

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 маркетплейса»  
по дисциплине «Проектный практикум»

Заказчик: Шестеров М. А.

Куратор: Хрущков А. Е.

Студенты команды

Васепцова Д. С.

Гаврилов К. А.

Киценко М. А.

Крупина П. А.

Матюхина А. С.

Екатеринбург, 2025

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
1 Требования заказчика.....	5
2 Целевая аудитория .....	7
3 Анализ и сопоставление аналогов разрабатываемого продукта.....	9
4 Архитектура программного обеспечения .....	12
4.1 Выбор архитектурного подхода .....	12
4.2 Основные компоненты архитектуры.....	12
5 Методология разработки .....	14
5.1 Информация о процессе разработки .....	14
5.2 Тестирование и обработка ошибок.....	15
5.3 Планирование деятельности .....	16
5.4 Выводы по процессу разработки .....	17
6 Отчёт о работе каждого участника команды .....	18
6.1 Крупина Полина Андреевна, РИ-320940, frontend-разработчик.....	18
6.2 Киценко Марина Алексеевна, РИ-320915, тимлид.....	20
6.3 Васепцова Дарья Сергеевна, РИ-320947, аналитик .....	22
6.4 Гаврилов Константин Александрович, РИ-320940, backend разработчик .....	24
6.5 Матюхина Анна Сергеевна, РИ-320947, дизайнер .....	25
ЗАКЛЮЧЕНИЕ .....	28
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	30

## **ВВЕДЕНИЕ**

Многие начинающие предприниматели, особенно те, кто занимается собственным производством, сталкиваются с трудностями при выходе на цифровой рынок. У них часто отсутствуют технические навыки, ресурсы для продвижения и понимание онлайн-торговли. Существующие платформы либо слишком сложны, либо ориентированы на опытных пользователей и крупный бизнес. Это ограничивает возможности таких предпринимателей в развитии, масштабировании и конкурентной борьбе.

Создание доступного маркетплейса с интуитивным интерфейсом, инструментами продвижения и встроенной системой управления бизнесом решит эту проблему, предоставив малому бизнесу полноценную точку входа в онлайн-коммерцию.

Цель проекта – создание доступного и удобного маркетплейса, который будет ориентирован на начинающих предпринимателей и малый бизнес, особенно на тех, кто занимается собственным производством.

Для достижения поставленной цели предусмотрено выполнение следующих задач:

- создание и настройка интернет-магазина;
- управление несколькими магазинами с одного аккаунта;
- управление добавлением, редактированием и удалением товаров;
- управление заказами и изменение их статусов;
- просмотр баланса и приём онлайн-оплаты, включая оплату банковскими картами;
- просмотр товаров и карточек продукта;
- добавление в корзину, оформление и оплата заказа;
- получение уведомлений о статусе заказа;
- система отзывов с возможностью добавления оценок и комментариев;
- регистрация и авторизация через email с подтверждением;

- пользовательский профиль с редактированием данных;
- управление способами оплаты;
- админ-панель для управления пользователями, товарами, категориями, жалобами и статистикой продавцов;
- масштабируемость и гибкость архитектуры для внедрения новых функций без переработки.

## **1 Требования заказчика**

В рамках проекта предусматривается разработка интернет-магазина с расширенным функционалом, включая кастомизацию визуального оформления и систему аналитики. Ключевым требованием является возможность настройки цветовой палитры в соответствии с корпоративным стилем, а также формирование статистических отчетов по продажам, заказам и клиентской базе.

Сервис должен обеспечивать выполнение CRUD-операций (создание, чтение, обновление, удаление) для таких сущностей, как товары, категории товаров, пользователи (покупатели и продавцы) и новостные публикации.

Для реализации проекта предполагается последовательное выполнение ряда этапов. На начальной стадии проводится детальный анализ требований заказчика с уточнением технического задания, а также сравнительный анализ существующих решений для определения конкурентных преимуществ. Далее осуществляется проектирование, включающее разработку пользовательских сценариев, определение функциональных и нефункциональных требований, создание технической документации и проектирование интерфейсов в Figma с последующим тестированием на удобство использования.

Архитектурная часть проекта подразумевает проектирование общей структуры приложения, разработку схемы базы данных с учетом хранения информации о заказах, пользователях и товарах, а также построение UML-диаграмм для наглядного отображения взаимодействия компонентов системы.

На этапе разработки выполняется настройка среды, реализация механизма аутентификации и авторизации, создание личных кабинетов для продавцов и покупателей, а также внедрение функционала управления товарами и заказами. Завершающая стадия включает подготовку демонстрационного прототипа и презентационных материалов, демонстрирующих ключевые возможности приложения.

Данный подход обеспечивает комплексную и структурированную реализацию проекта в соответствии с современными стандартами разработки программного обеспечения.

## **2 Целевая аудитория**

Целевая аудитория маркетплейса формируется из двух основных групп: покупателей и продавцов, каждая из которых включает несколько сегментов с характерными потребностями, проблемами и ожиданиями от платформы.

Среди покупателей выделяется три ключевых сегмента. Первый сегмент представлен молодежью и миллениалами в возрасте 18–35 лет, которые активно интересуются модой, электроникой, косметикой и актуальными трендами. Основными каналами взаимодействия с этой аудиторией являются социальные сети. К типичным проблемам данного сегмента относятся высокая стоимость товаров, длительные сроки доставки и отсутствие достоверных отзывов. Решением могут стать внедрение геймифицированных механик, системы кэшбэка, ускоренной доставки, а также отзывов с фото и видео подтверждением.

Второй сегмент составляют родители в возрасте 25–45 лет, ориентированные на приобретение детских товаров, игрушек, бытовой техники и других продуктов, обеспечивающих удобство в повседневной жизни. Основные сложности связаны с дефицитом времени, необходимостью поиска всего необходимого в одном месте и озабоченностью качеством товаров. Для удовлетворения запросов данной аудитории целесообразно предусмотреть специализированные категории, сервисы подписок, тематические подборки и гарантированно быструю доставку.

Третий сегмент включает хозяйственных покупателей старше 35 лет, которые преимущественно приобретают товары для дома, дачи, ремонта и кухни. Основными барьерами для них являются сложность навигации по платформе и недоверие к малоизвестным брендам. В качестве решений рекомендуется разработать упрощенный пользовательский интерфейс, предоставить гарантии на товары, обеспечить качественную поддержку и акцентировать внимание на проверенных продавцах.

Среди продавцов также можно выделить три ключевых сегмента. Первый сегмент включает представителей малого бизнеса и самозанятых, которые стремятся начать продажи в онлайн-формате без значительных финансовых затрат. Основные сложности связаны с процедурой регистрации, высокими комиссиями и общей сложностью адаптации к платформе. Для привлечения этой аудитории необходимо обеспечить упрощенную процедуру входа, минимальные комиссии, пошаговые инструкции и оперативную поддержку.

Второй сегмент представлен средним бизнесом и производителями, заинтересованными в расширении каналов сбыта и автоматизации бизнес-процессов. Их основные потребности включают доступ к аналитическим инструментам, стабильную логистику и возможности интеграции с внешними системами. В данном случае ключевыми решениями станут предоставление личного кабинета с расширенной аналитикой, а также инструменты для продвижения товаров.

Третий сегмент объединяет крафтовые и локальные бренды, которые стремятся выйти на массовую аудиторию. Основные проблемы заключаются в низкой заметности среди крупных игроков рынка и отсутствии навыков продвижения. Для поддержки данной категории продавцов целесообразно создать отдельную витрину для локальных брендов, предложить специальные условия продвижения и минимальные комиссии.

Таким образом, успешное функционирование маркетплейса требует учета специфики каждой группы пользователей и реализации решений, направленных на удовлетворение их ключевых потребностей.

### **3 Анализ и сопоставление аналогов разрабатываемого продукта**

Для создания конкурентоспособного маркетплейса, ориентированного на начинающих предпринимателей и производителей, был проведён анализ популярных платформ электронной коммерции: Wildberries, Ozon, Яндекс.Маркет, AliExpress и Amazon. Ниже приводится сравнительный обзор, включающий сильные и слабые стороны, а также идеи, которые могут быть полезны при разработке нашего продукта.

Wildberries — один из крупнейших российских маркетплейсов с широкой базой покупателей. Привлекает продавцов простотой регистрации и наличием автоматизированных инструментов загрузки товаров. Однако пользователи нередко жалуются на низкое качество клиентской поддержки и непрозрачные правила возврата, что может отпугивать как продавцов, так и покупателей. Среди сильных сторон — высокая степень логистической интеграции, охватывающая всю территорию РФ.

Идеи для заимствования:

- простая регистрация и онбординг продавцов;
- автоматизация загрузки и управления товарами;
- быстрая логистика на старте.

Ozon — платформа с одной из самых развитых логистических сетей в стране. Привлекательна наличием услуги "Склад под ключ" (FBO) и инструментами внутреннего маркетинга. Минусом является то, что Ozon сам выступает продавцом на платформе, создавая конкуренцию своим же партнёрам. Также платформе свойственна высокая насыщенность, что делает её менее привлекательной для начинающих.

Идеи для заимствования:

- встроенная рекламная система;
- интеграция со складскими и логистическими сервисами;
- маркетинговые инструменты внутри платформы.

Яндекс.Маркет — отличается глубокой интеграцией с экосистемой Яндекса и наличием умных рекомендаций и поиска по характеристикам. Тем не менее, для некоторых категорий товаров интерфейс остаётся сложным и перегруженным, что затрудняет навигацию.

Идеи для заимствования:

- умный поиск и подбор товаров;
- рекомендательные системы на основе поведения пользователей;
- гибкая настройка карточек товаров продавцом.

AliExpress — глобальный маркетплейс, ориентированный на доступность и массовость. Его сильные стороны — широкий ассортимент, низкие цены, мультивалютность и встроенные языковые переводы. Главные минусы — долгая доставка и частые проблемы с качеством товаров. Однако он сохраняет популярность за счёт удобной глобальной торговли.

Идеи для заимствования:

- встроенный перевод интерфейса и описаний;
- многоязычная и мультивалютная поддержка;
- гибкие настройки витрины под международную аудиторию.

Amazon — мировой лидер электронной торговли. Известен своей надёжностью, мощной системой доставки, подпиской Amazon Prime и функцией One-Click заказа. Основные недостатки — высокие комиссии и жесткие правила для продавцов, что делает его недоступным для многих новичков.

Идеи для заимствования:

- подписка с бонусами;
- упрощённая покупка;
- развитая система отзывов и рейтингов.

Проведённый анализ показывает, что большинство крупных платформ ориентированы либо на массовую аудиторию, либо на опытных продавцов с ресурсами. Наша задача — адаптировать лучшие практики для начинающих пользователей, избегая перегрузки интерфейса и чрезмерной сложности.

Наиболее перспективные идеи для реализации:

- AI-рекомендации и персонализация контента для покупателей;
- система подписки с бонусами;
- отзывы с видео/фото и рейтингами продавцов;
- простой процесс регистрации и настройки магазина;
- интеграция с платёжными системами и логистикой.

Эти элементы позволяют выделиться на фоне существующих решений и сделать платформу удобной, эффективной и привлекательной для целевой аудитории.

## **4 Архитектура программного обеспечения**

### **4.1 Выбор архитектурного подхода**

Разработка программного продукта осуществляется на основе клиент-серверной микросервисной архитектуры, где каждый компонент выполняет строго определённую функцию и взаимодействует с другими через API. Выбранный подход обеспечивает:

- масштабируемость;
- гибкость в обновлении и расширении функциональности;
- надёжность и отказоустойчивость за счёт модульности;
- безопасность обработки и хранения данных.

### **4.2 Основные компоненты архитектуры**

Архитектура программного продукта построена по принципу клиент-серверного взаимодействия и включает три ключевых компонента: клиентскую часть, серверную часть и базу данных. Кроме того, определены системные и инфраструктурные требования, необходимые для корректной работы платформы.

Клиентская часть реализована с использованием фреймворка React и языка программирования TypeScript. Для управления маршрутизацией используется библиотека React Router, что позволяет обеспечивать плавную навигацию между страницами без перезагрузки. Интерфейс построен по модели Single Page Application, поддерживает адаптивную вёрстку и корректно отображается в популярных браузерах, таких как Chrome, Firefox, Edge и Safari. Основной функционал клиентской части включает отображение каталога товаров, регистрацию и авторизацию пользователей, работу с

корзиной, оформление заказов, управление личным кабинетом и другие взаимодействия пользователя с системой.

Серверная часть разработана на языке Python с использованием Django и Django REST Framework. Она отвечает за обработку бизнес-логики приложения, работу с базой данных и предоставление API-эндпоинтов для клиентской части. Поддерживаются все основные CRUD-операции для сущностей, таких как товары, категории, пользователи, магазины, бренды и новости.

База данных построена на реляционной СУБД PostgreSQL, что обеспечивает строгую структуру данных и надёжность транзакций. Для визуального управления базой используется инструмент DBeaver. В базе хранятся данные пользователей, информация о товарах, заказах, чатах, отзывах и системных настройках.

Системные и инфраструктурные требования включают минимальный объём дискового пространства сервера в 36 ТБ. Также предусмотрена поддержка автоматического масштабирования, что позволяет системе динамически увеличивать ресурсы при росте пользовательской нагрузки и объёма данных.

## **5 Методология разработки**

В рамках реализации проекта маркетплейса была выбрана методология Agile с акцентом на итеративную и инкрементальную разработку. Методология разрабатывалась и адаптировалась участниками команды самостоятельно. Основной целью было создание минимально жизнеспособного продукта, который в дальнейшем можно масштабировать и расширять.

Ключевые принципы разработки:

- работа итерациями и деление задач на спринты;
- постоянный контроль прогресса внутри команды;
- регулярная демонстрация результатов и обратная связь от куратора проекта.

Учебный характер проекта исключал наличие внешнего заказчика, однако роль «условного заказчика» выполнял куратор, который отслеживал ход выполнения, направлял участников и давал рекомендации по улучшению.

### **5.1 Информация о процессе разработки**

Процесс разработки включал следующие этапы:

1) подготовительный этап:

- формулировка целей проекта;
- анализ аналогов и выделение ключевых функций;
- определение целевой аудитории;
- разработка общей архитектуры проекта и распределение ролей.

2) планирование и организация:

- работа организована в виде спринтов;
- вся деятельность фиксировалась и контролировалась через доску задач Miro;

– регулярные командные встречи проводились раз в неделю под руководством куратора, где каждый участник отчитывался о прогрессе.

3) разработка MVP:

– разработка фронтенд-интерфейса для пользователей и админ-панели;

– реализация бэкенд-логики: регистрация, аутентификация, взаимодействие с товарами и заказами;

– интеграция клиентской и серверной частей через REST API;

– разработка базовой модели базы данных и её реализация на PostgreSQL.

4) финальный этап:

– рефакторинг и оптимизация кода;

– устранение выявленных ошибок;

– подготовка презентационных материалов.

## 5.2 Тестирование и обработка ошибок

Тестирование проводилось вручную в рамках каждой итерации. Основной акцент делался на проверку работоспособности реализованного функционала и быструю обработку ошибок. Осуществлялась проверка корректности работы отдельных компонентов после реализации. Применялось интеграционное тестирование: взаимодействие клиента и сервера. А также визуальное тестирование интерфейса.

В процессе работы были выявлены ошибки на разных этапах разработки.

Фронтенд:

- некорректное отображение изображений в карточке товара;
- ошибки при обновлении состояния корзины.

Бэкенд:

- ошибки при валидации данных при регистрации пользователя;

- неверная обработка отсутствующих полей в запросе.

Интеграция:

- проблемы с CORS при первом подключении фронтенда к API;
- несовпадение форматов данных между клиентом и сервером.

Все выявленные ошибки устранились непосредственно после обнаружения, что позволило не накапливать технический долг.

### 5.3 Планирование деятельности

Проект выполнялся в команде, участники которой самостоятельно распределяли роли и задачи. Общее планирование происходило на первом этапе, а последующая работа строилась итерационно.

Распределение обязанностей между разработчиками проводилось в соответствии с их ролями. Так, в задачи фронтенд-разработчика входили:

- реализация пользовательского интерфейса маркетплейса;
- реализация админ-панели;
- интеграция с API;
- вёрстка;
- маршрутизация.

Задачей бэкенд-разработчика стояло:

- проектирование и реализация REST API;
- работа с базой данных;
- обеспечение безопасности и валидации данных.

Контроль прогресса осуществлялся в фиксировании задач в Kanban доске и продвижением этих задач по колонкам: новые задачи, в процессе, выполнены. Доска с задачами делилась на несколько строк под каждого участника команды для упорядочивания задач. Раз в неделю проходили встречи с куратором, где каждый участник делал краткий отчёт, что также способствовало контролю процесса работы над проектом.

## **5.4 Выводы по процессу разработки**

Выбранный подход обладает рядом преимуществ, таких как гибкое планирование, которое позволяет легко адаптироваться к изменениям и перераспределять задачи. Также стоит отметить высокую прозрачность процесса благодаря использованию доски задач и регулярным встречам. Постепенная интеграция функциональности позволила на раннем этапе получать рабочие версии системы, что способствовало более эффективному развитию проекта.

Однако у данного подхода есть и свои недостатки. Отсутствие автоматизированного тестирования может привести к ошибкам на продвинутых этапах разработки. Некоторые задачи решались дольше из-за совмещения учебной нагрузки и проектной деятельности, что замедляло общий прогресс. Маленький состав команды ограничивал параллельность выполнения задач, что также сказывалось на скорости реализации проекта.

Для улучшения процесса в будущем рекомендуется внедрение unit и интеграционных тестов на серверной и клиентской части, что поможет выявлять ошибки на ранних этапах. Автоматизация сборки и деплоя проекта позволит ускорить процесс разработки и снизить количество рутинных операций. Расширение команды или более эффективное распределение задач с учётом приоритетности и сроков также способствует повышению продуктивности и своевременному достижению целей проекта.

## **6 Отчёт о работе каждого участника команды**

### **6.1 Крупина Полина Андреевна, РИ-320940, frontend-разработчик**

Ниже представлен отчет о проделанной работе по разработке фронтенд-части маркетплейса. Описание охватывает реализацию пользовательского интерфейса как со стороны клиента, так и со стороны администратора системы. Все работы были выполнены с применением технологий React, TypeScript и Redux, обеспечивающих масштабируемость, устойчивость и высокую производительность приложения.

На первом этапе был реализован клиентский сервис, включающий в себя две пользовательские роли — покупателя и продавца. Пользовательская версия для покупателя состоит из нескольких ключевых страниц и компонентов. Главная страница отображает список товаров, каждый из которых представлен в виде карточки со ссылкой на детальную страницу. Страница товара содержит полную информацию, включая название, галерею изображений, таблицы с характеристиками и описанием, рейтинг, блок рекомендуемых товаров, а также превью отзывов с возможностью перехода на отдельную страницу с полным списком отзывов.

В шапке сайта расположен каталог, реализованный в виде выпадающего меню с тремя уровнями вложенности — категории, подкатегории и группы товаров. Навигация осуществляется как по наведению курсора, так и по клику, с последующим переходом на страницу выбранной категории. Также в шапке реализованы поиск, профиль пользователя, раздел заказов, избранное и корзина.

В личном кабинете доступны следующие разделы: редактируемые личные данные (имя, фамилия, электронная почта, пол), раздел отзывов с перечнем приобретенных товаров, ожидающих отзывов, и написанными ранее отзывами, раздел способов оплаты с отображением сохраненных карт и

возможностью привязки новых, а также система уведомлений и кнопка выхода.

Раздел «Заказы» содержит четыре вкладки: текущие, завершенные, купленные и возвращенные товары. В разделе «Избранное» представлен список всех добавленных товаров, а в «Корзине» реализовано управление выбранными товарами, включая чекбоксы, выбор всех и оформление заказа с переходом на страницу оплаты. После завершения оплаты реализован переход на страницу заказов.

Вход и регистрация в систему реализованы с использованием одноразового кода, отправляемого на электронную почту. При регистрации дополнительно запрашиваются имя и пол.

Версия для продавца начинается с аналитической панели, отображающей ключевые показатели эффективности, такие как выручка, возвраты и рейтинг, таблицы с остатками товаров и товарами, ожидающими отправки, а также графики по заказам и выручке за неделю. Раздел «Профиль» предоставляет возможность редактирования личных данных, загрузки документов и добавления способов получения выплат. В разделе уведомлений отображаются актуальные системные сообщения.

Раздел «Мои заказы» предоставляет доступ к списку заказов с возможностью просмотра подробной информации и изменения статуса заказа. Раздел «Мои товары» содержит интерфейс для редактирования, удаления и создания новых товаров, включая настройку названия, цены, описания, таблиц характеристик, выбора категорий и загрузки изображений. В разделе «Финансы» отображаются транзакции, связанные с продажами. Раздел «Отзывы» позволяет просматривать отклики на товары с возможностью ответа.

Фронтенд административной панели разработан как отдельный сервис. В рамках MVP администратор имеет возможность управлять категориями и характеристиками товаров. Характеристики жестко привязываются к

категориям, а при создании или редактировании товара продавец выбирает их из предложенного списка.

Таким образом, в рамках разработки была реализована комплексная и масштабируемая архитектура интерфейсов для всех ролей системы. Проектирование и реализация фронта обеспечили значительное расширение практических навыков работы с React, TypeScript, Redux и компонентным подходом в рамках создания сложных SPA-приложений. Работа над проектом включала активное взаимодействие с пользовательскими сценариями, сложными компонентами, адаптивной версткой, а также повышенное внимание к организации состояния и маршрутизации.

## **6.2 Киценко Марина Алексеевна, РИ-320915, тимлид**

В процессе реализации учебного проекта, направленного на разработку маркетплейса, ориентированного на начинающих предпринимателей и малый бизнес, была организована командная работа, требующая эффективной координации всех участников. Функции тимлида включали координацию рабочих процессов, постановку задач, организацию совместной деятельности, а также ведение проектной документации и обеспечение постоянной связи с куратором.

На подготовительном этапе были сформулированы ключевые параметры проекта. В первую очередь были определены проблема, цель и задачи проекта. Проблема заключалась в отсутствии доступной цифровой платформы для малых предпринимателей, позволяющей быстро выходить на онлайн-рынок. Целью стало создание маркетплейса с интуитивно понятным интерфейсом, базовым функционалом для продавцов и покупателей и административной частью для управления содержанием. К задачам проекта относились: анализ рынка и конкурентов, формирование пользовательских

сценариев, проектирование архитектуры, реализация интерфейса и бэкенда, а также проведение итогового тестирования.

Особое внимание уделялось постановке задач. На основе анализа этапов проектирования и применения гибкой методологии Agile, каждая задача фиксировалась и структурировалась в соответствии с приоритетами. В качестве инструмента визуализации и контроля использовалась доска задач в Miro. Бэклог проекта формировался на основании пользовательских историй, созданных аналитиком. При составлении бэклога учитывались рекомендации, представленные в материалах: «Что такое бэклог», «Как составлять бэклог: краткое руководство», а также материалы от Outlines Tech и Яндекс.Практикума по анализу конкурентов и управлению требованиями. Задачи классифицировались по уровням — бизнес-задачи, технические задачи и задачи проектной документации. Для разработчиков отдельно формулировались технические цели с привязкой к этапам реализации.

Для организации командного взаимодействия был создан централизованный репозиторий материалов проекта. В данном хранилище размещались ссылки на Google-диск с аналитическими данными, макетами интерфейсов, ссылкой на дизайн в Figma, доступом к репозиториям в GitHub и вспомогательным документам. Таким образом, участники проекта имели быстрый доступ к актуальной информации и единое рабочее пространство. Структуризация хранилища позволила минимизировать дублирование и упростить навигацию между разделами.

Связь с куратором проекта поддерживалась на регулярной основе. Онлайн встречи проводились еженедельно в заранее согласованное время. На каждой онлайн встрече все участники команды демонстрировали промежуточные результаты, озвучивали возникающие трудности и уточняли ожидания по предстоящим задачам. Результаты встреч фиксировались, при необходимости вносились корректировки в распределение задач и сроки выполнения.

Организация командной работы сопровождалась решением нескольких внутренних задач. На первом этапе была выявлена проблема отсутствия чёткой системы коммуникации между участниками, что приводило к задержкам и пересечению задач. В ходе итеративного анализа и обратной связи от куратора удалось внедрить единый формат взаимодействия, а также согласовать график синхронных и асинхронных коммуникаций. Внутри команды были установлены чёткие роли, а ответственность каждого участника закреплялась как за результат, так и за соблюдение сроков.

Проект показал, что грамотно выстроенная система управления задачами, регулярная обратная связь, а также централизованное хранение данных позволяют значительно повысить продуктивность и организованность команды даже в условиях учебного проектирования. Кроме того, успешное внедрение элементов Agile-методологии подтвердило её применимость и эффективность в условиях ограниченных сроков и ресурсов. Опыт ведения командной работы в рамках данного проекта способствовал развитию управленческих и организационных компетенций, а также продемонстрировал важность системного подхода в разработке цифровых продуктов.

### **6.3 Васецова Дарья Сергеевна, РИ-320947, аналитик**

В указанный период было выполнено значительное количество задач, направленных на успешную реализацию проекта. В рамках первой итерации осуществлялось активное взаимодействие с заказчиком и проектной командой, что позволило детально определить ключевые требования к веб-ресурсу. На основе собранной информации были разработаны пользовательские истории, описывающие основные сценарии использования приложения, которые впоследствии легли в основу функциональных требований.

В рамках данной итерации также был проведён анализ целевой аудитории с целью изучения её потребностей, поведения и ожиданий. Полученные данные позволили сосредоточиться на аспектах, наиболее значимых для конечных пользователей. В целях повышения конкурентоспособности проекта был выполнен обзор аналогичных решений, выявлены их сильные и слабые стороны, что дало возможность определить уникальные преимущества разрабатываемого продукта. Дополнительно осуществлялось проектирование информационной архитектуры сайта, включая определение основных разделов и построение удобной навигационной структуры.

На втором этапе основное внимание было сосредоточено на документировании и уточнении деталей функциональности. Завершена подготовка отчёта по результатам первой итерации, представленного на совместном обсуждении с командой и заказчиком.

Заключительный этап работы включал финализацию отчётной документации и подведение итогов. Итоговый отчёт о проделанной работе был подготовлен и принят в качестве основы для последующей оптимизации и доработки сайта. Продолжались консультации с заказчиком и участниками команды, в ходе которых уточнялись оставшиеся детали проекта и согласовывались финальные этапы его реализации.

Работа над проектом способствовала углублению профессиональных компетенций в области бизнес-анализа, проектирования пользовательских сценариев и внедрения эффективных подходов к оценке и приоритизации задач. Полученный опыт подтвердил важность системного подхода и тесного взаимодействия всех участников проекта на всех этапах его выполнения.

## **6.4 Гаврилов Константин Александрович, РИ-320940, backend разработчик**

За поставленный период были поставлены и выполнены ключевые цели, включающие проектировку архитектуры базы данных, разработку архитектуры API для серверной части приложения, непосредственную разработку API, интеграцию Backend и Frontend частей приложения, контейнеризацию и оркестрацию веб-приложения, а также деплой приложения на сервер. На протяжении всего семестра осуществлялась непрерывная разработка и интеграция решений между Backend и Frontend частями приложения. Разработка велась с учетом изменения изначально спроектированной архитектуры под нужды приложения, что позволило максимально гибко масштабировать приложение во всех направлениях, делая его более устойчивым, безотказным и удовлетворяющим всем требованиям бизнес-логики.

В ходе разработки была применена микросервисная архитектура для приложения. Изначальный монолит приложения был написан на синхронных запросах, что не удовлетворяло требованиям для сообщений и уведомлений, которые должны были приходить асинхронно. Для решения этой проблемы были разработаны два дополнительных микросервиса, которые работали обособленно от основного монолитного приложения и использовали его лишь для авторизации. Это позволило интегрировать в приложение асинхронные решения, что значительно улучшило пользовательский опыт.

Контейнеризация и оркестрация приложения осуществлялись с использованием восьми контейнеров Docker, что позволяло удобно разворачивать и управлять проектом при помощи Docker Compose. Такой подход обеспечил легкое управление версиями приложения и возможность параллельно вести разработку, сохраняя сервер включенным. В будущих планах рассматривается добавление отказоустойчивости с использованием

продвинутых оркестраторов, таких как Docker Swarm или Kubernetes. Это позволит настроить отказоустойчивую систему с использованием нескольких нод и реплик контейнеров, между которыми будет распределяться нагрузка.

В процессе деплоя приложения на сервер было принято решение использовать домашний компьютер вместо удаленного сервера, что позволило сэкономить на аренде виртуальных машин. В случае необходимости переноса решения на другой сервер, это можно было бы сделать быстро и удобно, так как все необходимые инструменты для этого уже настроены. Такой подход обеспечивает гибкость и мобильность в управлении инфраструктурой приложения.

В ходе разработки и внедрения решений особое внимание уделялось обеспечению безопасности и надежности приложения. Были реализованы механизмы аутентификации и авторизации, а также обеспечена защита данных пользователей. Кроме того, проводилось тестирование приложения на различных этапах разработки, что позволило выявить и устранить возможные ошибки и уязвимости.

Таким образом, выполненная работа представляет собой комплексное проектирование и разработку веб-приложения, учитывающего современные требования к безопасности, надежности и масштабируемости. Полученные результаты и перспективы дальнейшей разработки позволяют создать устойчивую и функциональную платформу, способствующую эффективному взаимодействию пользователей. В ходе работы были применены современные методы и технологии разработки, что позволило создать продукт, соответствующий современным требованиям и ожиданиям пользователей.

## **6.5 Матюхина Анна Сергеевна, РИ-320947, дизайнер**

В рамках семестровой работы осуществлялось проектирование и разработка дизайна маркетплейса, включающего три ключевые

пользовательские роли: покупатель, продавец и администратор. Основной задачей являлось создание интуитивно понятных и функциональных интерфейсов, обеспечивающих эффективное взаимодействие пользователей с платформой.

На этапе аналитической подготовки были разработаны user flow для каждой роли с детализацией ключевых сценариев взаимодействия. Определены целевые задачи пользователей: для покупателей — удобный поиск, отбор и приобретение товаров. Для продавцов — управление ассортиментом и заказами. Для администратора — контроль контента и настройка комиссий. Данный этап позволил выявить основные потребности и ожидания пользователей, что стало основой для дальнейшего проектирования.

Прототипирование интерфейсов началось с создания low прототипов основных экранов на основе аналитических данных. Проведены итерационные доработки с учетом проверки логики взаимодействия и устранения выявленных недостатков. Это позволило оптимизировать пользовательские пути и улучшить общую структуру интерфейсов.

Разработка дизайн-системы включала реализацию единого визуального стиля с соблюдением принципов юзабилити и доступности. Для каждой роли разработаны полностраничные макеты, что обеспечило согласованность дизайна и логики взаимодействия на всех этапах. Особое внимание уделялось созданию интуитивно понятных элементов интерфейса, которые облегчают пользователям выполнение их задач.

Результаты работы включают полностью завершенное проектирование интерфейсов для всех пользовательских ролей. Обеспечена согласованность дизайна и логики взаимодействия на всех этапах, что способствует повышению удобства использования платформы. Проведены итерации от анализа user flow до финального дизайна, что позволило учесть ключевые потребности каждой группы пользователей.

Перспективы дальнейшей разработки включают проведение юзабилити-тестирования с последующей доработкой интерфейсов. Оптимизация дизайна

на основе данных пользовательского поведения позволит улучшить взаимодействие с платформой и повысить удовлетворенность пользователей. Развитие системы дизайн-компонентов обеспечит масштабируемость проекта и упростит внесение изменений в будущем.

Работа над проектом позволила углубить навыки проектирования сложных интерфейсов с множеством сценариев взаимодействия. В дальнейшем планируется доработать детали на основе обратной связи, что позволит создать более удобную и эффективную платформу для всех пользователей. Особое внимание будет уделено анализу данных пользовательского поведения и внедрению улучшений, направленных на повышение удобства и функциональности интерфейсов.

Таким образом, выполненная работа представляет собой комплексное проектирование и разработку дизайна маркетплейса, учитывающего потребности различных пользовательских ролей. Полученные результаты и перспективы дальнейшей разработки позволяют создать удобную и функциональную платформу, способствующую эффективному взаимодействию пользователей.

## **ЗАКЛЮЧЕНИЕ**

Разработанный программный продукт в значительной степени соответствует поставленным требованиям, сформулированным на этапе проектирования. Несмотря на отсутствие внешнего заказчика, продукт создавался в контексте реальных потребностей потенциальных пользователей — начинающих предпринимателей и владельцев малого бизнеса. Проведённый анализ пользовательских сценариев, а также сформулированные функциональные и нефункциональные требования позволили сфокусировать реализацию на тех компонентах, которые обеспечивают удобство взаимодействия, стабильность и адаптируемость системы. На всех этапах разработки соблюдалась логическая взаимосвязь между исходными задачами и итоговым функционалом, что свидетельствует о корректной постановке проектных целей и их успешной реализации.

Оценка качества программного продукта проводилась с опорой на результаты тестирования, охватывающего как модульный, так и интеграционный уровни. Были выявлены отдельные дефекты, в том числе связанные с некорректной обработкой пользовательского ввода. Однако анализ последствий показал, что выявленные ошибки не критичны и не влияют на базовую работоспособность системы. Напротив, своевременное устранение обнаруженных сбоев в процессе итеративной разработки продемонстрировало устойчивость архитектуры и высокую степень гибкости программной платформы, что стало возможным благодаря использованию принципов Agile и непрерывному контролю качества со стороны команды.

В ходе реализации проекта были определены ключевые направления для дальнейшего развития продукта. Прежде всего, это касается масштабирования функционала — расширение возможностей личных кабинетов пользователей, внедрение системы аналитики для продавцов, а также интеграция с внешними платёжными и логистическими сервисами. Также целесообразно сосредоточиться на улучшении адаптивности интерфейса, повышении

производительности при высокой нагрузке и углублённой проверке безопасности пользовательских данных. Продолжение работы над проектом может быть направлено на разработку полноценного MVP для тестирования в условиях реального использования и последующего сбора пользовательской обратной связи.

Таким образом, проведённая работа позволила не только достичь поставленных целей, но и заложить прочную основу для последующего развития проекта. Полученный опыт подтвердил эффективность выбранных организационных и технических решений, а также продемонстрировал значимость системного подхода при создании цифровых продуктов, ориентированных на конкретную аудиторию.

## **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. 10 советов по использованию компонентов в Figma [Электронный ресурс] // VK. – URL: <https://vk.com/@aldsgnpro-10-sovetov-po-ispolzovaniu-komponentov-v-figma> (дата обращения: 27.03.2025).
2. Что такое бэклог [Электронный ресурс] // Compass. – URL: <https://getcompass.ru/blog/posts/chto-takoe-beklog> (дата обращения: 05.04.2025).
3. Как составлять бэклог. Краткое руководство [Электронный ресурс] // GeekBrains. – URL: <https://gb.ru/blog/kak-sostavlyat-beklog-kratkoe-rukovodstvo/> (дата обращения: 08.04.2025).
4. Как провести анализ конкурентов [Электронный ресурс] // Яндекс Практикум. – URL: <https://practicum.yandex.ru/blog/kak-provesti-analiz-konkurentov/> (дата обращения: 13.04.2025).
5. Как создать продукт, который нужен людям: Product Discovery и JTBD [Электронный ресурс] // Habr. – URL: <https://habr.com/ru/articles/340774/> (дата обращения: 01.04.2025).
6. Как мы создавали фреймворк для запуска стартапов [Электронный ресурс] // Habr, Outlines Tech. – URL: [https://habr.com/ru/companies/outlines\\_tech/articles/765034/](https://habr.com/ru/companies/outlines_tech/articles/765034/) (дата обращения: 16.04.2025).
7. Node.js Best Practices [Электронный ресурс] // GitHub. – URL: <https://github.com/goldbergoni/nodebestpractices> (дата обращения: 19.04.2025).
8. Официальный сайт Koa.js [Электронный ресурс]. – URL: <https://koajs.com/> (дата обращения: 05.04.2025).
9. Пример использования Koa.js и REST API [Электронный ресурс] // Medium. – URL: <https://javascript.plainenglish.io/koa-js-and-rest-api-cd0770fbe916> (дата обращения: 10.04.2025).
10. Prisma ORM для Node.js [Электронный ресурс]. – URL: <https://www.prisma.io/> (дата обращения: 24.04.2025).
11. Node-postgres – PostgreSQL клиент [Электронный ресурс] // npm. – URL: <https://www.npmjs.com/package/node-postgres> (дата обращения: 24.04.2025).

URL: <https://www.npmjs.com/package/pg> (дата обращения: 28.04.2025).

12. Документация React [Электронный ресурс]. – URL: <https://react.dev/learn> (дата обращения: 19.04.2025).

13. Документация Vite [Электронный ресурс]. – URL: <https://vitejs.dev/> (дата обращения: 30.03.2025).

14. Обучение React [Электронный ресурс] // W3Schools. – URL: <https://www.w3schools.com/REACT/default.asp> (дата обращения: 02.04.2025).

15. Документация TypeScript [Электронный ресурс]. – URL: <https://www.typescriptlang.org/docs/> (дата обращения: 23.04.2025).

16. Документация Material UI [Электронный ресурс]. – URL: <https://mui.com/material-ui/> (дата обращения: 15.04.2025).