

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: Обабков И.Н.

Куратор: Хрушков А.Е.

Студенты команды: КиберСамураи

Репин Михаил Евгеньевич РИ-230934

Валеев Илья Фагимович РИ-230944

Морозов Павел Александрович РИ-230941

Герасимов Анатолий Михайлович РИ-230942

Киреев Никита Максимович РИ-230944

Екатеринбург, 2025

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
КОМАНДА ПРОЕКТА.....	5
1. ОСНОВНАЯ ЧАСТЬ.....	6
1.1 Требования заказчика и пользователей.....	6
1.2 План действий (Backlog).....	9
1.3 Анализ аналогов.....	12
1.4 Описание методологии разработки.....	14
1.5 Процесс разработки.....	15
1.6 Отчет о результатах тестирования на промежуточных этапах.....	16
1.7 Обзор архитектуры программного продукта.....	17
1.8 Планирование деятельности.....	18
2. ОПИСАНИЕ РАБОТЫ.....	19
3. ЗАКЛЮЧЕНИЕ.....	21

## **ВВЕДЕНИЕ**

В условиях стремительного развития цифровых технологий и растущей популярности онлайн-торговли, создание эффективного маркетплейса становится не только актуальной, но и необходимой задачей для бизнеса. Цель данного проекта заключается в разработке многофункционального сайта маркетплейса, который объединяет продавцов и покупателей, обеспечивая удобный и безопасный процесс покупки и продажи товаров и услуг.

Задачи проекта включают в себя: анализ рынка и потребностей целевой аудитории, разработку пользовательского интерфейса, создание функционала для регистрации и управления аккаунтами пользователей, интеграцию систем оплаты и доставки, а также обеспечение высокого уровня безопасности данных. Все эти аспекты направлены на создание интуитивно понятной и эффективной платформы, способствующей росту продаж и удовлетворенности пользователей.

Актуальность данного проекта обусловлена тенденцией к увеличению доли онлайн-торговли в общем объеме розничной торговли. Многие предприниматели ищут новые каналы для реализации своих товаров. Создание маркетплейса позволит не только поддержать малый и средний бизнес, но и предложить потребителям широкий ассортимент товаров в одном месте, что существенно упростит процесс покупок.

Область применения разработанного программного продукта охватывает различные сегменты рынка, включая одежду, электронику, товары для дома и многое другое. Платформа будет полезна как для индивидуальных предпринимателей, так и для крупных компаний,

желающих расширить свои каналы сбыта и привлечь новых клиентов.

Ожидаемые результаты проекта включают в себя создание полностью функционального сайта маркетплейса, который будет отвечать современным требованиям пользователей и обеспечивать высокий уровень удобства и безопасности. Планируемые достижения по завершении проекта включают успешный запуск платформы, привлечение первых пользователей и продавцов, а также достижение устойчивого роста трафика и объема продаж в течение первых месяцев работы. Таким образом, данный проект имеет потенциал не только для коммерческого успеха, но и для значительного вклада в развитие электронной торговли.

## КОМАНДА ПРОЕКТА

**Тимлид:** Репин Михаил Евгеньевич РИ-230934.

**Аналитик:** Морозов Павел Александрович РИ-230941.

**Дизайнер:** Валеев Илья Фагимович РИ-230944.

**Frontend-разработчик:** Герасимов Анатолий Михайлович  
РИ-230942.

**Backend-разработчик:** Киреев Никита Максимович РИ-230944.

## Требования заказчика и пользователей

### 1.1 Функциональные требования:

- **Регистрация и авторизация пользователей:** Возможность создания учетных записей для покупателей и продавцов с использованием электронной почты, социальных сетей и других методов.
- **Управление товарами:** Продавцы должны иметь возможность добавлять, редактировать и удалять свои товары, загружать изображения и описания.
- **Поиск и фильтрация:** Необходимость в мощной системе поиска и фильтрации товаров по различным параметрам (категории, цена, рейтинг и т.д.).
- **Система оплаты:** Интеграция с различными платежными системами для обеспечения безопасных и удобных транзакций.
- **Система отзывов и рейтингов:** Возможность оставлять отзывы и оценки товаров и продавцов, что повысит доверие пользователей.
- **Административная панель:** Инструменты для администраторов для управления контентом, пользователями и заказами, а также для анализа статистики.

### 1.2. Нефункциональные требования:

- **Безопасность:** Обеспечение защиты данных пользователей и безопасности транзакций.

- **Масштабируемость:** Возможность расширения функционала и увеличения нагрузки на платформу без потери производительности.
- **Удобство использования:** Интуитивно понятный интерфейс, который обеспечит положительный пользовательский опыт.
- **Производительность:** Быстрая загрузка страниц и обработка запросов пользователей.

## 2.1. Требования покупателей:

- **Удобный интерфейс:** Легкость в навигации по сайту, доступность всех необходимых функций на видных местах.
- **Разнообразие товаров:** Широкий ассортимент товаров и услуг, доступных для покупки.
- **Информация о товарах:** Подробные описания, изображения и характеристики товаров, а также наличие отзывов от других пользователей.
- **Безопасность покупок:** Гарантия безопасности личных данных и средств при совершении покупок.
- **Поддержка:** Доступ к службе поддержки для решения возникающих вопросов и проблем.

## 2.2. Требования продавцов:

- **Простота управления товарами:** Удобный интерфейс для добавления и редактирования товаров, а также отслеживания заказов.

- **Аналитика и статистика:** Доступ к инструментам для анализа продаж, трафика и других метрик, которые помогут в принятии решений по улучшению бизнеса.
- **Обратная связь:** Возможность получать отзывы от покупателей и реагировать на них для улучшения качества товаров и услуг.



## **План действий (Backlog)**

### 1. Исследование и анализ

#### 1.1. Анализ рынка:

- Изучение существующих маркетплейсов и их функционала.
- Определение целевой аудитории и ее потребностей.

#### 1.2. Сбор требований:

- Проведение встреч с заказчиком для уточнения бизнес-целей и требований.
- Проведение опроса и интервью с потенциальными пользователями (покупателями и продавцами).

### 2. Проектирование

#### 2.1. Создание прототипа:

- Разработка низкоуровневых и высокоуровневых прототипов интерфейса.
- Проведение тестирования прототипов с пользователями для получения обратной связи.

#### 2.2. Архитектура системы:

- Составление архитектуры (монолитная, микросервисная и т.д.).
- Выбор технологий и инструментов для разработки (языки программирования, фреймворки, базы данных).

### 3. Разработка

#### 3.1. Настройка окружения:

- Подготовка серверов и баз данных.
- Настройка системы контроля версий (Git).

#### 3.2. Разработка функционала:

- Реализация модуля регистрации и авторизации пользователей.
- Создание системы управления товарами для продавцов.
- Реализация системы поиска и фильтрации товаров.
- Интеграция платежной системы.
- Разработка системы отзывов и рейтингов.
- Создание административной панели для администраторов.

#### 3.3. Тестирование:

- Проведение функционального тестирования всех модулей.
- Проведение нагрузочного тестирования для оценки производительности системы.
- Исправление выявленных ошибок и недостатков.

### 4. Запуск

#### 4.1. Подготовка к запуску:

- Завершение всех работ по разработке и тестированию.

#### 4.2. Запуск платформы:

- Открытие доступа к маркетплейсу для пользователей.

### 5. Поддержка и развитие

#### 5.1. Мониторинг и поддержка:

- Отслеживание работы платформы, выявление возможных проблем.
- Обработка обратной связи от пользователей и продавцов.

#### 5.2. Улучшение функционала:

- Анализ статистики использования платформы и выявление потребностей в новых функциях.
- Постепенная реализация улучшений и новых возможностей на основе полученной обратной связи.

## Анализ аналогов

Был проведен SWOT-анализ конкурентов:

- Ozon
- Wildberries
- ЯндексМаркет
- Мегамаркет
- МагнитМаркет

Сильные стороны конкурентов:

- **Простая регистрация:** Возможность быстрой регистрации по телефону
- **Инструменты для продавцов:** Предлагается множество инструментов и ресурсов для работы продавцов, включая аналитические данные.
- **Удобство оплаты:** Возможность оплаты различными способами, что расширяет круг покупателей.
- **Приятный дизайн:** Минимализм и узкая палитра цветов делают интерфейс легким и позволяют сосредоточиться на покупке

Слабые стороны конкурентов:

- **Высокие комиссии для продавцов:** Комиссии могут быть значительными, что может отпугнуть некоторых малых и средних предпринимателей.
- **Поддержка:** Слабый уровень поддержки затруднит решение возникшего конфликта, а впоследствии приведет к репутационным издержкам

## Описание методологии разработки

Для проекта была выбрана методология Agile с использованием спринтов продолжительностью по несколько недель. Этот подход позволил команде гибко реагировать на изменения требований и обеспечил последовательное и итеративное развитие продукта.

Каждый спринт включал следующие этапы:

1. Планирование спринта:

- Определение задач и целей на спринт.
- Оценка сложности задач и распределение их между членами команды.

2. Выполнение задач:

- Еженедельные встречи для координации работы.
- Регулярные демо-встречи для демонстрации прогресса и получения обратной связи.

3. Ретроспектива спринта:

- Анализ проделанной работы, выявление проблем и поиск путей их решения.
- Планирование улучшений для следующего спринта.

## **Процесс разработки**

Процесс разработки включал следующие этапы:

1. Анализ требований и планирование:

- Детальный анализ требований заказчика и пользователей.
- Создание плана проекта и распределение задач.

2. Разработка архитектуры системы:

- Выбор технологий и инструментов для разработки.
- Создание архитектуры системы.

3. Разработка функционала:

- Реализация функционала авторизации.
- Разработка модуля поиска товаров и оформления заказа.
- Внедрение функционала для панели продавца.
- Разработка функционала администратора.

4. Тестирование и отладка:

- Проведение юзабилити-тестирования.
- Исправление выявленных багов.

5. Запуск:

- Организация запуска сайта.
- Обеспечение поддержки пользователей.
- Планирование дальнейших улучшений на основе обратной связи.

## **Отчет о результатах тестирования на промежуточных этапах**

На каждом промежуточном этапе тестирования проводились следующие проверки:

1. Функциональное тестирование:

- Проверка всех функциональных возможностей сайта.
- Выявление несоответствий требованиям.

2. Тестирование производительности:

- Оценка времени обработки видеозаписей.
- Проверка скорости загрузки и обработки данных.

3. Юзабилити-тестирование:

- Оценка удобства использования интерфейса.
- Сбор обратной связи от пользователей.

4. Тестирование безопасности:

- Проверка мер по обеспечению безопасности данных



## Обзор архитектуры программного продукта

Тип архитектуры: Клиент-серверная с разделением на фронтенд и бэкенд.

Основные компоненты:

- Фронтенд:
  - Next.js (React-фреймворк) с SSR (Server-Side Rendering) для улучшенного SEO и производительности.
  - Клиентская часть реализована как SPA (Single Page Application) для динамического взаимодействия.
- Бэкенд:
  - Java (Spring Boot) для обработки бизнес-логики.
  - PostgreSQL как основная реляционная база данных.
- Дополнительные компоненты:
  - Redis для кэширования часто запрашиваемых данных.
- Взаимодействие компонентов:
  - Фронтенд общается с бэкендом через REST API (HTTP-запросы с использованием fetch API).
  - Для обеспечения кросс-доменных запросов настроен CORS на стороне бэкенда.
  - Контейнеризация через Docker обеспечивает изолированное развёртывание и масштабируемость.

Обоснование выбора архитектуры:

- Горизонтальная масштабируемость:
  - Stateless-дизайн бэкенда позволяет легко добавлять новые серверы.
  - Docker-контейнеры можно масштабировать в облачной среде.
- Отказоустойчивость:
  - Redis снижает нагрузку на БД, повышая стабильность.

Next.js обеспечивает резервный рендеринг даже при ошибках API.

- Быстрое развёртывание:
  - Docker Compose упрощает запуск всего стека (фронтенд + бэкенд + БД + Redis) одной командой.

## Планирование деятельности

Для планирования деятельности использовался сервис для управления проектами и контроля выполнения задач Yandex Tracker. Функционал данного сервиса помог нам скоординировать деятельность и распределить обязанности каждому участнику, а также следить за этапами работы по проекту.

На рисунке 1.1 показан пример хода выполнения задач.

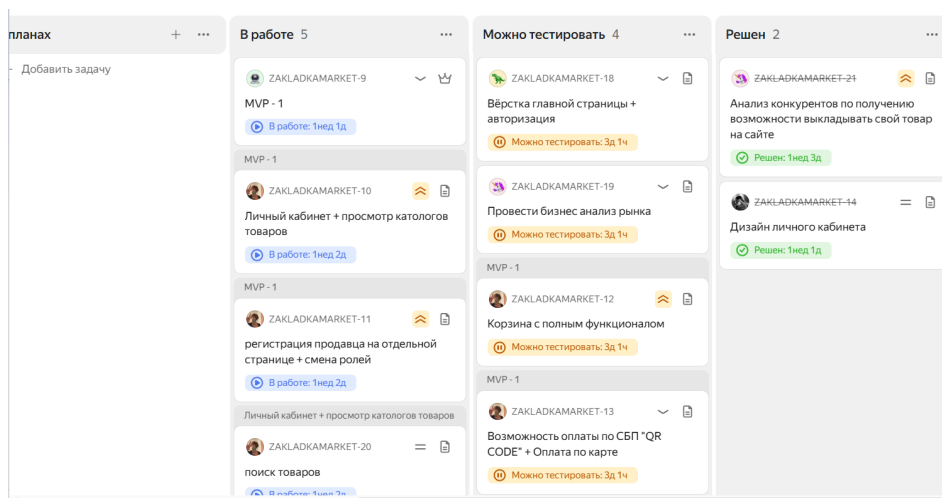


Рисунок 1.1 – Скриншот с Yandex Tracker

## **ОПИСАНИЕ РАБОТЫ**

Команда пользовалась методом Agile для работы над проектом. Проводились регулярные встречи для обсуждения прогресса (2 встречи в неделю), выявления текущих проблем и планирования следующих шагов. Каждый участник команды имел четко определенные задачи и сроки их выполнения, использовалась система управления проектами Yandex Tracker.

### **Тимлид**

Отвечал за общую координацию проекта, распределение задач между членами команды и контроль сроков выполнения. Основной задачей было обеспечить эффективное взаимодействие между участниками команды и заказчиком. Занимался управлением рисками и разрешением возникающих конфликтов, также занимался креативностью и предложением идей по развитию проекта.

### **Аналитик**

Собирали и анализировали требования к проекту. Проводил интервью с клиентами, чтобы изучить конкретные потребности и ожидания целевой аудитории от сервиса. Составлял сценарии использования приложения в графическом и текстовом виде. Проводил анализ конкурентов для выявления сильных и слабых сторон каждого. На основе полученной информации избрал нужное, что стало основой для разработки.

### **Дизайнер**

Разработал пользовательский интерфейс (UI), придерживаясь фирменной цветовой палитры и стиля. Создавал UI учитывающий потребности целевой аудитории.

Разработал детализированные макеты для различных экранов приложения, обеспечивая структурированное представление информации. Создавал прототипы, которые включали понятную и логичную навигацию, сокращая время, необходимое пользователям для выполнения основных задач. Учитывал сценарии взаимодействия, чтобы каждый шаг был интуитивно понятным.

Проводил тестирование пользовательских интерфейсов, чтобы определить их удобство использования.

Применял инструменты прототипирования для визуализации интерфейсов. Активно использовал автолейауты для создания гибких и адаптивных макетов, которые автоматически подстраивались под изменения содержания или размеров экранов.

### **Frontend-разработчик**

Основная зона ответственности — клиентская часть.

Проделанная работа:

- Разработка клиентской части:
  - Создание интерфейса маркетплейса на Next.js:
    - Вёрстка с использованием CSS-модулей.
    - Реализация динамических страниц (товары, корзина, профиль).
    - Использование App Router для роутинга по сайту
  - Интеграция с бэкендом через REST API (fetch).
- Контейнеризация:
  - Создание Dockerfile для фронтенда (сборка Next.js-приложения).
  - Настройка docker-compose.yml для совместного запуска с бэкендом.

Инструменты и технологии:

- Разработка:

- Next.js, React.
- GitHub для контроля версий.
- ESLint/Prettier для линтинга и форматирования.

Сложности и решения:

- Проблема: Ошибки CORS при запросах к бэкенду.
  - Решение: Координация с бэкенд-разработчиками для добавления правильных заголовков.
- Проблема: Оптимизация сборки Docker-образа для Next.js.
  - Решение: Многоэтапная сборка (уменьшение размера итогового образа).

Результаты:

- Завершён MVP маркетплейса с ключевыми функциями:
  - Просмотр товаров, фильтрация, корзина покупок.
  - Адаптивный интерфейс (поддержка мобильных устройств).
- Настроена инфраструктура для развёртывания (Docker).
- Обеспечено стабильное взаимодействие фронтенда и бэкенда.

## **Backend-разработчик**

### Разработка REST API:

- Спроектировал и реализовал эндпоинты для маркетплейса (управление товарами, заказами, категориями, пользователями).
- Использовал Spring Web для создания контроллеров и обработки HTTP-запросов.
- Настроил Swagger (springdoc-openapi) для автоматической документации API.

### Работа с базой данных:

- Разработал структуру БД (PostgreSQL) с использованием Spring Data JPA и Hibernate.
- Настроил миграции через Flyway (версия 9.22.3) для управления изменениями схемы БД.
- Оптимизировал SQL-запросы (индексы, избегание N+1 проблемы через @EntityGraph).

### Интеграция с внешними сервисами:

- Подключил платёжную систему ЮMoney — самая сложная часть из-за нюансов API и безопасности.
- Реализовал аутентификацию и авторизацию через Spring Security.
- Настроил отправку email-уведомлений через Spring Mail (подтверждение заказов, сброс пароля).

### Кэширование и производительность:

- Интегрировал Redis (через spring-boot-starter-data-redis) для кэширования часто запрашиваемых данных (каталог товаров).
- DevOps и развёртывание:
- Упаковал приложение в Docker-контейнер (сборка через Dockerfile).

- Настроил совместный запуск с фронтендом и БД через Docker Compose.

#### Результаты:

- Реализовано 20+ API-эндпоинтов для маркетплейса.
- Настроена аутентификация с ролями (USER, ADMIN).
- Интегрирована платёжная система.
- Готовый Docker-образ для быстрого развёртывания.



## ЗАКЛЮЧЕНИЕ

В результате реализации проекта маркетплейса мы достигли значительного соответствия поставленным требованиям как со стороны заказчика, так и пользователей. Основные цели, определенные на начальных этапах, были успешно выполнены, что позволило создать функциональную и удобную платформу для взаимодействия покупателей и продавцов. В процессе работы мы акцентировали внимание на потребностях целевой аудитории, что способствовало разработке интуитивно понятного интерфейса и эффективной системы управления товарами.

Анализ выполненных требований показал, что большинство из них были реализованы в соответствии с ожиданиями. Однако мы также выявили несколько областей, которые требуют дополнительного внимания, таких как улучшение пользовательского опыта в процессе оформления заказа и оптимизация системы поиска товаров. Эти аспекты были отмечены как ключевые для повышения удовлетворенности пользователей.

Оценка качества программного продукта, основанная на результатах тестирования, выявила несколько дефектов и ошибок, которые, хотя и не критичны, могут негативно сказаться на общей работоспособности платформы. Например, некоторые функции поиска показали замедленную реакцию при высоких нагрузках, что может привести к негативному опыту. Эти результаты подчеркивают необходимость в дальнейшей оптимизации кода и инфраструктуры, а также в регулярном проведении нагрузочного тестирования для обеспечения стабильной работы платформы в условиях растущей пользовательской базы.

В качестве предложений по улучшению продукта мы рекомендуем внедрить более продвинутые алгоритмы поиска и фильтрации, что

позволит пользователям быстрее находить нужные товары. Также следует рассмотреть возможность добавления новых функциональных возможностей, таких как персонализированные рекомендации на основе анализа поведения пользователей. Эти меры не только улучшат пользовательский опыт, но и создадут предпосылки для дальнейшего развития платформы, включая расширение ассортимента товаров и услуг, а также привлечение новых категорий продавцов.

В заключение, мы уверены, что с учетом полученных данных и рекомендаций, маркетплейс имеет все шансы стать успешной и востребованной платформой, способной удовлетворить потребности пользователей и обеспечить рост бизнеса. Мы готовы продолжать работу над проектом, чтобы сделать его еще более эффективным и удобным.