

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.

Хрушков Артем
Евгеньевич

Куратор: Фамилия И.О.

ученая степень, ученое звание, должность

Хрушков Артем
Евгеньевич

Студенты команды Patience Team

Фамилия И.О.

Шилова Валерия
Михайловна

Фамилия И.О.

Городничев Иван
Сергеевич

Фамилия И.О.

Шодыев Фирдавс
Джафарович

Фамилия И.О.

Порсев Михаил

Владимирович

Фамилия И.О.

Сабитова Регина

Дамировна

Екатеринбург, 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Основная часть	6
1.1 Информация о работе каждого участника	6
1.2 Backlog	6
1.3 Анализ аналогов	7
1.4 Архитектура продукта	10
1.4.1 Функциональная архитектура	10
1.4.2 Техническая архитектура	11
1.5 Процесс разработки	14
1.6 Планирование деятельности	14
ЗАКЛЮЧЕНИЕ	15
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	17

ВВЕДЕНИЕ

Целью проекта является разработка современной онлайн-платформы для покупки и продажи поношенной одежды, ориентированной на удобство, персонализацию и вовлечённость пользователей. Основные задачи включают:

- создание интуитивно понятного пользовательского интерфейса;
- реализацию эффективной системы поиска и фильтрации товаров;
- разработку личных кабинетов для покупателей и продавцов;
- внедрение инструментов для взаимодействия между пользователями;
- обеспечение удобного процесса оформления заказов с выбором доставки и оплаты;
- интеграцию функций персонализации и подписки на уведомления.

Проект актуален в контексте растущего интереса к устойчивому потреблению и экологичному образу жизни. Всё больше людей стремятся осознанно подходить к покупкам, продлевать жизненный цикл вещей и снижать негативное воздействие на окружающую среду. Повторное использование одежды является одним из самых доступных и эффективных способов реализовать эти принципы в повседневной жизни.

Особенно это актуально среди молодёжи, для которой одежда является важным средством самовыражения. При этом существует высокий интерес к брендовой и дизайнерской одежде, однако её покупка в новых коллекциях остаётся дорогостоящей. Секонд-хенд позволяет молодым людям приобретать стильные и оригинальные вещи по доступной цене, не отказываясь от желания выглядеть модно. Наша платформа отвечает на этот

запрос, предлагая удобный способ находить и покупать брендовые товары с историей.

Дополнительно проект решает проблему недостаточной удобности существующих решений: отсутствие эффективных фильтров, ограниченные возможности взаимодействия и персонализации часто затрудняют поиск нужных товаров и снижают интерес к платформам. Мы предлагаем современное решение, сочетающее принципы устойчивой моды с высоким уровнем цифрового сервиса.

Программный продукт предназначен для рынка электронной коммерции в сегменте поношенной одежды с акцентом на устойчивое потребление и доступность брендовой моды. Платформа ориентирована на:

- молодёжь, заинтересованную в стильной, брендовой одежде по сниженным ценам;
- частных лиц, желающих выгодно продавать и покупать вещи с историей;
- малые бизнесы и локальных продавцов, работающих в нише устойчивой моды и секонд-хенд;
- энтузиастов «slow fashion» и осознанного потребления, стремящихся минимизировать экологический след.

Платформа будет реализована как веб-сервис с возможностью последующей адаптации под мобильные устройства, обеспечивая удобный и персонализированный доступ к широкому ассортименту качественной поношенной одежды. По завершении проекта планируется:

- запуск полноценного прототипа платформы с ключевым функционалом;
- внедрение расширенной системы фильтрации и поиска товаров;
- реализация карточек товара с отзывами, фото и вариативностью;

- интеграция корзины, оформления заказов, истории покупок и отзывов;
- создание системы избранного и подписок;
- реализация базовых функций социальной вовлечённости и поддержки;
- положительные отклики от тестовой группы пользователей;
- подготовка платформы к публичному запуску и масштабированию.

Таким образом, проект позволит людям легко и с удовольствием приобретать брендовые вещи по доступной цене, поддерживая экологичный и ответственный подход к моде.

1 Основная часть

1.1 Информация о работе каждого участника

Тимлид создал и вёл доску с задачами, проводил спринты каждую неделю, вёл беклог проекта, наблюдал за работой каждого участника проекта.

Аналитик провёл системный анализ всего проекта, сделал архитектуру веб-сайта, подготовил все необходимые материалы для дизайнера.

Дизайнер создал весь макет веб-сайта, включая все материалы, которые давал аналитик.

Бэкенд разработчик предоставил документированное API через Swagger и вёл Changelog проекта.

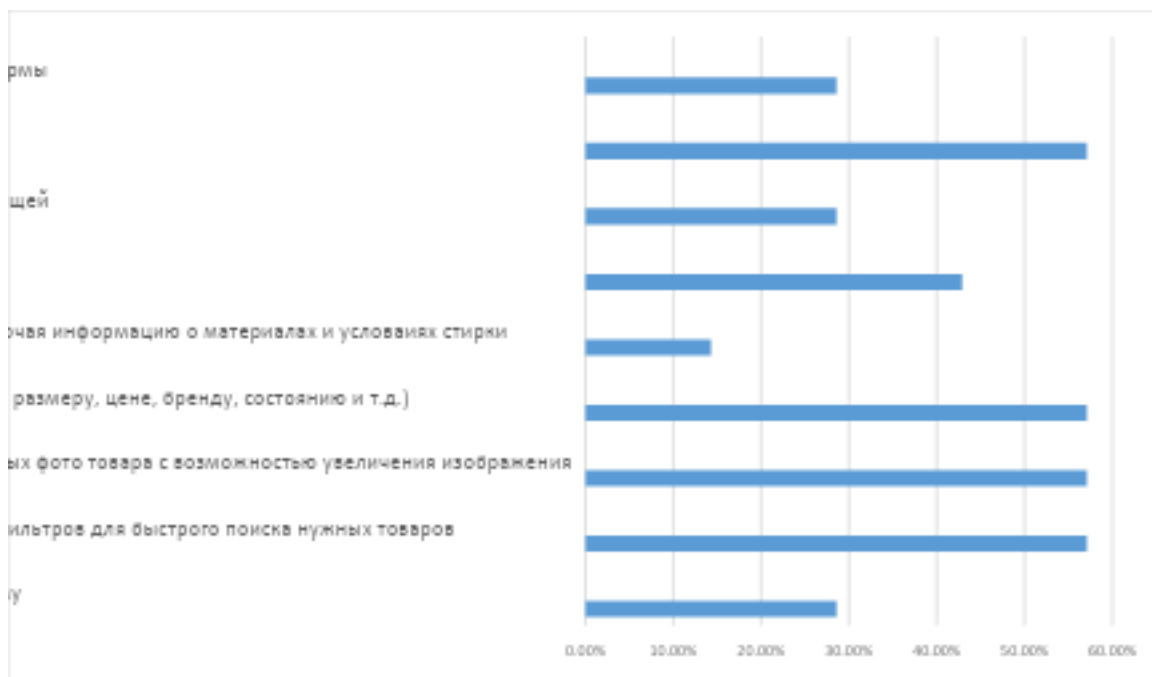
Фронтенд разработчик занимался вёрсткой сайта, включая все материалы, которые давал дизайнер.

1.2 Backlog

Для определения наиболее важных для пользователей функций интернет-магазина, был проведен опрос с помощью веб-сервиса Google Forms среди людей от 14 до 40 лет (1 Рисунок - Результаты опроса). По результатам опроса самыми важными функциями являются:

- Многообразие способов оплаты,
- Наличие системы сортировки товаров,
- Возможность просмотра детализированных фото товара с возможностью увеличения изображения;
- Удобство навигации по сайту и наличие фильтров для быстрого поиска нужных товаров.

Ещё многим пользователям важна функция сохранения избранных товаров.



1 Рисунок - Результаты опроса

Также был разработан план действий в виде беклога проекта, в нём собирались все важные и главные задачи, которые выполнялись в учрежденные сроки.

1.3 Анализ аналогов

В рамках разработки платформы для интернет-магазина поношенной одежды важно провести тщательный анализ существующих аналогов на рынке. Это позволит выявить как успешные решения, так и потенциальные недостатки, что в свою очередь поможет создать более конкурентоспособный и привлекательный продукт.

Будут рассмотрены несколько известных платформ, специализирующихся на секонд-хенде, таких как Baskoparty, Ozhur, Avekoo, Moda-Gorod, Mhand и Planeta-sh. Каждая из этих платформ предлагает уникальные функции и идеи, которые могут быть адаптированы к проекту.

Baskoparty имеет не очень приятный дизайн, много ярких деталей которые режут глаза и создают дискомфорт. У платформы имеется удобный интерфейс и возможность фильтрации товаров по категориям, что улучшает пользовательский опыт. Также отмечено наличие раздела с историями о каждом предмете, что создает эмоциональную связь покупателя и товара. Важно рассмотреть адаптацию такого подхода, который увеличивает ценность каждого предмета и способствует более глубокому взаимодействию с клиентами. Ещё стоит учесть цветовую гамму платформы чтобы не повторять ошибок в дизайне.

Ozhug замечательна своей продуманной навигацией и возможностью быстро находить нужные товары. Платформа имеет удобные фильтры, которые позволяют сортировать продукцию по нескольким критериям, таким как размер, цвет и тип одежды. Это следует учесть в проекте, добавив аналогичные механизмы для удержания клиентов.

Avekoo выделяется функцией «wish list», которая позволяет пользователям сохранять интересующие товары для дальнейшего просмотра. Этот элемент может быть интегрирован в разрабатываемую платформу, давая возможность пользователям отслеживать свои предпочтения и получать уведомления о снижении цен или поступлении товара в наличии.

Moda-Gorod предлагает разнообразные товары с качественными фотографиями и подробными описаниями, что значительно улучшает вероятность покупки. Возможность просматривать несколько изображений товара с разных ракурсов будет способствовать более взвешенному выбору покупателя. Также важно обратить внимание на разделы с отзывами и рейтингами, помогающие создать доверие к продукту.

Mhand имеет функцию сравнения товаров и опцию «купить похожее», что может увеличить объемы продаж. Также стоит рассмотреть реализацию раздела с акциями, который будет стимулировать внимание к специфическим товарам или категориям. Mhand также имеет некоторые минусы. Во-первых, избыточная реклама на сайте может отвлекать пользователей от основного

контента. Это создает неудобства и может привести к снижению конверсии. Кроме того, усложненная навигация по платформе иногда приводит к затруднениям в поиске нужного товара, что отрицательно сказывается на пользовательском опыте. К тому же отсутствие четкой информации о возврате товаров может вызывать недовольство у покупателей.

Planeta-sh имеет сильную мобильную версию сайта, что особенно актуально в условиях современной цифровизации и роста числа мобильных пользователей. Важно разработать адаптивный дизайн для интернет-магазина, обеспечивающий удобство использования на любых устройствах. Возможность загрузки приложения или полноценной мобильной версии может повысить доступность платформы. Planeta-sh также имеет свои недостатки. Несмотря на наличие удобного мобильного интерфейса, некоторые пользователи отмечают, что он не всегда оптимизирован для всех устройств, что может привести к проблемам отображения и навигации. Кроме того, планируемая система личного кабинета и функции, связанные с управлением заказами, иногда работают некорректно, что создаёт дополнительные трудности для клиентов. Наконец, ограниченные возможности фильтрации и поиска товаров могут привести к тому, что покупатели уйдут, не найдя то, что им нужно.

Изучая данные платформы, можно выделить несколько ключевых решений и фишек, которые могут быть успешно внедрены в проект. Создание уникального пользовательского интерфейса, развитие системы лояльности, использование качественного контента, акцент на мобильность и улучшение функций взаимодействия с клиентами — это те направления, которые помогут интернет-магазину поношенной одежды выделиться на фоне конкурентов и обеспечить высокий уровень удовлетворенности клиентов. Также при разработке важно учитывать недостатки других платформ. Следует избегать избыточной рекламы, упрощать навигацию, обеспечивать четкую политику возврата и делать приятный глазу дизайн.

Оптимизация функций личного кабинета и системы поиска помогут создать более приятный опыт для пользователей.

1.4 **Архитектура продукта**

1.4.1 **Функциональная архитектура**

Главная страница:

- Баннеры акций и новинок,
- Категории товаров,
- Поисковая строка,
- Кнопки «Войти» /
«Зарегистрироваться».

Каталог товаров:

- Список товаров с возможностью фильтрации по категориям, размерам, цвету, цене;
- Карточки товаров с изображениями, названиями и ценами.

Страница товара:

- Подробные фотографии товара,
- Описание и характеристики,
- Отзывы покупателей,
- Выбор размера, цвета и количества;
- Кнопки «Добавить в корзину» и «Добавить в избранное».

Корзина:

- Список выбранных товаров с возможностью изменения количества или удаления;
- Отображение общей стоимости заказа;
- Кнопка «Оформить заказ».

Оформление заказа:

- Форма ввода контактных данных;
- Выбор способа доставки и оплаты;
- Подтверждение заказа.

Регистрация и авторизация:

- Форма регистрации с вводом email, логина и пароля;
- Форма входа для авторизации существующих пользователей.

Личный кабинет:

- Просмотр истории заказов,
- Управление личными данными,
- Добавление и редактирование отзывов.

Избранное:

- Список товаров, добавленных в избранное;
- Возможность удаления товаров из избранного.

Отзывы:

- Форма добавления отзыва о товаре;
- Просмотр отзывов других пользователей.

1.4.2 Техническая архитектура

Frontend (Клиентская часть):

- реализован как одностраничное приложение (SPA);
- использует современные JavaScript-фреймворки;
- состоит из следующих компонентов:
 - а) главная страница, каталог, карточка товара;
 - б) поисковая система с фильтрацией;
 - в) корзина, избранное, оформление заказа, оплата;
 - г) личный кабинет пользователя, регистрация, вход, история заказов, отзывы;
 - д) форма обратной связи.

Backend (Серверная часть):

- построен на платформе ASP.NET Core 9.0;
- реализован на языке C#;
- архитектура разделена на следующие слои:
 - а) presentation layer (контроллеры API),
 - б) application layer (бизнес-логика),
 - в) domain layer (бизнес-модели и доменные правила),

г) infrastructure layer (доступ к БД и внешним API).

— основные модули:

а) аутентификация и авторизация (JWT, с возможностью интеграции OAuth2),

б) каталог товаров,

в) категории товаров,

г) корзина,

д) модуль оплаты (интеграция с платёжным шлюзом ЮMoney).

Преимущества выбранной архитектуры:

— слабая связанность
между слоями,

— упрощённое
тестирование,

— расширяемость и
модульность.

В качестве Data Access Layer используется Entity Framework Core (Code First), имеет следующие преимущества:

— упрощённая работа с
базой данных,

— поддержка миграций,

— гибкая настройка связей
и индексов,

— отличная интеграция с
ASP.NET Core.

Работа с API:

— REST API с поддержкой стандартных HTTP-методов (GET, POST, PUT, DELETE),

— чётко структурированные URL-адреса,

— единообразные коды ответов и форматы ошибок.

DTO (Data Transfer Objects):

- используются для разделения моделей базы данных и данных, передаваемых через API;
- повышают безопасность и минимизируют объём передаваемой информации.

Логирование и мониторинг интегрируются с Serilog (структурированное логирование).

База данных:

- реляционная СУБД: PostgreSQL 17.5,
- хранит информацию о:
 - а) пользователях,
 - б) товарах и категориях,
 - в) заказах и корзине,
 - г) изображениях товаров,
 - д) электронных чеках.

Административная панель предусмотрена для управления товарами, заказами и пользователями.

Интеграции:

- платёжные системы (ЮMoney),
- возможность расширения (например, OAuth2: Google, Яндекс).

Выбор ASP.NET Core обусловлен:

- высокой производительностью,
- поддержкой и регулярными обновлениями от Microsoft,
- перспективностью платформы и интересом команды разработки.

Многоуровневая архитектура обеспечивает удобную поддержку, масштабируемость и гибкое расширение функциональности.

1.5 Процесс разработки

Используется многоуровневая архитектура и модульный подход, что упрощает процесс разработки, отладки и тестирования. Применяются современные практики:

- разделение бизнес-логики и инфраструктуры;
- использование DTO для повышения безопасности и оптимизации передачи данных;
- поддержка миграций при работе с БД.

На промежуточных этапах проводилось тестирование. Были выявлены и устранены ошибки, связанные с валидацией данных, доступностью API, корректностью бизнес-логики. Логирование и отслеживание ошибок осуществляется через Serilog. Все модули можно независимо разрабатывать, тестировать и развивать, что облегчает масштабирование и сопровождение проекта.

1.6 Планирование деятельности

С первых этапов проекта за контроль и организацию разработки отвечал тимлид. Он активно координировал работу команды, включая:

- проведение регулярных встреч с командой для обсуждения текущего статуса и приоритетов задач;
- индивидуальные созвоны с каждым участником команды для оценки прогресса, выявления возможных затруднений и предоставления необходимой поддержки;

- планирование спринтов и составление графика работ;
- постановка и распределение задач через Яндекс.Трекер с последующим мониторингом их выполнения;
- ведение отчетности о проделанной работе и достижении промежуточных целей.

ЗАКЛЮЧЕНИЕ

В результате разработки программного продукта большая часть ключевых требований, изложенных в техническом задании, была успешно реализована. Основной стек технологий и архитектурные принципы соответствуют современным практикам и задачам проекта.

Архитектура приложения построена по принципу разделения на слои:

- presentation layer (контроллеры API),
- application layer (бизнес-логика),
- domain layer (бизнес-модели и доменные правила),
- infrastructure layer (доступ к БД и внешним API).

Данный подход обеспечивает слабую связанность компонентов, упрощает тестирование и поддержку проекта, а также облегчает масштабирование и развитие продукта в будущем.

Выбор ASP.NET Core для реализации бэкенда оправдан его высокой производительностью, активной поддержкой и стремлением команды развиваться на этой платформе. В качестве ORM используется EF Core с подходом Code First, что позволило наладить удобную работу с базой данных PostgreSQL и внедрить миграции для управления схемой данных.

Аутентификация и авторизация реализованы через JWT, что обеспечивает безопасное взаимодействие между клиентом и сервером. Применение DTO обеспечивает изоляцию между слоями и минимизацию утечки чувствительных данных.

Проект организован в модули, соответствующие бизнес-функциям: товары, корзина, оплата, категории, пользователи. Это упрощает управление кодовой базой и способствует удобному тестированию и расширению функционала.

Клиентская и административная части взаимодействуют с бэкендом через REST API с корректной маршрутизацией, четкой структурой URL-ов и единообразными кодами ответов. Используются стандартные HTTP-методы, что способствует предсказуемости и совместимости.

Несмотря на то, что часть требований микросервисной архитектуры, контейнеризации и ведения Changelog осталась на этапе планирования, заложенные принципы и структура проекта позволяют реализовать данные аспекты на следующих этапах.

В процессе промежуточного тестирования и отладки были выявлены и устранены ошибки в бизнес-логике, маршрутизации и взаимодействии компонентов. Логика приложения построена таким образом, чтобы изолировать критические компоненты и упростить их проверку.

Также стоит отметить, что проект реализован с учётом принципов разделения ответственности и слабой связанности, что в значительной степени снизило количество технических дефектов.

Однако в текущей версии продукта наблюдается частичное несоответствие архитектурной модели (микросервисной) — приложение выполнено по слоистой монолитной архитектуре. Это объясняется сжатыми сроками и ограниченным составом команды. Несмотря на это, текущая структура кода адаптируема под микросервисный подход при необходимости.

В итоге реализована основная бизнес-логика магазина секонд-хенда, включая управление товарами, корзиной, оплатой и авторизацией.

Архитектура проекта адаптивна, хорошо структурирована и соответствует принципам SOLID и DDD.

Дизайн-подход отвечает требованиям удобства, визуальной достоверности и специфике предметной области.

Необходима доработка микросервисного подхода и инфраструктурных требований на следующем этапе.

Потенциал проекта высок: возможна интеграция внешних платёжных систем, масштабирование и подключение новых сервисов без кардинальных изменений архитектуры.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. <https://practicum.yandex.ru/blog/chtotakoe-user-flow-i-zachem-ego-razrabatyvat/#chtotakoe>
2. <https://on-target.pro/blog/sozdanie-saytov/struktura-internet-magazina/>
3. <https://mwi.me/blog/struktura-internet-magazina-kak-pravilno-ee-sformirovat/>
4. <https://practicum.yandex.ru/blog/chtotakoe-user-story-i-kak-napisat/>
- 5.