

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Уральский федеральный университет
имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий – РТФ
Школа бакалавриата

ОТЧЕТ

По проекту
«Разработка платформы для создания интернет-магазинов»
по дисциплине «Проектный практикум»

Заказчик: Фамилия И.О.
Куратор: Фамилия И.О.
ученая степень, ученое звание, должность
Студенты команды Webers
Фамилия И.О.
Фамилия И.О.
Фамилия И.О.

Хрущков А.Е.
Хрущков А.Е.
Омаров Г.А.
Котлов К.А.
Гайсина Э.Б.

Екатеринбург, 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
Основная часть	5
ЗАКЛЮЧЕНИЕ	10

ВВЕДЕНИЕ

1. Цель и задачи проекта

Целью данного проекта является разработка современного программного продукта — маркетплейса для онлайн-продажи товаров. В рамках проекта были поставлены и решены следующие задачи:

- анализ требований к функционалу современного маркетплейса;
- проектирование архитектуры приложения и пользовательского интерфейса;
- реализация основных функций: регистрация и авторизация пользователей, каталог товаров, поиск, корзина, оформление заказов, личный кабинет пользователя;
- обеспечение безопасности хранения и обработки пользовательских данных;
- проведение тестирования и отладки приложения;
- подготовка сопроводительной документации.

2. Актуальность и важность проекта

В последние годы электронная коммерция стремительно развивается, и всё больше пользователей предпочитают совершать покупки онлайн. Озон является одним из лидеров российского рынка e-commerce, и его функционал служит эталоном для многих подобных сервисов. Создание клона такого приложения позволяет:

- изучить современные подходы к разработке масштабируемых и надёжных веб- и мобильных приложений;
- получить практический опыт реализации сложных пользовательских сценариев;
- оценить и внедрить лучшие практики в области UX/UI-дизайна, безопасности и производительности;
- подготовить платформу для дальнейших экспериментов и внедрения новых функций.

3. Область применения программного продукта

Разработанный программный продукт может быть использован в следующих областях:

- в качестве учебного и демонстрационного проекта для студентов и начинающих разработчиков;
- как основа для создания собственного интернет-магазина или маркетплейса;
- для тестирования и внедрения новых технологий и решений в сфере электронной коммерции;
- в качестве платформы для проведения хакатонов, соревнований и командных проектов.

4. Ожидаемые результаты и планируемые достижения

По завершении проекта планируется получить полнофункциональное приложение, обладающее следующими характеристиками:

- реализация ключевых функций оригинального приложения Озон;
- современный и интуитивно понятный пользовательский интерфейс;
- надёжная работа с пользовательскими данными и заказами;
- возможность масштабирования и дальнейшего развития продукта;
- подробная документация по установке, настройке и использованию приложения.

Реализация данного проекта позволит участникам команды значительно повысить уровень профессиональных компетенций, а также создать востребованный и практически значимый программный продукт.

Основная часть

Распределение задач (роли)

- Омаров Гасан Адамович – тимлид; осуществлял управление создания продукта. Также участвовал в разработке дизайна продукта.
- Котлов Климентий Андреевич – разработчик; реализовал сам продукт, с учётом функционала и дизайна.
- Гайсина Элина Булатовна – дизайнер; создала дизайн продукта в Figma.

Анализ аналогов продукта

1. **Shopify** – одна из самых популярных платформ для создания интернет-магазинов.
 - **что делает:** Позволяет быстро запустить интернет-магазин без навыков программирования, предлагает готовые темы, интеграции с платежными системами и маркетплейсами.
 - **где применяется:** Подходит для малого и среднего бизнеса, крупных брендов.
 - **плюсы:** Простота, высокая скорость развертывания, мощная экосистема приложений.
 - **минусы:** Высокие комиссии, ограниченная гибкость настройки без дополнительных модулей.
2. **WooCommerce** (плагин для WordPress)
 - **что делает:** Добавляет функционал интернет-магазина к сайтам на WordPress.
 - **где применяется:** Чаще используется малым бизнесом, блогерами и предпринимателями.
 - **плюсы:** Гибкость, бесплатное использование, большой выбор плагинов.
 - **минусы:** Требует технических знаний для настройки, низкая производительность на больших объемах данных.
3. **Magento (Adobe Commerce)**
 - **что делает:** Гибкая и мощная система для крупных eCommerce-проектов.
 - **где применяется:** Средний и крупный бизнес, корпоративные решения.
 - **плюсы:** Высокая кастомизация, масштабируемость.
 - **минусы:** Дорогая поддержка, сложность настройки и разработки.
4. **Tilda**

- **что делает:** Позволяет создать интернет-магазин с простым визуальным редактором.
- **где применяется:** Подходит для малого бизнеса, небольших брендов, дизайнеров.
- **плюсы:** Простота, хорошая адаптивность.
- **Минусы:** Ограниченный функционал, слабая кастомизация.

5. Wix eCommerce

- **что делает:** Дает возможность быстро создать магазин с помощью конструктора.
- **где применяется:** Малый бизнес, фрилансеры, креативные проекты.
- **плюсы:** Интуитивный интерфейс, встроенные маркетинговые инструменты.
- **минусы:** Ограниченнные возможности для расширения и настройки.

Обзор архитектуры программного продукта и основных компонентов

Фронтенд сайта делится на два основных модуля: общая страница и раздел продавца. Любой из пользователей при регистрации получает роль "покупатель", которая позже может быть изменена на "продавец". Общая страница доступна всем, а раздел продавца только для пользователей с ролью "продавец".

Общая страница включает в себя главную страницу с сеткой товаров, из которой можно попасть на страницы аккаунта, выбранных товаров, заказов, корзины и какого-либо товара.

На странице аккаунта содержится информация о пользователе (имя, телефон, почта, адреса доставки), которые также могут быть редактированы.

На странице выбранных товаров находятся отмеченные пользователем товары, которые он захочет в будущем купить.

На странице заказов отображаются актуальные и уже завершённые заказы с датой самого заказа, датой доставки и информации о заказанных товарах.

На странице корзины содержится информация о выбранных для покупки товарах (включая кол-во единиц каждого из них), а также общая стоимость корзины.

На странице товара можно посмотреть подробную информацию о нём (описание, количество на складе, отзывы и т.д.).

В разделе продавца имеется несколько страниц:

Страница с товарами продавца, каждый из которых можно редактировать, проверять и настраивать количество оставшихся единиц этого товара и состояние активности карточки.

Страница отзывов на магазин.

Страница заказов и информация о них, которой пользуются продавцы.

Страница финансов продавца.

Описание основных компонентов и связей между ними:

1. База данных (SQLite, файл sqlite.ts)

users — пользователи (id, username, email, password_hash, role и др.)

user_addresses — адреса пользователей

categories — категории товаров (с поддержкой иерархии через parent_category_id)

products — товары (seller_id, name, price, size, images, stock_quantity и др.)

product_category — связь многие-ко-многим между товарами и категориями

cart — корзина пользователя (user_id, product_id, quantity, selected)

favorites — избранные товары пользователя

orders и order_items — заказы и их содержимое

product_reviews, seller_reviews — отзывы о товарах и продавцах

2. Бэкенд (Bun, файл index.ts)

REST API для работы с пользователями, товарами, категориями, корзиной, избранным, заказами и отзывами.

Аутентификация через JWT (токен хранится на клиенте).

Работа с корзиной: добавление, удаление, изменение количества и статуса выбора товара.

Работа с товарами: получение списка, добавление, получение по id, фильтрация по продавцу.

Работа с категориями: получение всех, построение дерева, получение пути категории для товара.

Работа с избранным: добавление/удаление товара в избранное, получение избранных товаров пользователя.

Работа с заказами: оформление, получение истории заказов.

Работа с отзывами: добавление и получение отзывов к товарам и продавцам.

Работа с изображениями: хранение путей к файлам, отдача файлов по запросу.

3. Фронтенд (React, компоненты и страницы)

api-service.tsx — набор функций для общения с backend API (fetch-запросы).

Компоненты:

ProductContent — карточка товара, отображает инфу, отзывы, добавление в корзину/избранное.

BasketContent — корзина пользователя, изменение количества, удаление, выбор товаров.

Navbar, Catalog — навигация, дерево категорий.

ProductsPage (seller) — страница продавца, отображает его товары, форма добавления товара.

Формы регистрации, логина, добавления адреса, заказа и т.д.

Связи компонентов:

Все компоненты используют функции из api-service.tsx для получения/отправки данных.

Состояния (useState/useEffect) используются для хранения и отображения данных, полученных с backend.

4. Связи между слоями

Фронтенд отправляет запросы к бэкенду через функции из api-service.tsx.

Бэкенд работает с SQLite через запросы, реализует бизнес-логику и возвращает данные в формате JSON.

База данных хранит все сущности и связи между ними.

5. Типовой сценарий

Пользователь заходит на сайт, авторизуется.

Получает список товаров/категорий через API.

Может добавить товар в корзину или избранное — изменения уходят на backend и отражаются в базе.

Продавец может добавить товар, выбрав категорию, загрузив изображения и указав параметры.

Все изменения (корзина, избранное, отзывы, заказы) синхронизируются между фронтом и базой через backend.

Планирование и ход разработки

Планирование и распределение задач между участниками в команде осуществлялось через сервис GitHub.

ЗАКЛЮЧЕНИЕ

В ходе реализации проекта по созданию маркетплейса была проведена комплексная работа по проектированию, разработке и тестированию программного продукта. Проведённая оценка соответствия программного продукта поставленным требованиям показала, что основные функциональные и пользовательские ожидания были успешно реализованы. Приложение обеспечивает удобную регистрацию и авторизацию пользователей, предоставляет доступ к широкому каталогу товаров, реализует эффективный поиск, оформление заказов и управление личным кабинетом. Интерфейс приложения получился интуитивно понятным и адаптивным, что положительно сказывается на пользовательском опыте.

Вместе с тем, анализ выполненных требований выявил ряд аспектов, которые могут быть улучшены в дальнейшем. В частности, возможна доработка системы фильтрации и сортировки товаров, расширение возможностей личного кабинета, а также интеграция дополнительных платёжных систем. Для повышения конкурентоспособности продукта можно рассмотреть внедрение мобильной версии приложения, расширение аналитических инструментов для продавцов и покупателей.

В целом, проект можно считать успешным: были достигнуты основные цели, реализованы ключевые функции, обеспечено соответствие базовым требованиям заказчика и пользователя. Рекомендации по улучшению и развитию продукта создают прочную основу для его масштабирования и повышения эффективности в будущем.