**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**Рязанский государственный радиотехнический университет им. В.Ф. Уткина**

**Кафедра САПР ВС**

|  |  |  |
| --- | --- | --- |
|  |  | **К ЗАЩИТЕ**  руководитель КР  \_\_\_\_\_\_ А.Н. Сапрыкин  «\_\_\_»\_\_\_\_\_\_\_ 2025 г. |

**КУРСОВАЯ РАБОТА**

по дисциплине

**«Web-программирование»**

на тему

**«Разработка веб-сайта для кафе-кондитерской»**

Выполнил студент группы 247

Балоян А.А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата сдачи на проверку, подпись

Руководитель проекта

к.т.н., доцент

Сапрыкин А.Н. \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

оценка дата защиты, подпись

Рязань 2025

Министерство науки и высшего образования Российской Федерации

# ФГБОУ ВО «Рязанский государственный радиотехнический университет им. В.Ф. Уткина»

Факультет вычислительной техники

# КУРСОВАЯ РАБОТА

*по курсу «Web-программирование»*

Студент Балоян Арман Артурович группа 247

Тема: Разработка сайта для кафе-кондитерской

Срок представления работы к защите: «22» декабря 2024 г.

# Исходные данные для проектирования:

операционная система: *Windows XP/ Windows 7/ 8/ 10/ 11/ Linux/ macOS;*

язык/среда программирования: *HTML 5, CSS 3, JavaScript, LESS/SASS, библиотека Ajax/jQuery, Vue.js/ React.js.*

Требования к работе:

1. Семантическая верстка сайта должна соответствовать нотации HTML 5 и включать следующие разделы: <header>, <article>, <section>, <footer>, <nav>, <aside>.
2. Веб-сайт должен иметь адаптивную верстку, предусматривающую версию для смартфонов.
3. Все CSS-стили и JS-скрипты должны определяться во внешних файлах. Глобальные LESS-переменные должны быть заданы в отдельном LESS-файле.
4. Листинг кода страниц сайта (включая внешние файлы, например, .css, .js и др.) должен содержать поясняющие комментарии на русском языке
5. Пояснительная записка должна включать руководство пользователя по ГОСТ Р 59795 – 2021.
6. Библиографический список должен быть оформлен по ГОСТ Р 7.0.100 – 2018.
7. Требования к тексту: страницы текста и приложений должны соответствовать формату А4. Выполнение работы осуществляется печатным способом на одной стороне листа белой бумаги без рамок. Размер шрифта (высота букв, цифр и других знаков) 12 пт. (8 пт для приложений) через одинарный междустрочный интервал, отступ в первой строке абзаца 1,25 см и выравнивание текста по ширине. Границы текста следует печатать, соблюдая следующие размеры отступов от края листа: правое и нижнее – 10 мм, верхнее – 20 мм, левое – 30 мм.
8. Пояснительная записка представляется как в печатном, так и в электронном виде в соответствующем разделе личного портфолио студента на портале edu.rsreu.ru.

Задание принято к исполнению:

« » 2024 г.

подпись

Руководитель работы: *к.т.н., доцент Сапрыкин А.Н.*

« » 2024 г.

Подпись

**Содержание**

Введение............................................................................................................................4

1. Постановка задачи......................................................................................................5
2. Разработка веб-сайта.................................................................................................10
3. Руководство пользователя........................................................................................12

Заключение.......................................................................................................................14

Библиографический список............................................................................................15

Приложение А..................................................................................................................16

### **Введение**

В современном мире интернет-технологии играют ключевую роль в развитии бизнеса, особенно в сфере общественного питания и розничной торговли. Одним из наиболее востребованных направлений является создание интернет-магазинов, которые позволяют компаниям расширять свою аудиторию, увеличивать продажи и предоставлять клиентам удобный способ покупки товаров и услуг.

В данной курсовой работе рассматривается разработка веб-сайта для кафе-кондитерской "Сладкие Мечты", который предоставляет пользователям возможность выбора и заказа кондитерских изделий, а также управление товарами для администратора.

Целью работы является создание функционального и удобного веб-сайта, который будет соответствовать современным стандартам веб-разработки. Для достижения этой цели были поставлены следующие задачи:

1. Разработка семантической и адаптивной верстки сайта.
2. Реализация функционала для пользователей, включая выбор товаров, добавление их в корзину и оформление заказов.
3. Создание административной панели для добавления, редактирования и удаления товаров.
4. Обеспечение безопасности данных пользователей и администраторов.
5. Написание пояснительной записки, соответствующей требованиям ГОСТ.

В процессе разработки использовались современные технологии и инструменты, такие как HTML5, CSS3, JavaScript, фреймворк Vue.js, а также дополнительные библиотеки для работы с серверной частью и базой данных. Веб-сайт был разработан с учетом требований к адаптивности, что позволяет ему корректно отображаться на различных устройствах, включая смартфоны и планшеты.

**Основная часть**

1. **Постановка задачи**

Целью курсовой работы является разработка веб-сайта, написанного при помощи языков HTML, CSS и JavaScript в среде визуального программирования Visual Studio Code.

**Разработанный веб-сайт должен выполнять следующие функции:**

1. Позволяет пользователю оформить заказ, выбрав изделия, которые он хочет.
2. Войти в аккаунт как администратор или пользователь
3. Администратор может добавлять и удалять товар, после входа в систему
4. При оформлении заказа данные о заказе сохраняются в отдельном документе
5. При регистрации все данные сохраняются в отдельном документе
6. При добавлении товара администратором данные о новом товаре сохраняются в отдельном документе

На главной страничке пользователя встречает баннер “Добро пожаловать” и популярные продукты. На рисунке 1 представлен дизайн основной страницы веб-сайта.

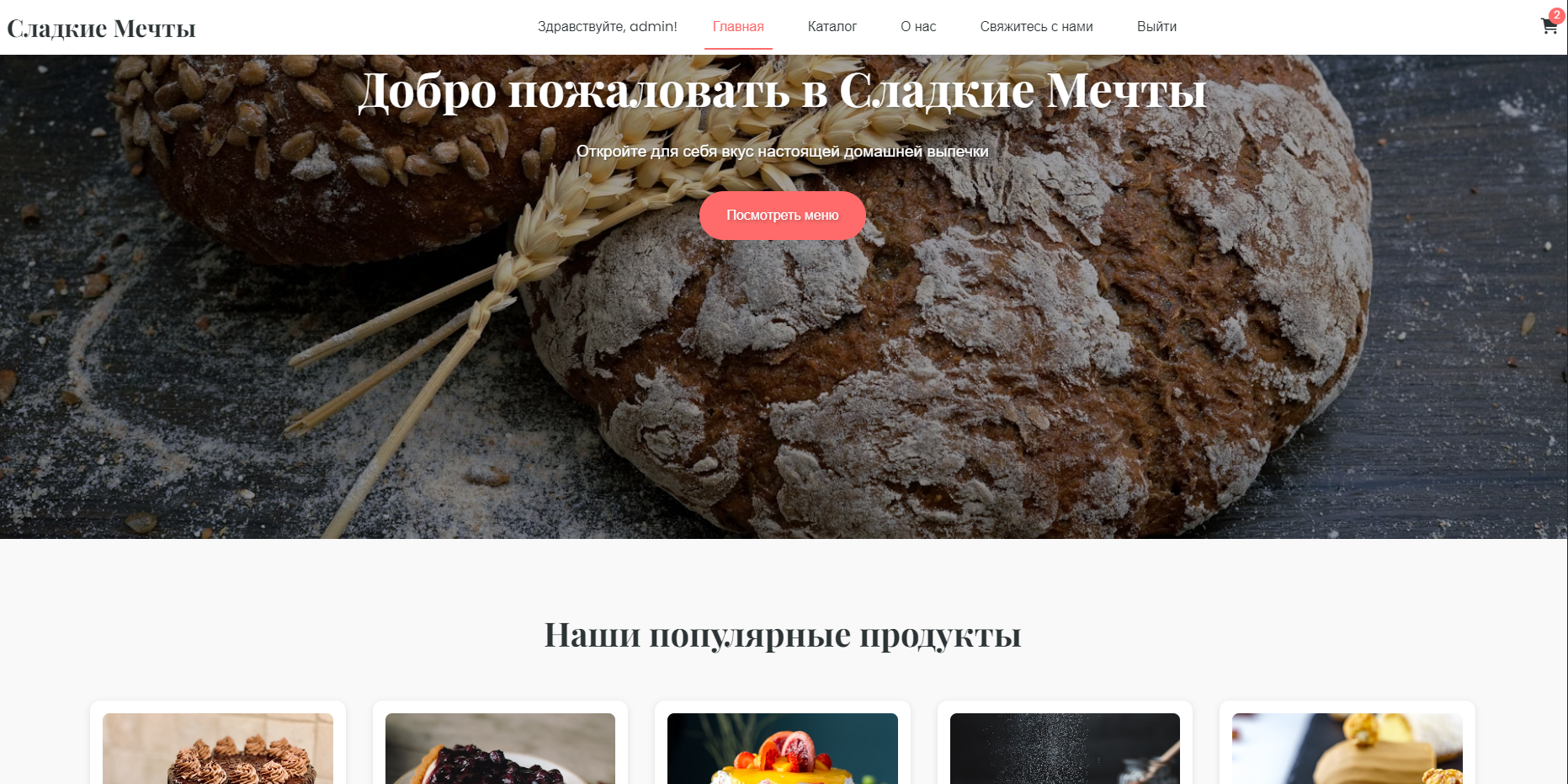


Рисунок 1 – Дизайн основной страницы веб-сайта

Сверху справа находятся основные функциональные кнопки такие как «О нас», «Свяжитесь с нами», «Вход». Если промотать ниже, то можно увидеть секции «О нас» и «Свяжитесь с нами». Они представлены на рисунке 2.

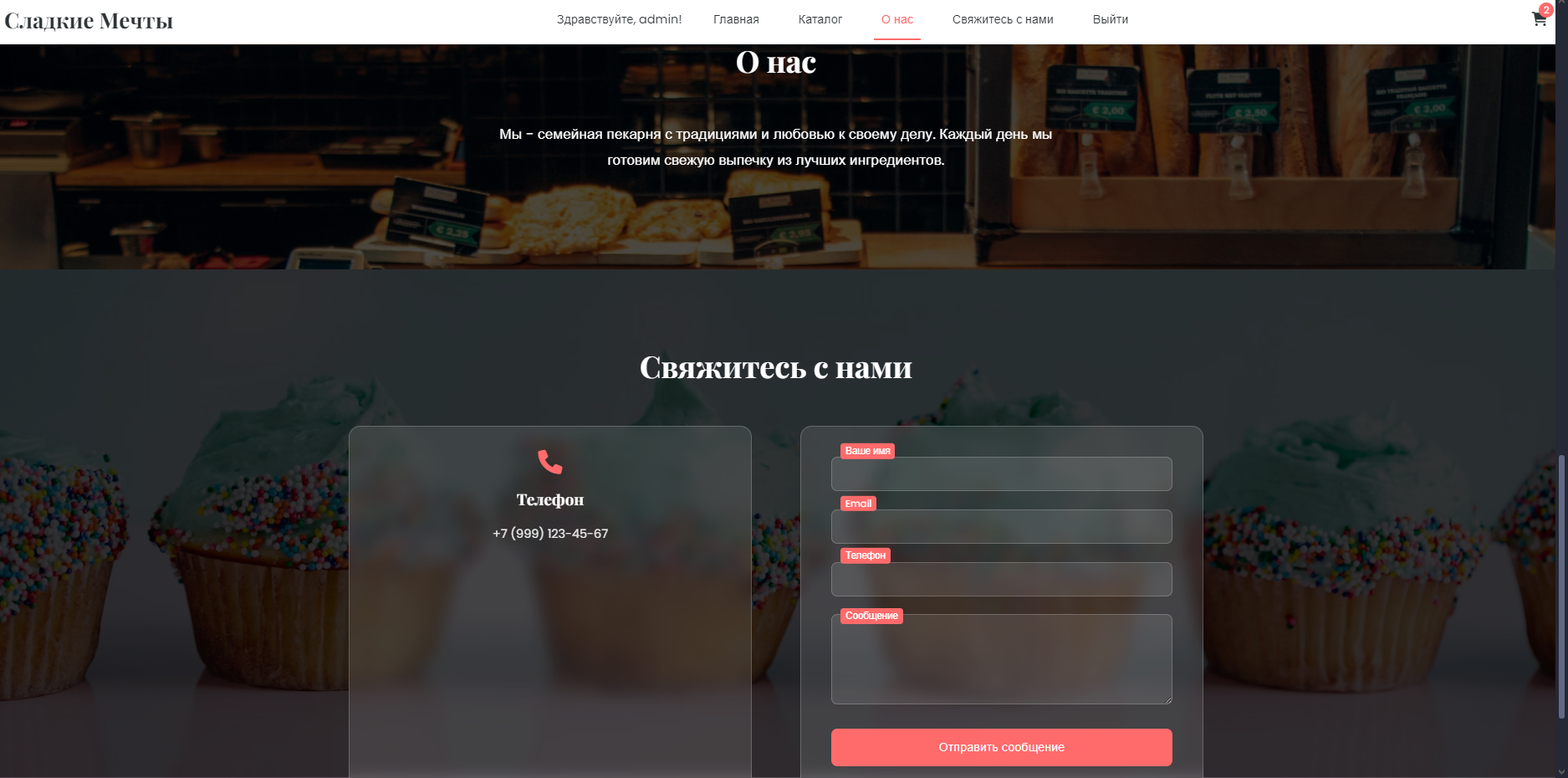


Рисунок 2 – Секции «О нас» и «Свяжитесь с нами»

При переходе на страничку каталога, пользователя встретят все карточки товаров вместе с боковой секцией для фильтрации и поиска товаров. Если пользователь является админом, для него появятся кнопки «Добавить новый продукт», «Удалить» и «Изменить». Страница показана на рисунке 3.

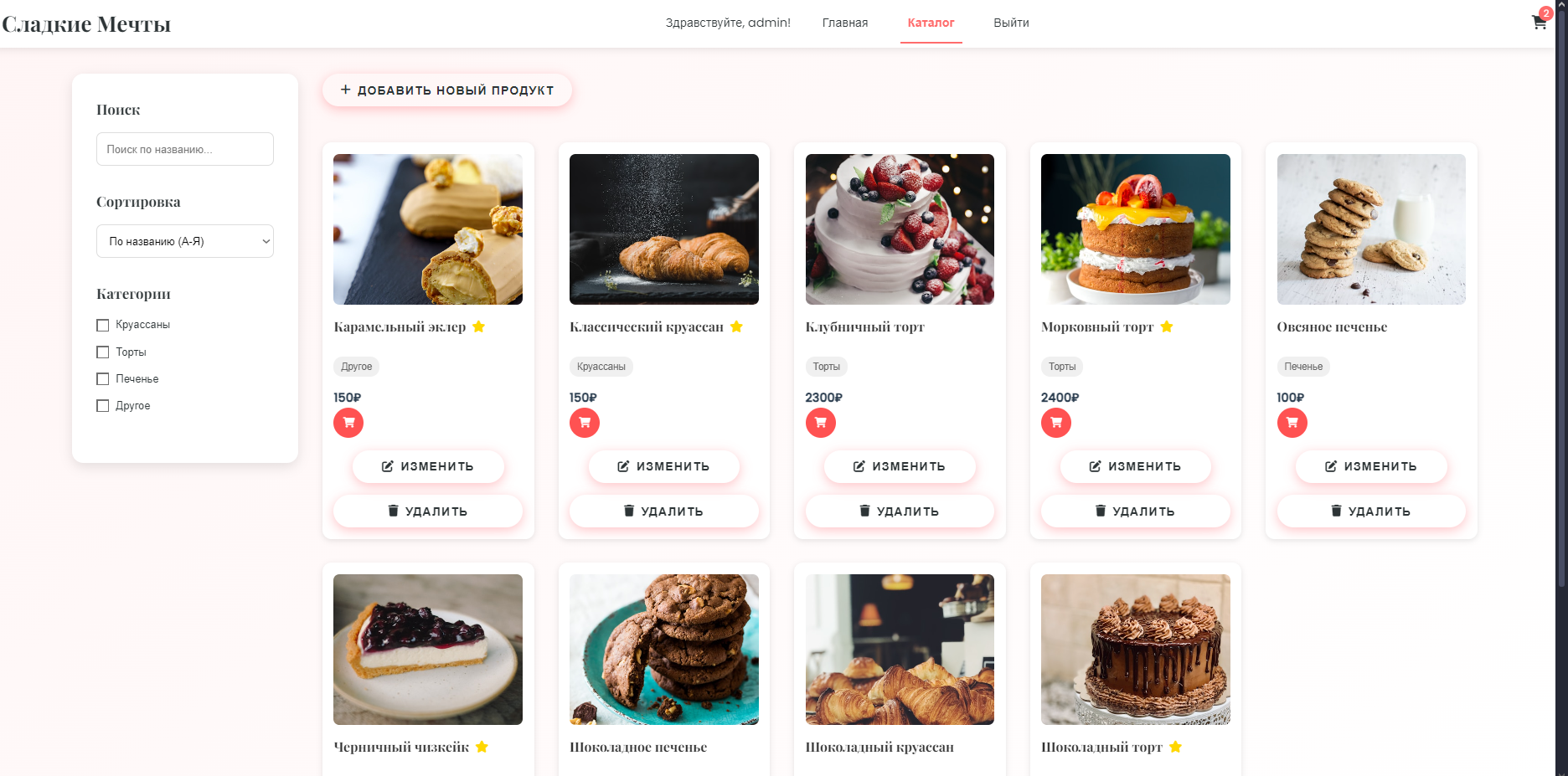


Рисунок 3 – Дизайн веб-страницы каталога

На рисунке 4 и 5 представлены окна авторизации и регистрации.

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 4 - Дизайн раздела окна авторизации

Изображение выглядит как текст, снимок экрана, еда

Автоматически созданное описание

Рисунок 5 - Дизайн раздела окна регистрации

На рисунке 6 представлен дизайн модального окна корзины.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 6 - Дизайн модального окна корзины

На рисунке 7 представлен дизайн окна добавления товара.

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

Рисунок 7 – Дизайн окна добавления товара

На рисунке 8 представлен дизайн модального окна изменения информации о товаре.

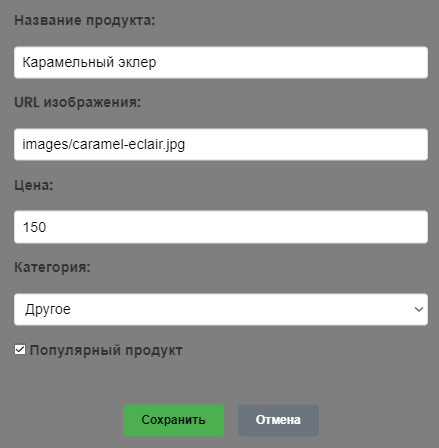


Рисунок 8 – Дизайн модального окна изменения информации о товаре

1. **Разработка веб-сайта**

По заданию был использованы языки HTML, CSS и JavaScript. Для простоты работы была использованы библиотеки Vue, jQuery.

* **Файл auth.js**

Отвечает за логику авторизации и регистрации пользователей. Основные функции: переключение между формами входа и регистрации, отправка данных на сервер для проверки и сохранения, сохранение данных пользователя в localStorage при успешной авторизации, перенаправление на главную страницу после успешного входа.

* **Файл catalog.js**

Управляет логикой каталога продуктов. Основные функции: загрузка продуктов с сервера, фильтрация и сортировка продуктов, добавление товаров в корзину, управление корзиной (изменение количества, удаление товаров), оформление заказа.

* **Файл ind.js**

Отвечает за логику главной страницы. Основные функции: загрузка популярных продуктов, добавление товаров в корзину, управление корзиной, отображение контактной формы и обработка её отправки.

* **Файл user-state.js**

Управляет состоянием пользователя. Основные функции: обновление навигации в зависимости от авторизации пользователя, выход из системы, отображение имени пользователя в навигации.

* **Файл server.py**

Локальный сервер, написанный на Python с использованием Flask. Основные функции: работа с пользователями (регистрация, вход), управление продуктами (добавление, обновление, удаление), обработка заказов (оформление, сохранение в orders.json),

* **Файл orders.json**

Содержит данные о всех заказах пользователей. Каждый заказ включает: идентификатор заказа, список товаров, общую сумму заказа, дату и время оформления.

* **Файл products.json**

Содержит данные о всех товарах в каталоге. Каждый товар включает: Идентификатор, Название, Цену, Изображение, Категорию, Флаг популярности.

* **Файл users.json**

Содержит данные о всех зарегистрированных пользователях. Каждый пользователь включает: Логин (email), Имя пользователя, Пароль, Роль (администратор или обычный пользователь).

* **Файл auth.html**

HTML-файл для страницы авторизации и регистрации. Включает: Форму входа, Форму регистрации, Переключение между формами.

* **Файл catalog.html**

HTML-файл для страницы каталога. Включает: Фильтры и сортировку, Сетку продуктов, Модальное окно корзины.

* **Файл ind.html**

HTML-файл для главной страницы. Включает: Главный экран с приветствием, Секцию популярных продуктов, Раздел "О нас", Контактную форму, Модальное окно корзины.

* **Файл catalog.css**

Стили для страницы каталога. Включает: Стили для фильтров и сортировки, Стили для карточек продуктов, Стили для корзины и модальных окон.

* **Файл common.less**

Общие стили для всех страниц, написанные на LESS. Включает: Переменные для цветов и шрифтов, Базовые стили для форм, кнопок и навигации, Стили для карточек и подвала.

* **Файл ind.css**

Стили для главной страницы. Включает: Стили для главного экрана, Стили для популярных продуктов, Стили для раздела "О нас" и контактной формы.

* **Файл auth.css**

Стили для страницы авторизации. Включает: Стили для форм входа и регистрации, Анимации и эффекты при переключении между формами.

Основной функционал прописан в файлах ind.js, catalog.js и server.py. Полный листинг всех файлов будет представлен в приложении А.

#### **3. Руководство пользователя**

#### **1. Введение**

Данное руководство предназначено для пользователей веб-сайта, разработанного для продажи изделий из кафе-кондитерской. Сайт предоставляет возможность выбора и покупки изделий, а также управление товарами для администраторов. Руководство поможет вам разобраться с основными функциями сайта и научит вас эффективно использовать его возможности.

#### **2**. **Основные функции сайта**

* **Просмотр товаров**: На главной странице сайта представлены карточки популярных товаров (выпечка). Вы можете просмотреть доступные товары, их цены. Если перейти на страничку с каталогом, то можно увидеть остальные товары.
* **Добавление товаров в корзину**: Для добавления товара в корзину нажмите на кнопку с иконкой корзины на карточке товара.
* **Оформление заказа**: После добавления товаров в корзину вы можете оформить заказ.
* **Регистрация и вход в аккаунт**: Для оформления заказа необходимо зарегистрироваться или войти в существующий аккаунт.
* **Управление товарами (для администраторов)**: Администраторы могут добавлять новые товары на сайт, удалять и изменять их.

#### **3.** **Пошаговое руководство**

##### **3.1**. **Просмотр товаров**

1. Откройте главную страницу сайта.
2. На главной странице вы увидите карточки популярных товаров. Каждая карточка содержит изображение, название, тип и цену товара.
3. Для просмотра всех товаров перейдите на страничку каталога.

##### **3.2. Добавление товаров в корзину**

1. На главной странице найдите понравившийся товар.
2. Нажмите на кнопку с иконкой корзины на карточке товара, чтобы добавить его в корзину.
3. Вы можете добавить несколько товаров в корзину.

##### **3.3.** **Просмотр корзины**

1. В верхнем правом углу страницы нажмите на иконку корзины.
2. В открывшемся окне вы увидите список добавленных товаров, их количество и общую стоимость.
3. Если вы хотите удалить товар из корзины, нажмите на иконку корзины рядом с товаром.

##### **3.4.** **Регистрация и вход в аккаунт**

1. В верхнем правом углу страницы нажмите на кнопку "Вход".
2. Если у вас уже есть аккаунт, введите логин и пароль, затем нажмите на кнопку "Войти".
3. Если у вас нет аккаунта, нажмите на кнопку "Зарегистрироваться".
4. Заполните поля:
   * **Имя**: Введите ваше имя.
   * **Email**: Введите email.
   * **Пароль**: введите пароль.
   * **Подтвердите** пароль: повторите пароль.
5. Нажмите на кнопку "Зарегистрироваться".
6. После регистрации вы сможете войти в аккаунт, используя указанные логин и пароль.

##### **3.5**. **Добавление товаров (для администраторов)**

1. Войдите в аккаунт администратора.
2. Перейдите на страничку каталога.
3. Нажмите на кнопку "Добавить новый продукт".
4. Заполните форму:
   * **Название**: Введите название товара.
   * **Цена**: Введите цену товара.
   * **Изображение**: Напишите URL изображения товара.
   * **Категория**: Укажите категорию товара.
   * **Популярный продукт**: Укажите, чтобы товар появился на главной страничке.
5. Нажмите на кнопку "Добавить".
6. Товар будет добавлен на сайт и станет доступен для покупки.

##### **3.6**. **Изменение или удаление товаров (для администраторов)**

1. Войдите в аккаунт администратора.
2. Перейдите на страничку каталога.
3. Нажмите на кнопку "Изменить" или “Удалить” на карточке продукта, который хотите изменить или удалить.
4. При изменении заполните форму:
   * **Название**: Введите название товара.
   * **Цена**: Введите цену товара.
   * **Изображение**: Напишите URL изображения товара.
   * **Категория**: Укажите категорию товара.
   * **Популярный продукт**: Укажите, чтобы товар появился на главной страничке.
5. Нажмите на кнопку "Сохранить".
6. Товар будет изменен или удален, в зависимости от выбора.

#### **4. Часто задаваемые вопросы (FAQ)**

##### **4.1. Как изменить данные в заказе?**

После оформления заказа изменить данные невозможно. Если вы допустили ошибку, свяжитесь с поддержкой по указанным на сайте контактам.

##### **4.2.** **Как отменить заказ?**

Для отмены заказа свяжитесь с поддержкой по указанным на сайте контактам.

##### **4.3.** **Как связаться с поддержкой?**

В секции "Свяжитесь с нами" вы найдете телефон магазина. Вы можете связаться с поддержкой.

#### **5.** **Заключение**

Данное руководство поможет вам эффективно использовать веб-сайт для покупки кондитерских изделий. Если у вас возникнут дополнительные вопросы, вы всегда можете обратиться в поддержку через раздел "Свяжитесь с нами". Надеемся, что наш сайт сделает ваш процесс покупки кондитерских изделий удобным и приятным!

### **Заключение**

В ходе выполнения курсовой работы был разработан веб-сайт для продажи кондитерских изделий, который соответствует всем поставленным задачам и требованиям. Сайт предоставляет пользователям удобный интерфейс для выбора и покупки кондитерских изделий, а также позволяет администраторам управлять ассортиментом товаров. Была реализована адаптивная верстка, что делает сайт доступным для использования на различных устройствах.

Основные функции сайта включают:

* Возможность выбора товаров и добавления их в корзину.
* Оформление заказов.
* Регистрация и авторизация пользователей и администраторов.
* Управление товарами.

В процессе разработки были использованы современные технологии веб-программирования, такие как Vue, LESS, JQuery, что позволило создать динамичное и интерактивное веб-приложение. Все стили и скрипты были вынесены в отдельные файлы, что упрощает поддержку и дальнейшее развитие проекта.

В заключение можно сказать, что поставленные задачи были успешно выполнены, и разработанный веб-сайт готов к использованию. В будущем возможно расширение функционала, например, добавление новых категорий товаров и улучшение пользовательского интерфейса.

**Библиографический список**

1. Книги и учебные пособия:
   1. Иванов И.И. Веб-программирование: учебное пособие. — М.: Издательство, 2023. — 300 с.
   2. Петров П.П. HTML5 и CSS3: основы разработки сайтов. — СПб.: Издательство, 2022. — 250 с.
2. Официальная документация и стандарты:  
    3. ГОСТ Р 59795–2021. Руководства пользователя программного обеспечения. Общие требования. — М.: Стандартинформ, 2021. — 20 с.  
    4. ГОСТ Р 7.0.100–2018. Библиографическая запись. Библиографическое описание. Общие требования и правила составления. — М.: Стандартинформ, 2018. — 35 с.
3. Онлайн-ресурсы:  
    5. Официальная документация Vue.js. — URL: <https://vuejs.org/> (дата обращения: 10.10.2024).  
    6. Руководство по адаптивной верстке на MDN Web Docs. — URL: <https://developer.mozilla.org/ru/docs/Learn/CSS/CSS_layout/Responsive_Design> (дата обращения: 10.10.2024).

**Приложение А**

**Листинг программы файла ind.html**

<!DOCTYPE html>

<html lang="ru">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Сладкие Мечты - Пекарня</title>

    <link rel="icon" href="images/icon.png" type="image/x-icon">

    <link rel="stylesheet/less" type="text/css" href="css/common.less">

    <link rel="stylesheet" href="css/ind.css">

    <link href="https://fonts.googleapis.com/css2?family=Playfair+Display:wght@400;700&family=Poppins:wght@300;400;500&display=swap" rel="stylesheet">

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">

    <script src="https://cdn.jsdelivr.net/npm/less@4.1.1"></script>

    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

    <script src="https://cdn.jsdelivr.net/npm/vue@2.6.14/dist/vue.js"></script>

    <script src="js/ind.js"></script>

    <script src="js/user-state.js"></script>

</head>

<body>

    <div id="app">

        <header>

            <nav class="navbar">

                <a href="ind.html" class="logo">Сладкие Мечты</a>

                <div class="nav-links">

                    <span id="userName" style="display: none;"></span>

                    <a href="ind.html" class="active">Главная</a>

                    <a href="catalog.html">Каталог</a>

                    <a href="#about">О нас</a>

                    <a href="#contact">Свяжитесь с нами</a>

                    <a href="auth.html" id="loginButton" class="login-link">Войти</a>

                    <a href="#" id="logoutButton" style="display: none;">Выйти</a>

                </div>

                <div class="cart-container">

                    <div class="cart-icon">

                        <i class="fas fa-shopping-cart"></i>

                        <span class="cart-count">0</span>

                    </div>

                </div>

            </nav>

        </header>

        <main>

            <header id="home" class="hero">

                <div class="hero-content">

                    <h1>Добро пожаловать в Сладкие Мечты</h1>

                    <p>Откройте для себя вкус настоящей домашней выпечки</p>

                    <a href="catalog.html" class="cta-button">Посмотреть меню</a>

                </div>

            </header>

            <section id="products" class="products">

                <h2>Наши популярные продукты</h2>

                <div class="product-grid">

                    <div class="product-card" v-for="product in products" :key="product.id">

                        <img :src="product.image" :alt="product.name" class="product-image">

                        <div class="product-info">

                            <div class="product-title">

                                <h3 class="product-name">{{ product.name }}</h3>

                                <i v-if="product.isPopular" class="fas fa-star star-icon"></i>

                            </div>

                            <span class="product-price">{{ product.price }}₽</span>

                            <button class="add-to-cart" @click="handleAddToCart(product)">

                                <i class="fas fa-shopping-cart"></i>

                            </button>

                        </div>

                    </div>

                </div>

            </section>

            <section id="about" class="about">

                <div class="about-container">

                    <div class="about-content">

                        <h2>О нас</h2>

                        <p class="about-description">Мы - семейная пекарня с традициями и любовью к своему делу. Каждый день мы готовим свежую выпечку из лучших ингредиентов.</p>

                    </div>

                </div>

            </section>

            <section id="contact" class="contact">

                <div class="contact-container">

                    <h2>Свяжитесь с нами</h2>

                    <div class="contact-content">

                        <div class="contact-info">

                            <div class="contact-card">

                                <i class="fas fa-phone"></i>

                                <h3>Телефон</h3>

                                <p>+7 (999) 123-45-67</p>

                            </div>

                        </div>

                        <div class="contact-form-container">

                            <form class="contact-form">

                                <div class="form-group">

                                    <input type="text" id="name" name="name" required>

                                    <label for="name">Ваше имя</label>

                                </div>

                                <div class="form-group">

                                    <input type="email" id="email" name="email" required>

                                    <label for="email">Email</label>

                                </div>

                                <div class="form-group">

                                    <input type="tel" id="phone" name="phone" required>

                                    <label for="phone">Телефон</label>

                                </div>

                                <div class="form-group">

                                    <textarea id="message" name="message" required></textarea>

                                    <label for="message">Сообщение</label>

                                </div>

                                <button type="submit" class="submit-button">Отправить сообщение</button>

                            </form>

                        </div>

                    </div>

                </div>

            </section>

        </main>

        <div class="cart-modal" id="cartModal">

            <div class="cart-content">

                <h2>Ваша корзина</h2>

                <div class="cart-items"></div>

                <div class="cart-total">

                    <span>Итого:</span>

                    <span class="total-amount">0₽</span>

                </div>

                <button class="checkout-button">Оформить заказ</button>

            </div>

        </div>

        <footer class="footer">

            <p>&copy; 2025 Сладкие Мечты. Все права защищены.</p>

        </footer>

</div>

</body>

</html>

**Листинг программы файла catalog.html**

<!DOCTYPE html>

<html lang="ru">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Каталог - Сладкие Мечты</title>

    <link rel="stylesheet/less" type="text/css" href="css/common.less">

    <link rel="stylesheet" href="css/catalog.css">

    <link href="https://fonts.googleapis.com/css2?family=Playfair+Display:wght@400;700&family=Poppins:wght@300;400;500&display=swap" rel="stylesheet">

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">

    <script src="https://cdn.jsdelivr.net/npm/less@4.1.1"></script>

    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

    <script src="https://cdn.jsdelivr.net/npm/vue@2.6.14/dist/vue.js"></script>

    <script src="js/user-state.js"></script>

    <script src="js/catalog.js"></script>

</head>

<body>

    <div id="app">

*<!-- Заголовок раздела навигации -->*

<header>

    <nav class="navbar">

        <a href="ind.html" class="logo">Сладкие Мечты</a>

        <div class="nav-links">

            <span id="userName" style="display: none;"></span>

            <a href="ind.html">Главная</a>

            <a href="catalog.html" class="active">Каталог</a>

            <a href="auth.html" id="loginButton" class="login-link">Войти</a>

            <a href="#" id="logoutButton" style="display: none;">Выйти</a>

        </div>

        <div class="cart-container">

            <div class="cart-icon">

                <i class="fas fa-shopping-cart"></i>

                <span class="cart-count">0</span>

            </div>

        </div>

    </nav>

</header>

    <div class="catalog-container">

        <aside class="filters">

            <div class="filter-section">

                <h3>Поиск</h3>

                <div class="search-container">

                    <input type="text" id="search-input" placeholder="Поиск по названию...">

                </div>

            </div>

            <div class="filter-section">

                <h3>Сортировка</h3>

                <div class="sort-container">

                    <select id="sort-select">

                        <option value="nameAsc">По названию (А-Я)</option>

                        <option value="nameDesc">По названию (Я-А)</option>

                        <option value="priceAsc">По цене (возрастание)</option>

                        <option value="priceDesc">По цене (убывание)</option>

                    </select>

                </div>

            </div>

            <div class="filter-section">

                <h3>Категории</h3>

                <div class="category-checkboxes">

                    <label>

                        <input type="checkbox" name="category" value="Круассаны"> Круассаны

                    </label>

                    <label>

                        <input type="checkbox" name="category" value="Торты"> Торты

                    </label>

                    <label>

                        <input type="checkbox" name="category" value="Печенье"> Печенье

                    </label>

                    <label>

                        <input type="checkbox" name="category" value="Другое"> Другое

                    </label>

                </div>

            </div>

        </aside>

*<!-- Основная область контента для продуктов каталога -->*

        <main class="catalog-content">

            <div class="catalog-section">

                <div class="product-grid">

*<!-- Продукты будут здесь -->*

                </div>

            </div>

        </main>

    </div>

    <div class="cart-modal" id="cartModal">

        <div class="cart-content">

            <h2>Ваша корзина</h2>

            <div class="cart-items"></div>

            <div class="cart-total">

                <span>Итого:</span>

                <span class="total-amount">0₽</span>

            </div>

            <button class="checkout-button">Оформить заказ</button>

        </div>

    </div>

    </div>

*<!-- Нижний колонтитул с информацией об авторских правах -->*

    <footer class="footer">

        <p>&copy; 2025 Сладкие Мечты. Все права защищены.</p>

    </footer>

</body>

</html>

**Листинг программы файла auth.html**

<!DOCTYPE html>

<html lang="ru">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Авторизация - Сладкие Мечты</title>

    <link rel="stylesheet/less" type="text/css" href="css/common.less">

    <link rel="stylesheet" href="css/auth.css">

    <link href="https://fonts.googleapis.com/css2?family=Playfair+Display:wght@400;700&family=Poppins:wght@300;400;500&display=swap" rel="stylesheet">

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">

    <script src="https://cdn.jsdelivr.net/npm/less@4.1.1"></script>

    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

    <script src="https://cdn.jsdelivr.net/npm/vue@2.6.14/dist/vue.js"></script>

    <script src="js/user-state.js"></script>

    <script src="js/auth.js"></script>

</head>

<body>

    <div id="app">

*<!-- Заголовок раздела навигации -->*

    <header>

        <nav class="navbar">

            <a href="ind.html" class="logo">Сладкие Мечты</a>

            <div class="nav-links">

                <span id="userName" style="display: none;"></span>

                <a href="ind.html">Главная</a>

                <a href="catalog.html">Каталог</a>

                <a href="auth.html" class="active">Войти</a>

            </div>

        </nav>

    </header>

*<!-- Основная область контента -->*

    <main>

        <section class="auth-container">

            <div class="auth-box">

                <div class="auth-tabs">

                    <button class="auth-tab active" data-tab="login">Вход</button>

                    <button class="auth-tab" data-tab="register">Регистрация</button>

                </div>

*<!-- Форма входа -->*

                <form id="loginForm" class="auth-form active" @submit.prevent="handleLogin">

                    <h2>Добро пожаловать!</h2>

                    <div class="form-group">

                        <i class="fas fa-envelope"></i>

                        <input type="email" v-model="loginEmail" placeholder="Email" required>

                    </div>

                    <div class="form-group">

                        <i class="fas fa-lock"></i>

                        <input type="password" v-model="loginPassword" placeholder="Пароль" required>

                    </div>

                    <div class="form-options">

                        <label class="remember-me">

                            <input type="checkbox" v-model="rememberMe">

                            <span>Запомнить меня</span>

                        </label>

                        <a href="ind.html#contact" class="forgot-password">Забыли пароль?</a>

                    </div>

                    <button type="submit" class="auth-button">Войти</button>

                </form>

*<!-- Форма регистрации -->*

                <form id="registerForm" class="auth-form" @submit.prevent="handleRegister">

                    <h2>Создать аккаунт</h2>

                    <div class="form-group">

                        <i class="fas fa-user"></i>

                        <input type="text" v-model="registerName" placeholder="Имя" required>

                    </div>

                    <div class="form-group">

                        <i class="fas fa-envelope"></i>

                        <input type="email" v-model="registerEmail" placeholder="Email" required>

                    </div>

                    <div class="form-group">

                        <i class="fas fa-lock"></i>

                        <input type="password" v-model="registerPassword" placeholder="Пароль" required>

                    </div>

                    <div class="form-group">

                        <i class="fas fa-lock"></i>

                        <input type="password" v-model="confirmPassword" placeholder="Подтвердите пароль" required>

                    </div>

                    <button type="submit" class="auth-button">Зарегистрироваться</button>

                </form>

            </div>

        </section>

    </main>

*<!-- Нижний колонтитул с информацией об авторских правах -->*

    <footer class="footer">

        <p>&copy; 2025 Сладкие Мечты. Все права защищены.</p>

    </footer>

</div>

</body>

</html>

**Листинг программы файла server.py**

*from* flask *import* Flask, jsonify, request, send\_from\_directory, render\_template

*from* flask\_cors *import* CORS

*import* json

*import* os

*import* smtplib

*from* flask\_mail *import* Mail, Message

*from* email.mime.text *import* MIMEText

*from* email.mime.multipart *import* MIMEMultipart

app = Flask(\_\_name\_\_)

CORS(app)

*# Пути к файлам*

PRODUCTS\_FILE = 'data/products.json'

USERS\_FILE = 'data/users.json'

ORDERS\_FILE = 'data/orders.json'

*# Загрузка продуктов из файла*

def **load\_products**():

*if* not os.path.exists(PRODUCTS\_FILE):

*return* []

*with* open(PRODUCTS\_FILE, 'r', *encoding*='utf-8') *as* f:

*return* json.load(f)

*# Сохранение продуктов в файл*

def **save\_products**(*products*):

*with* open(PRODUCTS\_FILE, 'w', *encoding*='utf-8') *as* f:

        json.dump(*products*, f, *ensure\_ascii*=False, *indent*=4)

*# Загрузка пользователей из файла*

def **load\_users**():

*if* not os.path.exists(USERS\_FILE):

*return* {"users": []}

*with* open(USERS\_FILE, 'r', *encoding*='utf-8') *as* f:

*return* json.load(f)

*# Сохранение пользователей в файл*

def **save\_users**(*users\_data*):

*with* open(USERS\_FILE, 'w', *encoding*='utf-8') *as* f:

        json.dump(*users\_data*, f, *ensure\_ascii*=False, *indent*=4)

*# Загрузка заказов из файла*

def **load\_orders**():

*if* not os.path.exists(ORDERS\_FILE):

*return* []

*with* open(ORDERS\_FILE, 'r', *encoding*='utf-8') *as* f:

*return* json.load(f)

*# Сохранение заказов в файл*

def **save\_orders**(*orders*):

*with* open(ORDERS\_FILE, 'w', *encoding*='utf-8') *as* f:

        json.dump(*orders*, f, *ensure\_ascii*=False, *indent*=4)

*# API для работы с пользователями*

@app.route('/api/users', *methods*=['GET'])

def **get\_users**():

    users = load\_users()

*return* jsonify(users)

@app.route('/api/users/register', *methods*=['POST'])

def **register\_user**():

    new\_user = request.json

    users\_data = load\_users()

*# Проверяем, существует ли пользователь*

*if* any(user['login'] == new\_user['login'] *for* user *in* users\_data['users']):

*return* jsonify({'error': 'User already exists'}), 400

*# Добавляем нового пользователя*

    users\_data['users'].append(new\_user)

    save\_users(users\_data)

*return* jsonify({'message': 'User registered successfully'}), 201

@app.route('/api/users/login', *methods*=['POST'])

def **login\_user**():

    login\_data = request.json

    users\_data = load\_users()

*# Ищем пользователя*

    user = next((user *for* user *in* users\_data['users']

*if* user['login'] == login\_data['login']

                 and user['password'] == login\_data['password']), None)

*if* user:

*return* jsonify({'message': 'Login successful', 'user': user}), 200

*else*:

*return* jsonify({'error': 'Invalid credentials'}), 401

*# Получение всех продуктов*

@app.route('/api/products', *methods*=['GET'])

def **get\_products**():

*try*:

*with* open('data/products.json', 'r', *encoding*='utf-8') *as* f:

            products = json.load(f)

*return* jsonify(products)

*except* Exception *as* e:

*return* jsonify({'error': str(e)}), 500

*# Добавление нового продукта*

@app.route('/api/products', *methods*=['POST'])

def **add\_product**():

    new\_product = request.json

    products = load\_products()

    products.append(new\_product)

    save\_products(products)

*return* jsonify(new\_product), 201

*# Обновление продукта*

@app.route('/api/products/<int:product\_id>', *methods*=['PUT'])

def **update\_product**(*product\_id*):

    updated\_product = request.json

    products = load\_products()

*for* index, product *in* enumerate(products):

*if* product['id'] == *product\_id*:

            products[index] = updated\_product

            save\_products(products)

*return* jsonify(updated\_product)

*return* jsonify({'error': 'Product not found'}), 404

*# Удаление продукта*

@app.route('/api/products/<int:product\_id>', *methods*=['DELETE'])

def **delete\_product**(*product\_id*):

    products = load\_products()

*for* index, product *in* enumerate(products):

*if* product['id'] == *product\_id*:

*del* products[index]

            save\_products(products)

*return* '', 204

*return* jsonify({'error': 'Product not found'}), 404

*# Обслуживание статических файлов*

@app.route('/<path:path>')

def **serve\_static**(*path*):

*return* send\_from\_directory('.', *path*)

*# Обслуживание данных*

@app.route('/data/<path:path>')

def **serve\_data**(*path*):

*return* send\_from\_directory('data', *path*)

*# Обслуживание JavaScript файлов*

@app.route('/js/<path:path>')

def **serve\_js**(*path*):

*return* send\_from\_directory('js', *path*)

*# Обслуживание изображений*

@app.route('/images/<path:path>')

def **send\_image**(*path*):

*return* send\_from\_directory('images', *path*)

mail = Mail(app)

def **send\_receipt**(*email*, *order\_details*):

    msg = Message('Ваш чек', *sender*='noreply@вашсайт.com', *recipients*=[*email*])

    msg.body = f'Спасибо за ваш заказ!\n\nДетали заказа:\n{*order\_details*}'

    mail.send(msg)

@app.route('/api/send-receipt', *methods*=['POST'])

def **send\_receipt**():

    data = request.json

    email = data.get('email')

    order\_details = data.get('orderDetails')

*# Настройка SMTP*

    sender\_email = "your-email@gmail.com"

    sender\_password = "your-email-password"

*# Создание сообщения*

    msg = MIMEMultipart()

    msg['From'] = sender\_email

    msg['To'] = email

    msg['Subject'] = 'Ваш заказ в Сладкие Мечты'

    body = f"Спасибо за ваш заказ! Вот детали вашего заказа:\n\n{order\_details}"

    msg.attach(MIMEText(body, 'plain'))

*try*:

*# Отправка письма*

*with* smtplib.SMTP('smtp.gmail.com', 587) *as* server:

            server.starttls()

            server.login(sender\_email, sender\_password)

            server.send\_message(msg)

*return* jsonify({'message': 'Чек отправлен'}), 200

*except* Exception *as* e:

*return* jsonify({'error': str(e)}), 500

*# Обработка оформления заказа*

@app.route('/api/orders', *methods*=['POST'])

def **place\_order**():

    order\_data = request.json

    orders = load\_orders()

*# Присваиваем ID на основе ID последнего заказа*

*if* orders:

        last\_order\_id = orders[-1]['id']  *# Получаем ID последнего заказа*

        order\_data['id'] = last\_order\_id + 1  *# Присваиваем новый ID*

*else*:

        order\_data['id'] = 1  *# Если нет заказов, начинаем с 1*

    orders.append(order\_data)

    save\_orders(orders)

*return* jsonify({'message': 'Order placed successfully', 'order\_id': order\_data['id']}), 201

*if* \_\_name\_\_ == '\_\_main\_\_':

    app.run(*debug*=True)

**Листинг программы файла common.less**

*// Переменные*

@primary-color: #ff6b6b;

@secondary-color: #4ecdc4;

@text-color: #2d3436;

@font-primary: 'Poppins', sans-serif;

@font-secondary: 'Playfair Display', serif;

:root {

    --primary-color: #ff5252; */\* Пример значения \*/*

    --secondary-color: #4ecdc4;

    --text-color: #2d3436;

    --font-primary: 'Poppins', sans-serif;

    --font-secondary: 'Playfair Display', serif;

}

*// Сброс и базовые стили*

\* {

    margin: 0;

    padding: 0;

    box-sizing: border-box;

}

html {

    scroll-behavior: smooth;

    scroll-padding-top: 80px;

}

body {

    margin: 0;

    padding: 0;

    min-height: 100vh;

    font-family: @font-primary;

    line-height: 1.6;

    color: @text-color;

    background: linear-gradient(135deg, #fff5f5 0%, #fff 100%);

    background-attachment: fixed;

}

*// Общие стили для форм*

.form-group {

    position: relative;

    margin-bottom: 1.5rem;

    input {

        width: 100%;

        padding: 0.8rem 1rem;

        border: 1px solid #ddd;

        border-radius: 8px;

        font-size: 0.9rem;

        box-sizing: border-box;

        background-color: #fff;

        color: #333;

        &:focus {

            outline: none;

            border-color: @primary-color;

            box-shadow: 0 0 0 2px rgba(255, 107, 107, 0.1);

        }

        &::placeholder {

            color: #999;

            opacity: 1;

        }

    }

}

*// Общие стили для кнопок*

.button {

    padding: 0.8rem 2rem;

    background: @primary-color;

    color: white;

    border: none;

    border-radius: 8px;

    font-size: 1rem;

    font-weight: 500;

    cursor: pointer;

    transition: background 0.3s ease;

    &:hover {

        background: darken(@primary-color, 5%);

    }

}

*// Стили для навигационной панели*

.navbar {

    position: fixed;

    top: 0;

    left: 0;

    width: 100%;

    background: white;

    padding: 1rem 5%;

    display: flex;

    justify-content: space-between;

    align-items: center;

    box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);

    z-index: 1000;

    .logo {

        font-family: @font-secondary;

        font-size: 1.8rem;

        font-weight: 700;

        color: @text-color;

        text-decoration: none;

    }

    .nav-links {

        display: flex;

        gap: 2rem;

        align-items: center;

        a {

            color: @text-color;

            text-decoration: none;

            font-weight: 500;

            transition: color 0.3s ease;

            position: relative;

            &:hover {

                color: @primary-color;

            }

            &.active {

                color: @primary-color;

                &:after {

                    content: '';

                    position: absolute;

                    bottom: -4px;

                    left: 0;

                    width: 100%;

                    height: 2px;

                    background-color: @primary-color;

                }

            }

        }

    }

}

*// Общие стили для карточек*

.card {

    background: white;

    border-radius: 12px;

    padding: 2rem;

    box-shadow: 0 5px 15px rgba(0, 0, 0, 0.1);

}

*// Общие стили заголовков*

h1, h2, h3, h4, h5, h6 {

    font-family: @font-secondary;

    color: @text-color;

    margin-bottom: 1rem;

}

*// Общие отступы контейнера*

.container {

    padding: 2rem 5%;

    padding-top: 100px;

}

*// Общие стили для подвала*

.footer {

    text-align: center;

    padding: 20px;

    background-color: #f8f8f8;

    font-size: 14px;

    color: #333;

}

**Листинг программы файла auth.js**

document.addEventListener('DOMContentLoaded', () => {

*// Создаем экземпляр Vue*

    const app = new Vue({

        el: '#app',

        data: {

            loginEmail: '',

            loginPassword: '',

            rememberMe: false,

            registerName: '',

            registerEmail: '',

            registerPassword: '',

            confirmPassword: '',

        },

        methods: {

**switchForm**(*formId*) {

*// Убираем класс active у всех вкладок и форм*

                document.querySelectorAll('.auth-tab').forEach(*tab* => *tab*.classList.remove('active'));

                document.querySelectorAll('.auth-form').forEach(*form* => *form*.classList.remove('active'));

*// Добавляем класс active нужной вкладке и форме*

                document.querySelector(`[data-tab="${*formId*}"]`).classList.add('active');

                document.getElementById(`${*formId*}Form`).classList.add('active');

            },

**handleLogin**(*event*) {

*event*.preventDefault();

                const email = this.loginEmail; *// Use Vue data property*

                const password = this.loginPassword; *// Use Vue data property*

                $.ajax({

                    url: '/api/users/login',

                    method: 'POST',

                    contentType: 'application/json',

                    data: JSON.stringify({

                        login: email,

                        password: password

                    }),

**success**: (*data*) => {

*// Сохраняем данные пользователя в localStorage*

                        localStorage.setItem('currentUser', JSON.stringify(*data*.user));

*// Запоминаем email и пароль, если выбрано*

                        if (this.rememberMe) {

                            localStorage.setItem('rememberedEmail', email);

                            localStorage.setItem('rememberedPassword', password);

                        } else {

                            localStorage.removeItem('rememberedEmail');

                            localStorage.removeItem('rememberedPassword');

                        }

*// Успешный вход - перенаправляем на главную страницу*

                        window.location.href = 'ind.html';

                    },

**error**: (*xhr*) => {

                        const data = *xhr*.responseJSON;

                        alert(data.error || 'Ошибка при входе');

                    }

                });

            },

**handleRegister**(*event*) {

*event*.preventDefault();

                const name = this.registerName;

                const email = this.registerEmail;

                const password = this.registerPassword;

                const confirmPassword = this.confirmPassword;

*// Проверяем совпадение паролей*

                if (password !== confirmPassword) {

                    alert('Пароли не совпадают');

                    return;

                }

                $.ajax({

                    url: '/api/users/register',

                    method: 'POST',

                    contentType: 'application/json',

                    data: JSON.stringify({

                        login: email,

                        username: name,

                        password: password,

                        role: 'user'

                    }),

**success**: (*data*) => {

                        alert('Регистрация успешна! Теперь вы можете войти в систему.');

                        this.switchForm('login');

                        document.getElementById('registerForm').reset();

                    },

**error**: (*xhr*) => {

                        const data = *xhr*.responseJSON;

                        alert(data.error || 'Ошибка при регистрации');

                    }

                });

            }

        },

**mounted**() {

*// Добавляем обработчики для переключения вкладок*

            document.querySelectorAll('.auth-tab').forEach(*tab* => {

                tab.addEventListener('click', () => this.switchForm(tab.getAttribute('data-tab')));

            });

        }

    });

});

*// Функция для переключения между формами*

function switchForm(*formId*) {

*// Убираем класс active у всех вкладок и форм*

    document.querySelectorAll('.auth-tab').forEach(*tab* => tab.classList.remove('active'));

    document.querySelectorAll('.auth-form').forEach(*form* => form.classList.remove('active'));

*// Добавляем класс active нужной вкладке и форме*

    document.querySelector(`[data-tab="${formId}"]`).classList.add('active');

    document.getElementById(`${formId}Form`).classList.add('active');

}

*// Функция для входа пользователя*

function handleLogin(*event*) {

    event.preventDefault();

    const email = document.getElementById('loginEmail').value;

    const password = document.getElementById('loginPassword').value;

    const rememberMe = document.getElementById('rememberMe').checked;

    $.ajax({

        url: '/api/users/login',

        method: 'POST',

        contentType: 'application/json',

        data: JSON.stringify({

            login: email,

            password: password

        }),

        success: function(*data*) {

*// Сохраняем данные пользователя в localStorage*

            localStorage.setItem('currentUser', JSON.stringify(data.user));

*// Запоминаем email и пароль, если выбрано*

            if (rememberMe) {

                localStorage.setItem('rememberedEmail', email);

                localStorage.setItem('rememberedPassword', password);

            } else {

                localStorage.removeItem('rememberedEmail');

                localStorage.removeItem('rememberedPassword');

            }

*// Успешный вход - перенаправляем на главную страницу*

            window.location.href = 'ind.html';

        },

        error: function(*xhr*) {

            const data = xhr.responseJSON;

            alert(data.error || 'Ошибка при входе');

        }

    });

}

*// Функция для регистрации пользователя*

function handleRegister(*event*) {

    event.preventDefault();

    const name = document.getElementById('registerName').value;

    const email = document.getElementById('registerEmail').value;

    const password = document.getElementById('registerPassword').value;

    const confirmPassword = document.getElementById('confirmPassword').value;

*// Проверяем совпадение паролей*

    if (password !== confirmPassword) {

        alert('Пароли не совпадают');

        return;

    }

    $.ajax({

        url: '/api/users/register',

        method: 'POST',

        contentType: 'application/json',

        data: JSON.stringify({

            login: email,

            username: name,

            password: password,

            role: 'user'

        }),

        success: function(*data*) {

            alert('Регистрация успешна! Теперь вы можете войти в систему.');

            switchForm('login');

            document.getElementById('registerForm').reset();

        },

        error: function(*xhr*) {

            const data = xhr.responseJSON;

            alert(data.error || 'Ошибка при регистрации');

        }

    });

}

*// Инициализация при загрузке страницы*

document.addEventListener('DOMContentLoaded', function() {

*// Добавляем обработчики для переключения вкладок*

    document.querySelectorAll('.auth-tab').forEach(*tab* => {

        tab.addEventListener('click', () => switchForm(tab.getAttribute('data-tab')));

    });

*// Добавляем обработчик события для формы входа*

    const loginForm = document.getElementById('loginForm');

    if (loginForm) {

        loginForm.addEventListener('submit', handleLogin);

    }

*// Добавляем обработчик события для формы регистрации*

    const registerForm = document.getElementById('registerForm');

    if (registerForm) {

        registerForm.addEventListener('submit', handleRegister);

    }

});

**Листинг программы файла catalog.js**

*// Создаем экземпляр Vue*

document.addEventListener('DOMContentLoaded', () => {

*// Создаем экземпляр Vue*

    const app = new Vue({

        el: '#app',

        data: {

            products: [],

            searchQuery: '',

            selectedCategories: [],

            sortOption: 'nameAsc',

            categories: ['Круассаны', 'Торты', 'Печенье', 'Другое'], *// Example categories*

        },

        computed: {

**filteredProducts**() {

                let filtered = this.products;

*// Фильтрация по названию*

                if (this.searchQuery) {

                    filtered = filtered.filter(*product* =>

*product*.name.toLowerCase().includes(this.searchQuery.toLowerCase())

                    );

                }

*// Фильтрация по категориям*

                if (this.selectedCategories.length > 0) {

                    filtered = filtered.filter(*product* =>

                        this.selectedCategories.includes(*product*.category)

                    );

                }

*// Сортировка*

                filtered.sort((*a*, *b*) => {

                    switch (this.sortOption) {

                        case 'nameAsc':

                            return *a*.name.localeCompare(*b*.name);

                        case 'nameDesc':

                            return *b*.name.localeCompare(*a*.name);

                        case 'priceAsc':

                            return *a*.price - *b*.price;

                        case 'priceDesc':

                            return *b*.price - *a*.price;

                        default:

                            return 0;

                    }

                });

                return filtered;

            }

        },

        methods: {

**loadProducts**() {

                $.ajax({

                    url: '/api/products',

                    method: 'GET',

**success**: (*data*) => {

                        this.products = *data*;

                    },

**error**: (*xhr*, *status*, *error*) => {

                        console.error('Ошибка загрузки продуктов:', *error*);

                    }

                });

            },

**handleAddToCart**(*product*) {

*// Implement add to cart functionality*

            },

**showEditProductModal**(*product*) {

*// Implement show edit modal functionality*

            },

**deleteProduct**(*productId*) {

                $.ajax({

                    url: `/api/products/${*productId*}`,

                    method: 'DELETE',

**success**: () => {

                        console.log('Продукт удален');

                        this.loadProducts(); *// Обновляем список продуктов после удаления*

                    },

**error**: (*xhr*, *status*, *error*) => {

                        console.error('Ошибка удаления продукта:', *error*);

                    }

                });

            },

**isAdmin**() {

                const currentUser = JSON.parse(localStorage.getItem('currentUser'));

                return currentUser && currentUser.role === 'admin';

            },

**updateFilters**() {

*// This method can be used to trigger updates if needed*

            }

        },

**mounted**() {

            this.loadProducts(); *// Load products when the component is mounted*

        }

    });

});

*// Загрузка продуктов с сервера*

function loadProducts() {

    $.ajax({

        url: '/api/products',

        method: 'GET',

        success: function(*data*) {

            window.products = data;

            window.renderProducts();

        },

        error: function(*xhr*, *status*, *error*) {

            console.error('Ошибка загрузки продуктов:', error);

        }

    });

}

*// Функция для проверки, является ли текущий пользователь администратором*

*/\*\**

*\* Проверяет, является ли текущий пользователь администратором.*

*\* Возвращает true, если пользователь является администратором, и false в противном случае.*

*\*/*

function isAdmin() {

    const currentUser = JSON.parse(localStorage.getItem('currentUser'));

    return currentUser && currentUser.role === 'admin';

}

*// Функция для отображения продуктов - переместить в глобальную область видимости*

*/\*\**

*\* Отображает список продуктов на странице.*

*\* Принимает три параметра: searchQuery - строка поиска, selectedCategories - массив выбранных категорий, sortOption - вариант сортировки.*

*\*/*

window.renderProducts = function(*searchQuery* = '', *selectedCategories* = [], *sortOption* = 'nameAsc') {

    const productGrid = document.querySelector('.product-grid');

    let filteredProducts = window.products;

*// Применить фильтр поиска*

    if (searchQuery) {

        filteredProducts = filteredProducts.filter(*product* =>

            product.name.toLowerCase().includes(searchQuery.toLowerCase())

        );

    }

*// Применить фильтр по категории*

    if (selectedCategories.length > 0) {

        filteredProducts = filteredProducts.filter(*product* =>

            selectedCategories.includes(product.category)

        );

    }

*// Применить сортировку*

    filteredProducts.sort((*a*, *b*) => {

*// Определяем порядок сортировки в зависимости от выбранного варианта*

        switch (sortOption) {

            case 'nameAsc':

                return a.name.localeCompare(b.name);

            case 'nameDesc':

                return b.name.localeCompare(a.name);

            case 'priceAsc':

                return a.price - b.price;

            case 'priceDesc':

                return b.price - a.price;

            default:

                return 0;

        }

    });

*// Генерируем HTML-код для отображения продуктов*

    productGrid.innerHTML = filteredProducts.map(*product* => `

        <div class="product-card">

            <img src="${product.image}" alt="${product.name}" class="product-image">

            <div class="product-info">

                <div class="product-title">

                    <h3 class="product-name">${product.name}</h3>

                    ${product.isPopular ? '<i class="fas fa-star star-icon"></i>' : ''}

                </div>

                <span class="category-tag">${product.category}</span>

                <div class="product-bottom">

                    <p class="product-price">${product.price}₽</p>

                    <button class="add-to-cart" onclick="handleAddToCart(${JSON.stringify(product).replace(/"/g, '&quot;')})">

                        <i class="fas fa-shopping-cart"></i>

                    </button>

                </div>

            </div>

            ${isAdmin() ? `

                <button class="edit-button" onclick='showEditProductModal(${JSON.stringify(product).replace(/'/g, "\'")})'>

                    <i class="fas fa-edit"></i>

                    Изменить

                </button>

                <button class="delete-button" onclick='deleteProduct(${product.id})'>

                    <i class="fas fa-trash"></i>

                    Удалить

                </button>

            ` : ''}

        </div>

    `).join('');

}

*// Функция для отображения модального окна редактирования продукта*

*/\*\**

*\* Отображает модальное окно редактирования продукта.*

*\* Принимает объект продукта как параметр.*

*\*/*

function showEditProductModal(*product*) {

    const modal = document.getElementById('editProductModal');

    const form = document.getElementById('editProductForm');

*// Заполните форму данными продукта*

    document.getElementById('editProductId').value = product.id;

    document.getElementById('editProductName').value = product.name;

    document.getElementById('editProductImage').value = product.image;

    document.getElementById('editProductPrice').value = product.price;

    document.getElementById('editProductCategory').value = product.category;

    document.getElementById('editProductPopular').checked = product.isPopular;

    modal.style.display = 'block';

}

*// Функция для скрытия модального окна редактирования продукта*

*/\*\**

*\* Скрыть модальное окно редактирования продукта.*

*\*/*

function hideEditProductModal() {

    const modal = document.getElementById('editProductModal');

    modal.style.display = 'none';

}

*// Функция для сохранения отредактированного продукта*

*/\*\**

*\* Сохраняет отредактированный продукт.*

*\* Принимает событие отправки формы как параметр.*

*\*/*

function saveEditedProduct(*event*) {

    event.preventDefault();

    const productId = document.getElementById('editProductId').value;

    const name = document.getElementById('editProductName').value;

    const image = document.getElementById('editProductImage').value;

    const price = Number(document.getElementById('editProductPrice').value);

    const category = document.getElementById('editProductCategory').value;

    const isPopular = document.getElementById('editProductPopular').checked;

*// Удаляем старый продукт перед сохранением нового*

    $.ajax({

        url: `/api/products/${productId}`,

        method: 'DELETE',

        success: function() {

            console.log('Старый продукт удален');

*// Теперь сохраняем новый продукт*

            const newProduct = {

                id: Number(productId),

                name,

                image,

                price,

                category,

                isPopular

            };

*// Отправка обновленного продукта на сервер*

            $.ajax({

                url: '/api/products',

                method: 'POST',

                contentType: 'application/json',

                data: JSON.stringify(newProduct),

                success: function(*data*) {

                    console.log('Продукт обновлен:', data);

*// Повторно отобразить продукты*

                    window.renderProducts();

*// Обновите страницу после изменения продукта*

                    location.reload();

*// Скрыть модальное окно*

                    hideEditProductModal();

                },

                error: function(*xhr*, *status*, *error*) {

                    console.error('Ошибка обновления продукта:', error);

                }

            });

        },

        error: function(*xhr*, *status*, *error*) {

            console.error('Ошибка удаления старого продукта');

        }

    });

}

*// Функция для удаления продукта*

*/\*\**

*\* Удаляет продукт.*

*\* Принимает идентификатор продукта как параметр.*

*\*/*

function deleteProduct(*productId*) {

    $.ajax({

        url: `/api/products/${productId}`,

        method: 'DELETE',

        success: () => {

            console.log('Продукт удален');

            this.loadProducts(); *// Обновляем список продуктов после удаления*

        },

        error: (*xhr*, *status*, *error*) => {

            console.error('Ошибка удаления продукта:', error);

        }

    });

}

*// Определить addNewProduct в глобальной области видимости*

*/\*\**

*\* Добавляет новый продукт.*

*\*/*

window.addNewProduct = async function() {

*// Получаем данные из формы*

    const name = document.getElementById('productName').value.trim();

    const price = Number(document.getElementById('productPrice').value);

    const image = document.getElementById('productImage').value.trim();

    const category = document.getElementById('productCategory').value;

    const isPopular = document.getElementById('productIsPopular').checked;

*// Проверяем, заполнены ли все поля*

    if (!name || !price || !image || !category) {

        alert('Пожалуйста, заполните все поля');

        return;

    }

*// Создаем новый продукт*

    const newProduct = {

        id: window.products.length ? Math.max(...window.products.map(*p* => p.id)) + 1 : 1,

        name,

        price,

        image,

        category,

        isPopular

    };

*// Отправка нового продукта на сервер*

    $.ajax({

        url: '/api/products',

        method: 'POST',

        contentType: 'application/json',

        data: JSON.stringify(newProduct),

        success: function(*data*) {

            window.products.push(newProduct);

            window.renderProducts();

            alert('Продукт успешно добавлен!');

        },

        error: function(*xhr*, *status*, *error*) {

            console.error('Ошибка добавления продукта:', error);

        }

    });

}

*// Сделать функцию глобальной*

window.hideAddProductForm = function() {

    document.getElementById('addProductForm').style.display = 'none';

};

*// Инициализация корзины в глобальном контексте*

window.cart = JSON.parse(localStorage.getItem('cart') || '[]');

*// Функция для обновления количества товаров в корзине*

window.updateCartCount = function() {

    const cartCount = document.querySelector('.cart-count');

    if (cartCount) {

        cartCount.textContent = window.cart.reduce((*total*, *item*) => total + item.quantity, 0);

    }

}

*// Функция для обновления количества товара в корзине*

window.updateCartItemQuantity = function(*id*, *quantity*) {

    const item = window.cart.find(*item* => item.id === id);

    if (item) {

        item.quantity = quantity;

        if (item.quantity <= 0) {

            window.cart = window.cart.filter(*item* => item.id !== id);

        }

        localStorage.setItem('cart', JSON.stringify(window.cart));

        window.updateCart();

        window.updateCartCount();

    }

}

*// Обновление отображения корзины*

window.updateCart = function() {

    const cartItems = document.querySelector('.cart-items');

    const totalAmount = document.querySelector('.total-amount');

    if (!cartItems || !totalAmount) return;

    cartItems.innerHTML = window.cart.map(*item* => `

        <div class="cart-item">

            <span>${item.name} x <input type="number" value="${item.quantity}" min="0" data-id="${item.id}" class="quantity-input"></span>

            <span>${item.price \* item.quantity}₽</span>

            <button class="remove-item" data-id="${item.id}">×</button>

        </div>

    `).join('');

*// Добавляем обработчики событий для кнопок удаления и изменения количества*

    document.querySelectorAll('.remove-item').forEach(*button* => {

        button.addEventListener('click', () => {

            const id = parseInt(button.dataset.id);

            window.cart = window.cart.filter(*item* => item.id !== id);

            localStorage.setItem('cart', JSON.stringify(window.cart));

            window.updateCart();

            window.updateCartCount();

        });

    });

    document.querySelectorAll('.quantity-input').forEach(*input* => {

        input.addEventListener('change', (*e*) => {

            const id = parseInt(e.target.dataset.id);

            const quantity = parseInt(e.target.value);

            window.updateCartItemQuantity(id, quantity);

        });

    });

*// Обновляем общую сумму*

    const total = window.cart.reduce((*sum*, *item*) => sum + (item.price \* item.quantity), 0);

    totalAmount.textContent = total + '₽';

}

*// Добавление товара в корзину*

window.handleAddToCart = function(*product*) {

    const currentUser = localStorage.getItem('currentUser');

    if (!currentUser) {

        alert('Пожалуйста, войдите в систему, чтобы добавить товары в корзину');

        window.location.href = 'auth.html';

        return;

    }

    const existingItem = window.cart.find(*item* => item.id === product.id);

    if (existingItem) {

        existingItem.quantity += 1;

    } else {

        window.cart.push({ ...product, quantity: 1 });

    }

    localStorage.setItem('cart', JSON.stringify(window.cart));

    window.updateCartCount();

    window.updateCart();

}

document.addEventListener('DOMContentLoaded', () => {

*// Загружаем продукты*

    loadProducts();

*// Проверяем статус входа и обновляем UI*

    const currentUser = localStorage.getItem('currentUser');

    const logoutButton = document.getElementById('logoutButton');

    const loginLink = document.querySelector('.login-link');

*// Обновляем навигацию в зависимости от статуса входа*

    if (currentUser) {

        if (loginLink) loginLink.style.display = 'none';

        if (logoutButton) logoutButton.style.display = 'inline-block';

*// Добавляем отображение имени пользователя*

        const userInfo = JSON.parse(currentUser);

        const userNameDisplay = document.createElement('span');

        userNameDisplay.className = 'user-name';

        userNameDisplay.textContent = userInfo.name;

        document.querySelector('.cart-container').prepend(userNameDisplay);

    } else {

        if (loginLink) loginLink.style.display = 'inline-block';

        if (logoutButton) logoutButton.style.display = 'none';

    }

*// Добавляем обработчик выхода*

    if (logoutButton) {

        logoutButton.addEventListener('click', (*e*) => {

            e.preventDefault();

            localStorage.removeItem('currentUser');

            window.location.reload();

        });

    }

*// Корзина*

    const cartIcon = document.querySelector('.cart-icon');

    const cartModal = document.getElementById('cartModal');

    if (cartIcon) {

        cartIcon.addEventListener('click', () => {

            if (!currentUser) {

                alert('Пожалуйста, войдите в систему, чтобы просмотреть корзину');

                window.location.href = 'auth.html';

                return;

            }

            if (cartModal) {

                cartModal.style.display = cartModal.style.display === 'block' ? 'none' : 'block';

            }

        });

    }

*// Закрытие корзины*

    if (cartModal) {

        cartModal.addEventListener('click', (*e*) => {

            if (e.target === cartModal) {

                cartModal.style.display = 'none';

            }

        });

    }

*// Проверка на админа*

    function isAdmin() {

        const currentUser = JSON.parse(localStorage.getItem('currentUser'));

        return currentUser && currentUser.role === 'admin';

    }

*// Добавление UI админа, если пользователь является админом*

    function setupAdminUI() {

        const catalogSection = document.querySelector('.catalog-section');

        if (!catalogSection) {

            console.error('Catalog section not found');

            return;

        }

        if (isAdmin()) {

            const adminSection = document.createElement('div');

            adminSection.className = 'admin-section';

            adminSection.innerHTML = `

                <button id="addProductBtn" class="btn btn-primary">

                    <i class="fas fa-plus"></i>

                    Добавить новый продукт

                </button>

                <div id="addProductForm" style="display: none;" class="add-product-form">

                    <h2>Добавить новый продукт</h2>

                    <div class="form-group">

                        <label for="productName">Название продукта</label>

                        <input type="text" id="productName" placeholder="Название продукта" required>

                    </div>

                    <div class="form-group">

                        <label for="productPrice">Цена</label>

                        <input type="number" id="productPrice" placeholder="Цена" required>

                    </div>

                    <div class="form-group">

                        <label for="productImage">URL изображения</label>

                        <input type="text" id="productImage" placeholder="URL изображения" required>

                    </div>

                    <div class="form-group">

                        <label for="productCategory">Категория</label>

                        <select id="productCategory" required>

                            <option value="Торты">Торты</option>

                            <option value="Круассаны">Круассаны</option>

                            <option value="Печенье">Печенье</option>

                            <option value="Другое">Другое</option>

                        </select>

                    </div>

                    <div class="checkbox-group">

                        <input type="checkbox" id="productIsPopular">

                        <label for="productIsPopular">Популярный продукт</label>

                    </div>

                    <div class="form-buttons">

                        <button onclick="addNewProduct()" class="add-product-button">

                            <i class="fas fa-plus"></i>

                            Сохранить

                        </button>

                        <button onclick="hideAddProductForm()" class="cancel-button">

                            <i class="fas fa-times"></i>

                            Отмена

                        </button>

                    </div>

                </div>

                <div id="editProductModal" style="display: none;" class="edit-product-modal">

                    <form id="editProductForm">

                        <input type="hidden" id="editProductId">

                        <label for="editProductName">Название продукта:</label>

                        <input type="text" id="editProductName" required>

                        <label for="editProductImage">URL изображения:</label>

                        <input type="text" id="editProductImage" required>

                        <label for="editProductPrice">Цена:</label>

                        <input type="number" id="editProductPrice" required>

                        <label for="editProductCategory">Категория:</label>

                        <select id="editProductCategory" required>

                            <option value="Торты">Торты</option>

                            <option value="Круассаны">Круассаны</option>

                            <option value="Печенье">Печенье</option>

                            <option value="Другое">Другое</option>

                        </select>

                        <label>

                            <input type="checkbox" id="editProductPopular"> Популярный продукт

                        </label>

                        <div class="form-buttons">

                            <button type="submit" class="btn btn-success">Сохранить</button>

                            <button type="button" onclick="hideEditProductModal()" class="btn btn-secondary">Отмена</button>

                        </div>

                    </form>

                </div>

            `;

            catalogSection.insertBefore(adminSection, catalogSection.firstChild);

            document.getElementById('addProductBtn').addEventListener('click', () => {

                document.getElementById('addProductForm').style.display = 'block';

            });

        }

    }

*// Поиск*

    const searchInput = document.getElementById('search-input');

    searchInput.addEventListener('input', updateFilters);

*// Сортировка*

    const sortSelect = document.getElementById('sort-select');

    sortSelect.addEventListener('change', updateFilters);

*// Фильтры*

    const categoryCheckboxes = document.querySelectorAll('input[name="category"]');

    categoryCheckboxes.forEach(*checkbox* => {

        checkbox.addEventListener('change', updateFilters);

    });

    function updateFilters() {

        const searchQuery = searchInput.value;

        const selectedCategories = Array.from(categoryCheckboxes)

            .filter(*cb* => cb.checked)

            .map(*cb* => cb.value);

        const sortOption = sortSelect.value;

        window.renderProducts(searchQuery, selectedCategories, sortOption);

    }

*// Вызов функции админа*

    setupAdminUI();

*// Добавление UI админа, если пользователь является админом*

    const editForm = document.getElementById('editProductForm');

    if (editForm) {

        editForm.addEventListener('submit', saveEditedProduct);

    }

*// Рендер*

    window.updateCartCount();

    window.updateCart();

});

let isOrderInProgress = false; *// Флаг для предотвращения повторного вызова*

*// Функция для оформления заказа*

function placeOrder() {

    if (isOrderInProgress) return; *// Если заказ уже в процессе, выходим*

    isOrderInProgress = true; *// Устанавливаем флаг*

    const currentUser = JSON.parse(localStorage.getItem('currentUser'));

    if (!currentUser) {

        alert('Пожалуйста, войдите в систему, чтобы оформить заказ');

        window.location.href = 'auth.html';

        return;

    }

*// Проверка на пустую корзину*

    if (window.cart.length === 0) {

        alert('Ваша корзина пуста. Пожалуйста, добавьте товары перед оформлением заказа.');

        isOrderInProgress = false; *// Сбрасываем флаг*

        return;

    }

    const orderDetails = {

        userId: currentUser.id,

        items: window.cart,

        totalAmount: window.cart.reduce((*total*, *item*) => total + (item.price \* item.quantity), 0),

        orderDate: new Date().toISOString()

    };

*// Отправка заказа на сервер*

    $.ajax({

        url: '/api/orders',

        method: 'POST',

        contentType: 'application/json',

        data: JSON.stringify(orderDetails),

        success: function(*response*) {

            alert(`Заказ оформлен! Ваш номер заказа: ${response.order\_id}`);

            window.cart = []; *// Очистка корзины после оформления*

            localStorage.setItem('cart', JSON.stringify(window.cart));

            window.updateCartCount();

            window.updateCart();

        },

        error: function(*xhr*, *status*, *error*) {

            console.error('Ошибка оформления заказа:', error);

            alert('Произошла ошибка при оформлении заказа. Пожалуйста, попробуйте еще раз.');

        },

        complete: function() {

            isOrderInProgress = false; *// Сбрасываем флаг после завершения*

        }

    });

}

*// Добавьте обработчик события для кнопки "Оформить заказ"*

document.addEventListener('DOMContentLoaded', () => {

    const checkoutButton = document.querySelector('.checkout-button');

    if (checkoutButton) {

        checkoutButton.addEventListener('click', (*event*) => {

            event.preventDefault(); *// Предотвращаем двойное срабатывание*

            placeOrder();

        });

    }

});

**Листинг программы файла ind.js**

document.addEventListener('DOMContentLoaded', () => {

*// Создаем экземпляр Vue*

    const app = new Vue({

        el: '#app',

        data: {

            products: [],

            cart: JSON.parse(localStorage.getItem('cart')) || [],

        },

        methods: {

            loadProducts() {

                $.ajax({

                    url: '/api/products',

                    method: 'GET',

                    success: (*data*) => {

                        this.products = data;

                    },

                    error: (*xhr*, *status*, *error*) => {

                        console.error('Ошибка загрузки продуктов:', error);

                    }

                });

            },

            handleAddToCart(*product*) {

                const existingItem = this.cart.find(*item* => item.id === product.id);

                if (existingItem) {

                    existingItem.quantity += 1;

                } else {

                    this.cart.push({ ...product, quantity: 1 });

                }

                localStorage.setItem('cart', JSON.stringify(this.cart));

                window.updateCartCount();

            },

            updateCartCount() {

                const cartCount = document.querySelector('.cart-count');

                if (cartCount) {

                    cartCount.textContent = this.cart.reduce((*total*, *item*) => total + item.quantity, 0);

                }

            },

        },

        mounted() {

            this.loadProducts(); *// Load products when the component is mounted*

            this.updateCartCount(); *// Update cart count on load*

        }

    });

});

*// Загружаем продукты из localStorage*

window.products = JSON.parse(localStorage.getItem('products')) || [];

*// Инициализируем корзину в глобальной области*

window.cart = JSON.parse(localStorage.getItem('cart') || '[]');

*// Функция обновления количества товаров в корзине - глобальная область*

window.updateCartCount = function() {

    const cartCount = document.querySelector('.cart-count');

    if (cartCount) {

        cartCount.textContent = window.cart.reduce((*total*, *item*) => total + item.quantity, 0);

    }

}

*// Функция для обновления количества товара в корзине*

window.updateCartItemQuantity = function(*id*, *quantity*) {

    const item = window.cart.find(*item* => item.id === id);

    if (item) {

        item.quantity = quantity;

        if (item.quantity <= 0) {

            window.cart = window.cart.filter(*item* => item.id !== id);

        }

        localStorage.setItem('cart', JSON.stringify(window.cart));

        window.updateCart();

        window.updateCartCount();

    }

}

*// Обновление отображения корзины*

window.updateCart = function() {

    const cartItems = document.querySelector('.cart-items');

    const totalAmount = document.querySelector('.total-amount');

    if (!cartItems || !totalAmount) return;

    cartItems.innerHTML = window.cart.map(*item* => `

        <div class="cart-item">

            <span>${item.name} x <input type="number" value="${item.quantity}" min="0" data-id="${item.id}" class="quantity-input"></span>

            <span>${item.price \* item.quantity}₽</span>

            <button class="remove-item" data-id="${item.id}">×</button>

        </div>

    `).join('');

*// Добавляем обработчики событий для кнопок удаления и изменения количества*

    document.querySelectorAll('.remove-item').forEach(*button* => {

        button.addEventListener('click', () => {

            const id = parseInt(button.dataset.id);

            window.cart = window.cart.filter(*item* => item.id !== id);

            localStorage.setItem('cart', JSON.stringify(window.cart));

            window.updateCart();

            window.updateCartCount();

        });

    });

    document.querySelectorAll('.quantity-input').forEach(*input* => {

        input.addEventListener('change', (*e*) => {

            const id = parseInt(e.target.dataset.id);

            const quantity = parseInt(e.target.value);

            window.updateCartItemQuantity(id, quantity);

        });

    });

*// Обновляем общую сумму*

    const total = window.cart.reduce((*sum*, *item*) => sum + (item.price \* item.quantity), 0);

    totalAmount.textContent = total + '₽';

}

*// Обработка добавления в корзину - глобальная область*

window.handleAddToCart = function(*product*) {

    const currentUser = localStorage.getItem('currentUser');

    if (!currentUser) {

        alert('Пожалуйста, войдите в систему, чтобы добавить товары в корзину');

        window.location.href = 'auth.html';

        return;

    }

    const existingItem = window.cart.find(*item* => item.id === product.id);

    if (existingItem) {

        existingItem.quantity += 1;

    } else {

        window.cart.push({ ...product, quantity: 1 });

    }

    localStorage.setItem('cart', JSON.stringify(window.cart));

    window.updateCartCount();

    window.updateCart();

}

*// Загружаем популярные продукты из JSON файла*

async function loadPopularProducts() {

    try {

        const response = await fetch('/data/products.json');

        const products = await response.json();

*// Фильтруем только популярные продукты*

        const popularProducts = products.filter(*product* => product.isPopular);

*// Получаем контейнер для сетки продуктов*

        const productGrid = document.querySelector('.product-grid');

        if (!productGrid) return;

*// Очищаем существующее содержимое*

        productGrid.innerHTML = '';

*// Добавляем каждый популярный продукт в сетку*

        popularProducts.forEach(*product* => {

            const productCard = document.createElement('div');

            productCard.className = 'product-card';

            productCard.innerHTML = `

                <img src="${product.image}" alt="${product.name}">

                <div class="product-info">

                    <span class="product-category">${product.category}</span>

                    <h3 class="product-name">${product.name}</h3>

                    <div class="product-price">${product.price} ₽</div>

                    <button class="add-to-cart" onclick="handleAddToCart(${JSON.stringify(product).replace(/"/g, '&quot;')})">

                        <i class="fas fa-shopping-cart"></i>

                    </button>

                </div>

            `;

            productGrid.appendChild(productCard);

        });

    } catch (error) {

        console.error('Ошибка при загрузке продуктов:', error);

        const productGrid = document.querySelector('.product-grid');

        if (productGrid) {

            productGrid.innerHTML = '<p>Ошибка при загрузке продуктов</p>';

        }

    }

}

document.addEventListener('scroll', function() {

    const sections = document.querySelectorAll('section');

    const navLinks = document.querySelectorAll('.nav-links a');

    sections.forEach(*section* => {

        const sectionTop = section.offsetTop;

        const sectionHeight = section.clientHeight;

        if (pageYOffset >= sectionTop - sectionHeight / 3) {

            navLinks.forEach(*link* => {

                link.classList.remove('active');

                if (link.getAttribute('href') === '#' + section.id) {

                    link.classList.add('active');

                }

            });

        }

    });

*// Проверка, если прокручиваемся к верхней части страницы*

    if (pageYOffset < sections[0].offsetTop) {

        navLinks.forEach(*link* => {

            link.classList.remove('active');

        });

        navLinks[0].classList.add('active'); *// Активируем "Главная"*

    }

});

document.addEventListener('DOMContentLoaded', function() {

*// Проверка статуса входа и обновление интерфейса*

    const currentUser = localStorage.getItem('currentUser');

    const logoutButton = document.getElementById('logoutButton');

    const loginLink = document.querySelector('.login-link');

*// Обновление навигации в зависимости от статуса входа*

    if (currentUser) {

        if (loginLink) loginLink.style.display = 'none';

        if (logoutButton) logoutButton.style.display = 'inline-block';

*// Добавляем отображение имени пользователя*

        const userInfo = JSON.parse(currentUser);

        const userNameDisplay = document.createElement('span');

        userNameDisplay.className = 'user-name';

        userNameDisplay.textContent = userInfo.name;

        document.querySelector('.cart-container').prepend(userNameDisplay);

    } else {

        if (loginLink) loginLink.style.display = 'inline-block';

        if (logoutButton) logoutButton.style.display = 'none';

    }

*// Добавляем обработчик выхода*

    if (logoutButton) {

        logoutButton.addEventListener('click', (*e*) => {

            e.preventDefault();

            localStorage.removeItem('currentUser');

            window.location.reload();

        });

    }

*// Функция для проверки, является ли текущий пользователь администратором*

    function isAdmin() {

        const currentUser = JSON.parse(localStorage.getItem('currentUser'));

        return currentUser && currentUser.role === 'admin';

    }

*// Функция рендеринга продуктов*

    function renderProducts() {

*// Рендеринг популярных продуктов*

        const popularProductsContainer = document.querySelector('#products .product-grid');

        if (popularProductsContainer) {

            const popularProducts = window.products.filter(*product* => product.isPopular);

            popularProducts.forEach(*product* => {

                const productCard = createProductCard(product);

                popularProductsContainer.appendChild(productCard);

            });

        }

*// Рендеринг продуктов каталога*

        const catalogContainer = document.querySelector('#catalog .products');

        if (catalogContainer) {

            window.products.forEach(*product* => {

                const productCard = createProductCard(product);

                catalogContainer.appendChild(productCard);

            });

        }

    }

*// Функция для создания карточки продукта*

    function createProductCard(*product*) {

        const card = document.createElement('div');

        card.className = 'product-card';

        card.innerHTML = `

            <img src="${product.image}" alt="${product.name}" class="product-image">

            <div class="product-info">

                <div class="product-title">

                    <h3 class="product-name">${product.name}</h3>

                    ${product.isPopular ? '<i class="fas fa-star star-icon"></i>' : ''}

                </div>

                <span class="category-tag">${product.category}</span>

                <div class="product-bottom">

                    <p class="product-price">${product.price}₽</p>

                    <button class="add-to-cart" onclick="handleAddToCart(${JSON.stringify(product).replace(/"/g, '&quot;')})">

                        <i class="fas fa-shopping-cart"></i>

                    </button>

                </div>

            </div>

        `;

        return card;

    }

*// Загружаем популярные продукты*

    loadPopularProducts();

*// Функциональность переключения корзины*

    const cartIcon = document.querySelector('.cart-icon');

    const cartModal = document.getElementById('cartModal');

    if (cartIcon) {

        cartIcon.addEventListener('click', () => {

            const currentUser = localStorage.getItem('currentUser');

            if (!currentUser) {

                alert('Пожалуйста, войдите в систему, чтобы просмотреть корзину');

                window.location.href = 'auth.html';

                return;

            }

            if (cartModal) {

                cartModal.style.display = cartModal.style.display === 'block' ? 'none' : 'block';

            }

        });

    }

*// Закрытие корзины*

    if (cartModal) {

        cartModal.addEventListener('click', (*e*) => {

            if (e.target === cartModal) {

                cartModal.style.display = 'none';

            }

        });

    }

*// Инициализация корзины*

    window.updateCartCount();

    window.updateCart();

*// Функциональность контактного формы*

    const contactForm = document.querySelector('.contact-form');

    if (contactForm) {

        contactForm.addEventListener('submit', (*e*) => {

*e*.preventDefault();

            const formData = {

                name: document.getElementById('name').value,

                email: document.getElementById('email').value,

                phone: document.getElementById('phone').value,

                message: document.getElementById('message').value

            };

*// Отправка данных на сервер*

            console.log('Form submitted:', formData);

            alert('Спасибо за ваше сообщение! Мы свяжемся с вами в ближайшее время.');

            contactForm.reset();

        });

    }

*// Плавная прокрутка к якорям*

    document.querySelectorAll('a[href^="#"]').forEach(*anchor* => {

*anchor*.addEventListener('click', function(*e*) {

*e*.preventDefault();

            const targetId = this.getAttribute('href');

            if (targetId === '#') return; *// Пропускаем клик на #*

            const targetElement = document.querySelector(targetId);

            if (!targetElement) return; *// Пропускаем несуществующий элемент*

            const navbarHeight = document.querySelector('.navbar').offsetHeight;

            const targetPosition = targetElement.getBoundingClientRect().top + window.pageYOffset;

            const offsetPosition = targetPosition - navbarHeight;

            window.scrollTo({

                top: offsetPosition,

                behavior: 'smooth'

            });

        });

    });

});

let isOrderInProgress = false; *// Флаг для предотвращения повторного вызова*

*// Функция для оформления заказа*

function **placeOrder**() {

    if (isOrderInProgress) return; *// Если заказ уже в процессе, выходим*

    isOrderInProgress = true; *// Устанавливаем флаг*

    const currentUser = JSON.parse(localStorage.getItem('currentUser'));

    if (!currentUser) {

        alert('Пожалуйста, войдите в систему, чтобы оформить заказ');

        window.location.href = 'auth.html';

        return;

    }

*// Проверка на пустую корзину*

    if (window.cart.length === 0) {

        alert('Ваша корзина пуста. Пожалуйста, добавьте товары перед оформлением заказа.');

        isOrderInProgress = false; *// Сбрасываем флаг*

        return;

    }

    const orderDetails = {

        userId: currentUser.id,

        items: window.cart,

        totalAmount: window.cart.reduce((*total*, *item*) => *total* + (*item*.price \* *item*.quantity), 0),

        orderDate: new Date().toISOString()

    };

*// Отправка заказа на сервер*

    $.ajax({

        url: '/api/orders',

        method: 'POST',

        contentType: 'application/json',

        data: JSON.stringify(orderDetails),

**success**: function(*response*) {

            alert(`Заказ оформлен! Ваш номер заказа: ${*response*.order\_id}`);

            window.cart = []; *// Очистка корзины после оформления*

            localStorage.setItem('cart', JSON.stringify(window.cart));

            window.updateCartCount();

            window.updateCart();

        },

**error**: function(*xhr*, *status*, *error*) {

            console.error('Ошибка оформления заказа:', *error*);

            alert('Произошла ошибка при оформлении заказа. Пожалуйста, попробуйте еще раз.');

        },

**complete**: function() {

            isOrderInProgress = false; *// Сбрасываем флаг после завершения*

        }

    });

}

*// Добавьте обработчик события для кнопки "Оформить заказ"*

document.addEventListener('DOMContentLoaded', () => {

    const checkoutButton = document.querySelector('.checkout-button');

    if (checkoutButton) {

        checkoutButton.addEventListener('click', (*event*) => {

*event*.preventDefault(); *// Предотвращаем двойное срабатывание*

            placeOrder();

        });

    }

});