученный результат
Запуск исполняемого файла	Отображение формы авторизации.
Переход по кнопке "Зарегистрироваться"	Отображение формы регистрации.
Заполнение данных	Переход на страницу авторизации
Заполнение существующими данными	Вывод сообщения " Электронная почта/номер телефона уже зарегистрирован(а) "
Авторизация с ролью «ROLE_USER»	Отображение интерфейса пользователя.
Нажатие на кнопку «Добавить в корзину»	Товар перенесется к вам в корзину с указанным рядом количеством(при условии, что оно не превышает общее)
Переход по кнопке "Корзина"	Открытие списка добавленных вами товаров и их количество

Таблица 1 – методика тестирования разработанной программы

Нажатие на кнопку “Оформить заказ”	При условии наличия продуктов в корзине, откроется окно для ввода адреса, после успешного ввода создаться заказ
Нажатие на кнопку «Заказы»	Открытия списка всех неотработанных заказов
Нажатие на элемент списка	Переход к подробной информации о заказе
Переход по кнопке “История заказов”	Открытия списка всех отработанных заказов
Переход по кнопке “Выход”	Выход из аккаунта, переход на страницу авторизации
Авторизация с ролью «ROLE_ADMIN»	Отображение интерфейса администратора.
Нажатие кнопки редактировать на товаре	Открытые окна с возможностью редактирования информации или удаления товара

Результаты, полученные в ходе тестирования разработанного программного продукта, позволяют сделать заключение в том, что разработанная программа соответствует требованиям технического задания. Демонстрация тестирования и другие скриншоты показаны в приложении 3.

Заключение

В ходе выполнения курсового проекта было разработано веб-приложение для автоматизации работы предприятия, занимающегося доставкой продуктов питания. Разработка такого приложения представляет собой сложную задачу, требующую интеграции различных технических и функциональных компонентов.

На этапе анализа технического задания были выявлены основные требования к системе, необходимые для ее полноценной и корректной работы. Это включало реализацию CRUD-операций для управления данными о продуктах и заказах, разработку функционала для взаимодействия клиентов с системой, генерацию отчетов, а также обеспечение безопасности данных и адаптивности интерфейса.

На этапе разработки моделей данных была проведена детальная проработка структуры базы данных, что обеспечило основу для эффективного хранения и обработки информации. Проектирование архитектуры приложения позволило реализовать удобное и логичное взаимодействие между компонентами системы, обеспечив надежность и масштабируемость.

Фаза разработки и реализации включала создание основных модулей приложения с использованием фреймворка Spring. Были успешно интегрированы такие функции, как регистрация пользователей, управление данными о продуктах, оформление заказов, генерация отчетов в формате Excel и отправка уведомлений по электронной почте. Также реализован механизм разделения доступа в зависимости от роли пользователя, что обеспечивает безопасность и удобство работы.

Таким образом, выполнение поставленных задач позволило успешно разработать веб-приложение для автоматизации работы предприятия по доставке продуктов питания. Система обеспечивает эффективное управление данными, улучшает обслуживание клиентов и предоставляет руководству инструменты для анализа деятельности компании и принятия обоснованных решений.

					МИВУ 09.03.04 – 0.016 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		17

Список литературы:

1. Блох, Дж. Java. Эффективное программирование / Дж. Блох ; перевод В. Стрельцов ; под редакцией Р. Усманов. — 2-е изд. — Саратов : Профобразование, 2019. — 310 с. — ISBN 978-5-4488-0127-3. — Текст : электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <http://www.iprbookshop.ru/89870.html>
2. Свистунов, А. Н. Построение распределенных систем на Java : учебное пособие / А. Н. Свистунов. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 316 с. — ISBN 978-5-4497-0940-0. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/102045.html>
3. Мухамедзянов, Р. Р. JAVA. Серверные приложения / Р. Р. Мухамедзянов. — Москва : СОЛОН-Р, 2016. — 336 с. — ISBN 5-93455-134-5. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/90352.html>

Приложение

Приложение 1. Модели данных

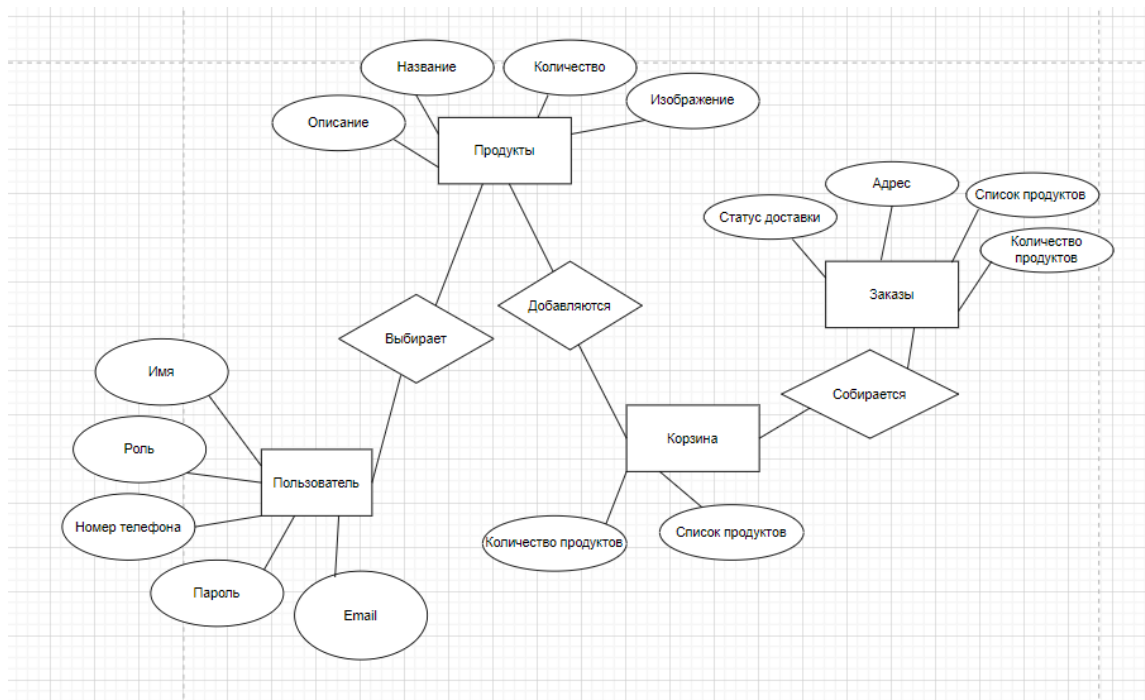


Рисунок 1 - Концептуальная модель данных

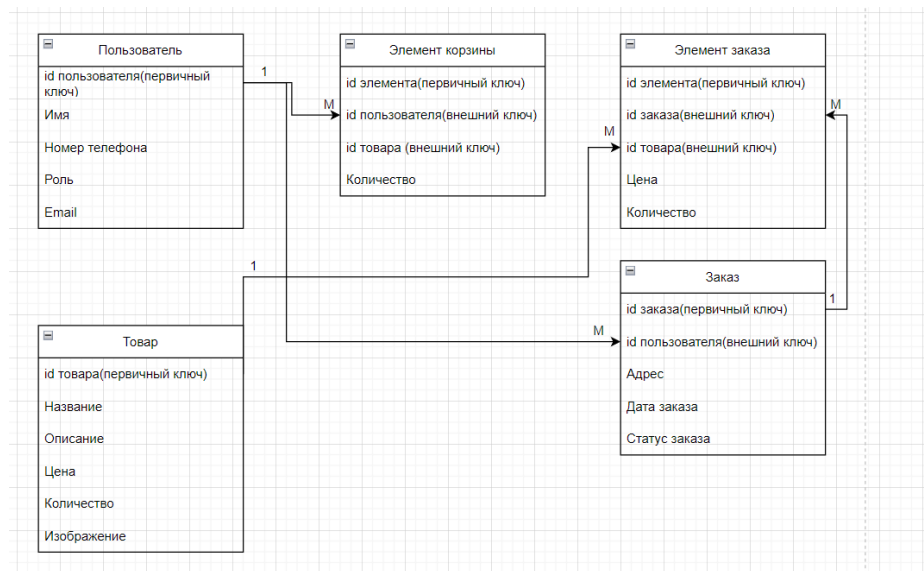


Рисунок 2 – Логическая модель данных

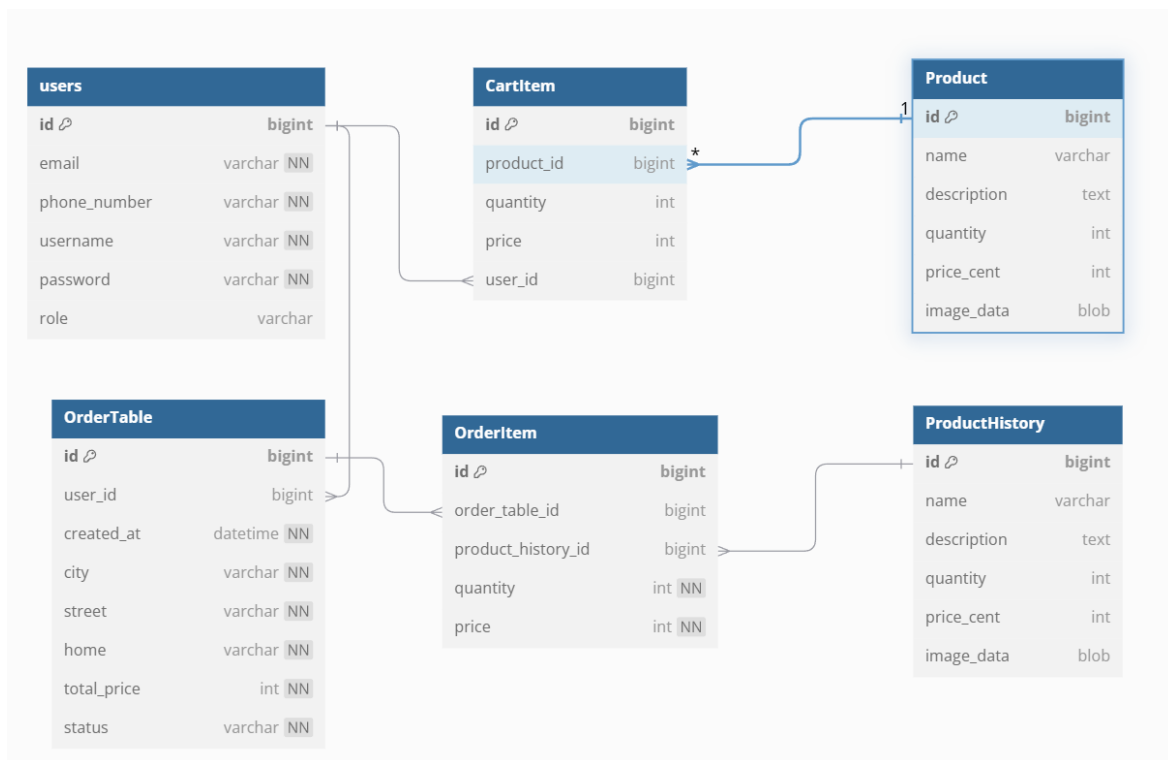


Рисунок 3 – Физическая модель данных

Приложение 2. Текст программы

ссылка на репозиторий GitHub с исходным кодом -
<https://github.com/Onzeboy/springcourse>

					МИВУ 09.03.04 – 0.016 ПЗ	Лист
						21
Изм.	Лист	№ докум.	Подпись	Дата		

Приложение 3. Снимки окон программы (скриншоты программы)

Авторизация

E-mail

Пароль

Войти

Нет аккаунта? Зарегистрируйтесь

Рисунок 1 – Стартовое окно(окно входа)

Регистрация

Имя

E-mail

Номер телефона

Пароль

Зарегистрироваться

Уже есть аккаунт? Авторизируйтесь

Рисунок 2 – Окно регистрации

Регистрация

Электронная почта уже зарегистрирована

Имя

E-mail

Номер телефона

Пароль

Зарегистрироваться

Уже есть аккаунт? Авторизируйтесь

Рисунок 3 – Попытка регистрации с уже зарегистрированной почтой

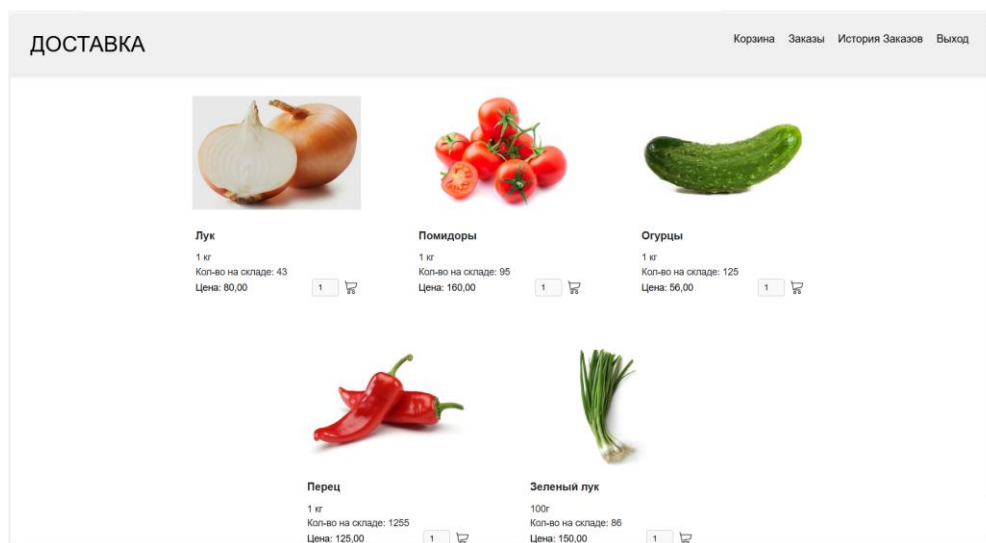


Рисунок 4 – Стартовая страница и каталог товаров

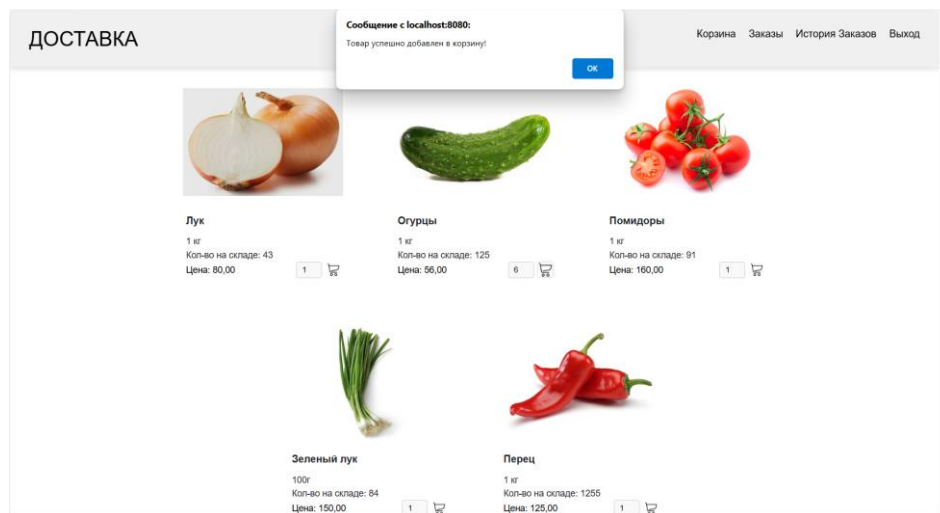


Рисунок 5 – Добавление товара в корзину

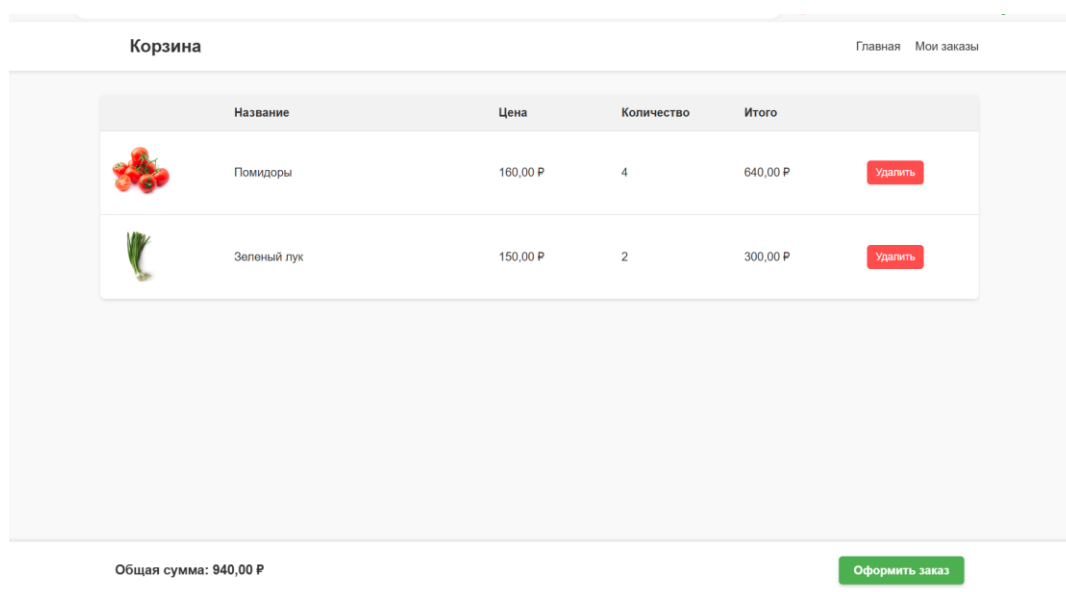


Рисунок 6 – Корзина

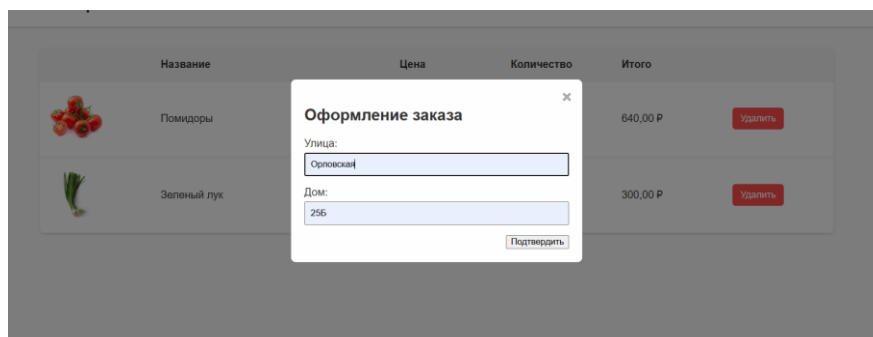


Рисунок 7 – Ввод адреса

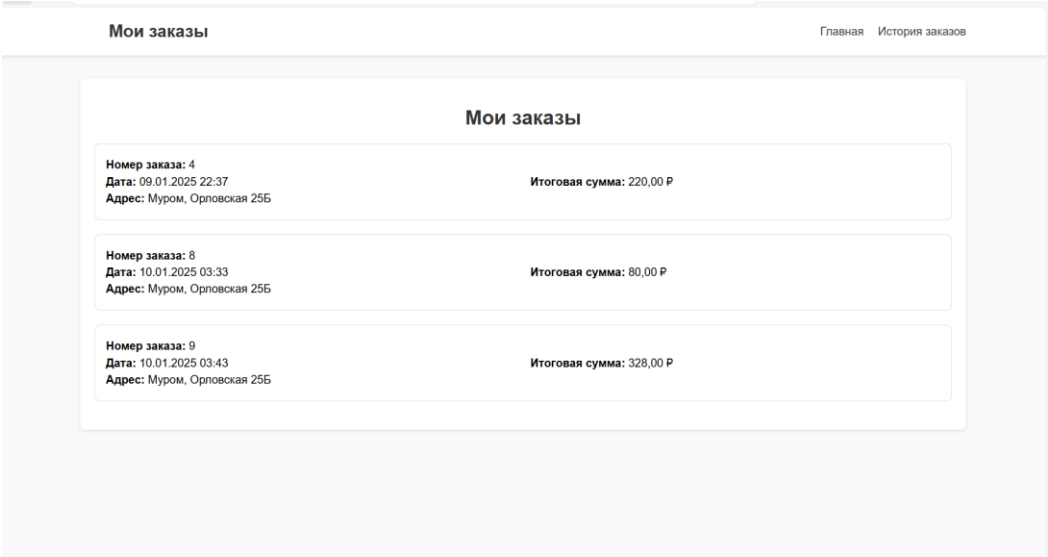


Рисунок 8 – Активные заказы

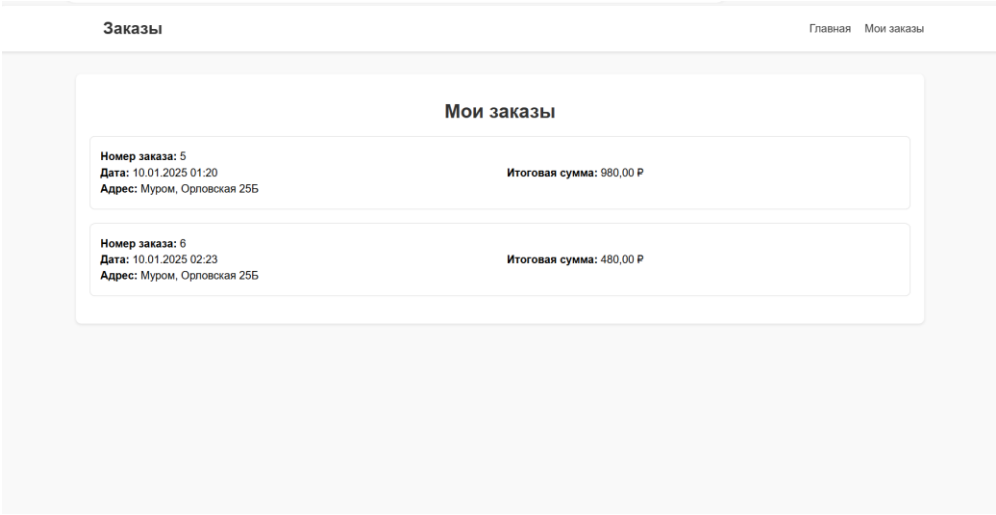


Рисунок 9 – История заказов

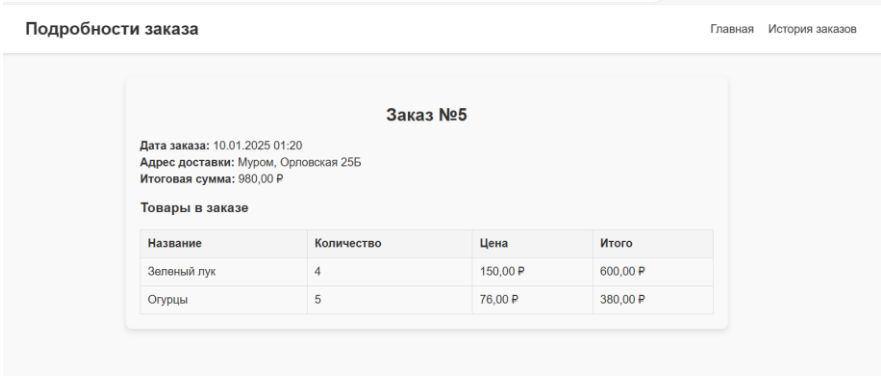


Рисунок 10 – Информация о заказе

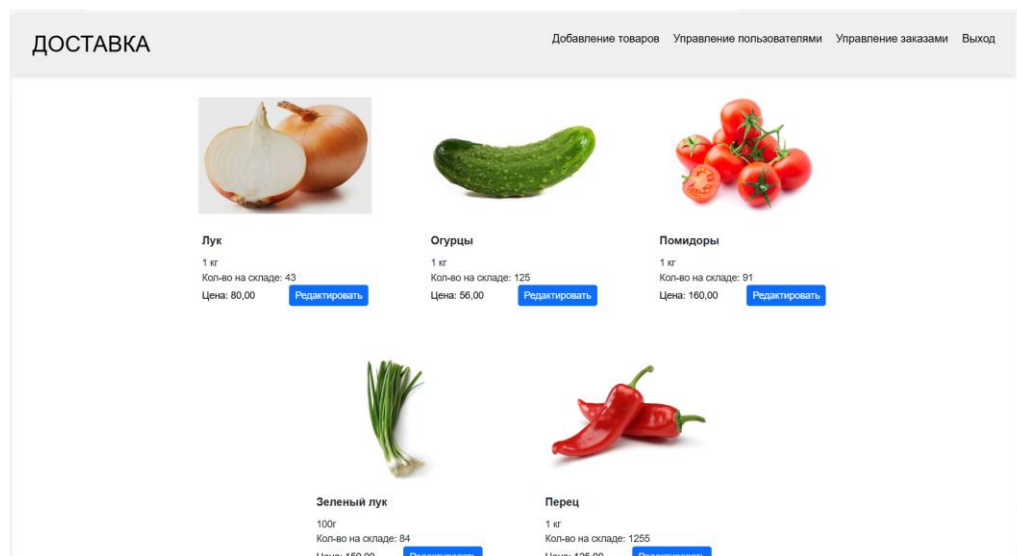


Рисунок 11 – Интерфейс администратора

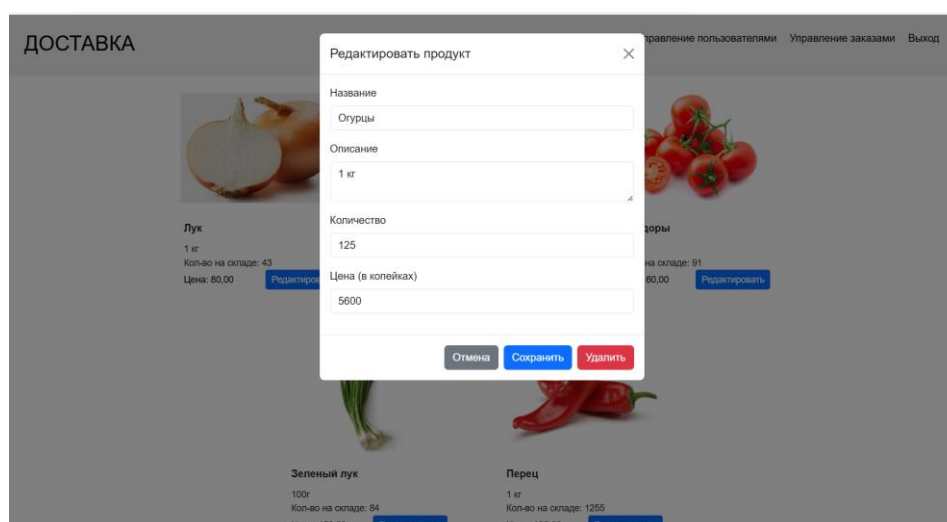


Рисунок 12 – Редактирование товара

Добавление товара

Название

Введите название

Размер

Введите описание

Цена

Введите цену (например: 10.50)

Количество

Введите количество

Изображение

Выбор файла

Не выбран ни один файл

Добавить товар

Рисунок 13 – Добавление товара

Управление заказами

ГлавнаяСоздать отчет

ID	Пользователь	Дата	Адрес	Сумма	Статус	Действия
4	Egor	09.01.2025 22:37	Муром, Орловская 25Б	220,00 Р	Создан	<div>ПросмотрИзменить статус</div>
5	Egor	10.01.2025 01:20	Муром, Орловская 25Б	980,00 Р	Доставлен	<div>ПросмотрИзменить статус</div>
6	Egor	10.01.2025 02:23	Муром, Орловская 25Б	980,00 Р	Доставлен	<div>ПросмотрИзменить статус</div>
8	Egor	10.01.2025 03:33	Муром, Орловская 25Б	980,00 Р	Доставлен	<div>ПросмотрИзменить статус</div>
9	Egor	10.01.2025 03:43	Муром, Орловская 25Б	980,00 Р	Доставлен	<div>ПросмотрИзменить статус</div>
7	Александр	10.01.2025 03:11	муром, Орловская 25Б	840,00 Р	Отправлен	<div>ПросмотрИзменить статус</div>

Рисунок 14 – Редактирование статуса заказа

Управление пользователями

Главная

ID	Имя пользователя	Email	Роли	Действия
3	Александр	onze@mail.ru	ROLE_USER	<div>Редактировать</div>
1	Egor	12@mail.ru	ROLE_USER	<div>Редактировать</div>

Рисунок 15 – Редактирование других аккаутов