Частное учреждение образования

«Колледж бизнеса и права»

«Разработка веб ‒ приложения ДЛЯ ПУБЛИКАЦИИ, ПРОДАЖИ И ВЗАИМОДЕЙСТВИЯ С ЦИФРОВЫМИ ИЗОБРАЖЕНИЯМИ «Online gallery»

Пояснительная записка

к курсовому проекту по дисциплине

«Конструирование программ и языки программирования»

КП Т.318017.401

Руководитель проекта (А.В.Кривошеина)

Учащийся (А. А. Примшиц)

2025

Изм.

Лист

№ докум.

Подпись

Датаа

Лист

3

КП Т.318017.401

Разраб.

Примшиц А. А.

Провер.

Кривошеина А.В

Реценз.

Н. Контр.

Утверд.

*Разработка веб-приложения для публикации, продажи и взаимодействия с цифровыми изображениями «OnlineGallery»*

Лит.

Листов

КБиП

у

СОДЕРЖАНИЕ

[Введение 4](#_Toc200978271)

[1 Описание задачи 6](#_Toc200978272)

[1.1 Анализ предметной области 6](#_Toc200978273)

[1.2 Постановка задачи 7](#_Toc200978274)

[2 Проектирование веб-приложения 9](#_Toc200978275)

[2.1 Проектирование модели 9](#_Toc200978276)

[2.2 Требования к веб-приложению 10](#_Toc200978277)

[2.3 Структура веб-приложения 11](#_Toc200978278)

[2.4 Проектирование макета веб-приложения 12](#_Toc200978279)

[2.5 Программно-технические средства, необходимые для разработки приложения 13](#_Toc200978280)

[2.6 Защита и сохранность данных 15](#_Toc200978281)

[2.7 Организация и ведение информационной базы (модели) 15](#_Toc200978282)

[3 Реализация веб-приложения 22](#_Toc200978283)

[3.1 Описание разделов веб-приложения 22](#_Toc200978284)

[3.2 Разработка административной части приложения 23](#_Toc200978285)

[3.3 Разработка клиентской части приложения 24](#_Toc200978286)

[3.4 Описание используемых функций и процедур 25](#_Toc200978287)

[3.5 Функциональное тестирование 28](#_Toc200978288)

[4 Применение 30](#_Toc200978289)

[4.1 Назначение веб-приложения 30](#_Toc200978290)

[4.2 Программно-аппаратное обеспечение сервера и клиента 30](#_Toc200978291)

[Заключение 32](#_Toc200978292)

[Список информационных источников 33](#_Toc200978293)

[Приложения А Ментальная карта 34](#_Toc200978294)

[Приложение Б Логическая и физическая структура 35](#_Toc200978295)

[Приложение В Макеты страниц веб – приложения 37](#_Toc200978296)

[Приложение Г Код модулей веб – приложения 46](#_Toc200978297)

[Приложение Д Результаты работы веб – приложения 67](#_Toc200978298)

# Введение

Целью курсового проекта на тему «Разработка веб-приложения Online Gallery» является создание полнофункционального программного средства, обеспечивающего автоматизацию процессов размещения, демонстрации, покупки и взаимодействия с изображениями в цифровом формате. Разрабатываемая система представляет собой универсальное веб-приложение, объединяющее функции онлайн-галереи и магазина цифровых изображений, предназначенное как для авторов графических материалов, так и для конечных пользователей, заинтересованных в приобретении визуального контента.

Современные тенденции развития цифровых сервисов требуют создания надёжных, гибких и функциональных решений, которые обеспечивают высокое качество пользовательского взаимодействия, безопасность и удобство в использовании. Web-платформа OnlineGallery отвечает этим требованиям, предлагая интуитивно понятный интерфейс с поддержкой таких функций, как регистрация и авторизация, просмотр каталога изображений, добавление работ в избранное, комментирование, оформление покупок, просмотр чеков, а также управление личным профилем. Пользователь может в любой момент просмотреть историю своих действий, загрузить ранее купленные изображения, а также отредактировать личную информацию.

Отдельное внимание в проекте уделено административной части веб-приложения, в рамках которой реализовано централизованное управление данными: администраторы могут редактировать категории, просматривать информацию о пользователях, анализировать транзакции, а также формировать отчёты в формате Word. Такой подход позволяет обеспечить надёжную поддержку бизнес-логики, контроль за содержанием галереи и эффективную модерацию пользовательской активности.

Актуальность проекта обусловлена растущим спросом на цифровые сервисы в сфере графики, дизайна и иллюстрации. Многие авторы заинтересованы в демонстрации и продаже своих работ в интернете, а покупатели — в удобном и безопасном способе их приобретения. При этом немаловажным является аспект персонализации взаимодействия, адаптивность интерфейса под разные устройства, а также высокая скорость обработки данных. В условиях цифровизации культуры и искусства платформа OnlineGallery способна занять достойное место среди аналогичных решений, предлагая удобный, доступный и функционально насыщенный сервис.

В рамках проекта реализован широкий спектр функциональных возможностей как для обычных пользователей, так и для администраторов. Приложение построено на базе клиент-серверной архитектуры, с применением современных технологий ASP.NET Core MVC, Razor Pages, SQL Server и HTML/CSS. Это позволило создать стабильную, масштабируемую и безопасную систему, полностью соответствующую требованиям курсового проектирования и отражающую ключевые принципы разработки программных решений для сферы электронной коммерции.

Материалы проекта представлены в виде пояснительной записки, включающей все этапы: от постановки задачи и анализа предметной области до проектирования, программной реализации и тестирования системы. Кроме текстовой части, прилагается графическая документация, содержащая диаграммы прецедентов, классов, развёртывания и структуру базы данных, а также макеты основных страниц веб-приложения. В приложениях приведены примеры кода и результаты тестирования, подтверждающие работоспособность и корректность реализованного функционала.

# Описание задачи

## Анализ предметной области

Темой курсового проекта является разработка веб-приложения «**Online Gallery»,** предназначенного для публикации, продажи и обсуждения цифровых изображений. Предметной областью выступает сфера цифрового контента и электронной коммерции, ориентированная на взаимодействие между авторами визуальных материалов и конечными пользователями, заинтересованными в приобретении графических работ. Система реализует основные функции платформы цифровой галереи с элементами онлайн-магазина.

Основными объектами предметной области являются:

* + изображения с описанием, названием, ценой и метаданными (автор, категория, формат);
  + категории изображений и система фильтрации для удобного поиска;
  + учётные записи пользователей (личные кабинеты авторов и покупателей);
  + транзакции (покупки изображений с фиксацией способа оплаты);
  + система избранного (отметки понравившихся работ);
  + административный интерфейс для управления контентом и пользователями.

Бизнес-процессы, реализуемые в системе, охватывают цикл взаимодействия с изображениями:  
пользователь проходит регистрацию или авторизацию, просматривает галерею изображений, применяет фильтры по категории или ключевым словам, добавляет изображения в избранное или оформляет покупку. После успешной оплаты изображение становится доступным для скачивания в личном кабинете. Пользователь также может оставить комментарий или изменить данные своего профиля. Администратор через закрытую панель управления получает доступ к операциям с изображениями, категориями, пользователями и транзакциями, а также может выгружать отчёты.

Необходимость автоматизации обусловлена постоянным ростом объёма графического контента, потребностью в безопасной продаже изображений, а также необходимостью централизованного хранения данных и оперативной обработки заказов. Web-приложение «Online Gallery» упрощает взаимодействие между авторами и покупателями, обеспечивает безопасность операций и позволяет масштабировать платформу при необходимости.

Ключевые задачи, подлежащие реализации в рамках проекта:

* + проектирование и реализация базы данных для хранения информации о пользователях, изображениях, категориях, покупках, комментариях и избранных записях;
  + реализация системы регистрации, авторизации и защиты персональных данных;
  + создание контроллеров и API для управления изображениями, комментариями, заказами, избранным и административными действиями;
  + разработка адаптивного пользовательского интерфейса на Razor Pages с поддержкой всех базовых сценариев: просмотр, покупка, избранное, профиль;
  + реализация экспорта чеков и отчётов по транзакциям в формате Word;
  + интеграция с системой ролей (администратор, пользователь) и реализация соответствующего разграничения прав доступа.

Целевую аудиторию веб-приложения составляют:

* авторы, художники, дизайнеры и фотографы, публикующие и продающие свои работы;
* частные пользователи, заинтересованные в покупке изображений для личного или коммерческого использования;
* администраторы, осуществляющие контроль за контентом и пользователями;
* специалисты, использующие изображения в оформлении, публикациях или веб-разработке.

На рынке существует ряд аналогичных решений, таких как Etsy (категория «Digital Art»), Shutterstock, DeviantArt, однако большинство из них требует либо интеграции со сторонними сервисами, либо не предоставляет прямого взаимодействия с авторами или доступа к исходным файлам. Преимущество «Online Gallery» заключается в полной управляемости, модульной архитектуре и возможности последующего масштабирования платформы с учётом особенностей работы с авторским цифровым контентом.

## Постановка задачи

Основная задача разработки веб-приложения «Online Gallery» заключается в создании современного, функционального и безопасного инструмента для размещения, покупки и обсуждения цифровых изображений, с возможностью управления контентом и пользователями в единой информационной системе. Приложение должно объединять удобство социальной платформы и надёжность интернет-магазина, обеспечивая авторам и покупателям максимально комфортный и прозрачный процесс взаимодействия.

С помощью разрабатываемого программного обеспечения решаются следующие задачи в рамках предметной области:

* **автоматизация отображения и фильтрации изображений** — пользователям предоставляется удобная галерея с возможностью поиска по категориям, ключевым словам и авторам, что облегчает навигацию и ускоряет нахождение нужного материала;
* **покупка изображений и управление транзакциями** — реализуется система оформления заказов с возможностью выбора способа оплаты и последующего получения изображения через личный кабинет;
* **система избранного** — пользователи могут добавлять понравившиеся изображения в список избранных, чтобы возвращаться к ним позднее;
* **личный кабинет и история покупок** — каждому пользователю доступен персональный профиль с возможностью редактирования данных, просмотра истории покупок и загрузки купленных файлов;
* **администрирование платформы** — через специальную панель администраторы управляют категориями, пользователями, изображениями и получают доступ к отчётам по продажам, скачиваемым в формате Word;
* **обеспечение безопасности** — реализована авторизация с разграничением прав (пользователь, администратор), а также защита персональных данных и шифрование ключевой информации;
* **адаптивный и интуитивный интерфейс** — все страницы корректно отображаются как на компьютерах, так и на мобильных устройствах благодаря использованию Razor Pages и адаптивной верстке.

На основе анализа предметной области были выделены следующие ключевые функции, которые подлежат автоматизации:

* + публикация и фильтрация изображений по категориям, авторам и ключевым словам;
  + оформление покупки с безопасной оплатой и генерацией чека;
  + регистрация, авторизация и управление профилем пользователя;
  + добавление изображений в избранное;
  + хранение и отображение истории покупок;
  + администрирование пользователей, изображений и категорий;
  + экспорт отчётов и чеков в формате Word;
  + защита персональных данных, разграничение прав доступа и логирование действий.

Среди существующих решений можно выделить такие платформы, как **Shutterstock**, **Etsy** (в категории цифровых товаров), **Behance** и **DeviantArt**. Однако большая часть из них либо фокусируется исключительно на витринной функции, либо требует интеграции с внешними модулями для полноценной продажи изображений. Более того, такие платформы не всегда позволяют гибко управлять контентом и получать экспортируемые отчёты по транзакциям.

Разработка собственного веб-приложения «**Online Gallery»** позволяет учесть все технические и организационные требования, обеспечить контроль над внутренней логикой системы и создать безопасную, масштабируемую платформу с возможностью индивидуального расширения функционала в будущем.

# Проектирование веб-приложения

## Проектирование модели

Диаграмма прецедентов веб-приложения «COOLZY» отражает структуру взаимодействия пользователей с системой, включая администратора и покупателя. Она демонстрирует все основные действия, доступные каждому типу пользователя, а также зависимости между ними.

Покупатель начинает взаимодействие с системой через регистрацию и авторизацию, после чего получает доступ к ключевому функционалу: просмотру каталога товаров, поиску и фильтрации, оформлению заказа и добавлению товара в корзину. Добавление товара встроено в оформление заказа и связано через механизм include. Также к оформлению заказа может быть подключён выбор способа доставки, реализованный через extend.

Администратор имеет расширенные полномочия: управление заказами, каталогом товаров и пользователями. Эти действия позволяют администрировать систему, контролировать содержимое и поведение пользователей.

Диаграмма прецедентов представлена на листе 1 графической части.

Диаграмма классов отображает внутреннюю структуру приложения «COOLZY» и показывает основные сущности, их свойства и связи между ними. Она необходима для построения базы данных, а также для описания логики работы системы.

Класс Пользователь содержит поля: userID, username, email, password, isAdmin. Один пользователь может иметь одну корзину и оформлять заказы. Также реализованы методы: register(), login(), logout(), updateProfile().

Класс Корзина содержит поля: cartID, user, items, где items — список объектов типа CartItem. Методы: addItem(), removeItem(), clearCart(), checkout(). Связан с пользователем и с товаром.

Класс Товар включает: productID, name, description, price, stockQuantity, category. Методы: updateStock(), getDetails(). Связан с категорией и участвует в заказах и корзине.

Класс Категория: categoryID, name, description. Методы: addProduct(), removeProduct(), listProducts().

Класс Заказ: orderID, user, orderDate, totalAmount, status. Методы: calculateTotal(), updateStatus(), cancelOrder(). Один пользователь может иметь множество заказов.

Диаграмма классов представлена на листе 2 графической части.

Диаграмма развёртывания веб-приложения «COOLZY» иллюстрирует архитектуру системы и распределение компонентов между клиентской, серверной и облачной частями. Она отражает потоки данных, направления обмена информацией и назначение каждого узла.

На стороне клиентского узла расположен веб-браузер с адаптивным интерфейсом, реализованным с помощью React. Через протокол HTTPS осуществляется обмен с сервером.

Серверный узел включает в себя веб-сервер, на котором размещено серверное приложение на ASP.NET Core. Логика сервера взаимодействует с базой данных через SQL-запросы.

База данных развернута на MySQL/SQL Server и хранит информацию о пользователях, товарах, заказах, корзинах и категориях.

Для размещения проекта используется облачный узел — Vercel или Azure, куда осуществляется деплой серверного и клиентского кода.

Диаграмма развёртывания представлена на листе 3 графической части.

## Требования к веб-приложению

При разработке веб-приложения «**Online Gallery»** необходимо учитывать совокупность требований, охватывающих визуальный стиль, графику, функциональность и технические характеристики. Особое внимание уделяется созданию минималистичного и интуитивно понятного интерфейса, ориентированного как на авторов изображений, так и на конечных пользователей, просматривающих и покупающих цифровой контент.

Графическое оформление должно отражать современность, визуальную чистоту и лёгкость восприятия. Использование высококачественных изображений, понятных иконок и акцентных элементов помогает создать ощущение профессиональной среды и повышает уровень доверия со стороны пользователей. Все управляющие элементы — кнопки, фильтры, поля поиска, меню — должны быть чётко обозначены и удобно сгруппированы, а каждый логический блок — визуально отделён.

Визуальный стиль интерфейса выполнен в светлой цветовой гамме с акцентами сине-фиолетового спектра, что подчёркивает цифровую тематику проекта. Основной шрифт — **Montserrat**, современный, геометричный и хорошо читаемый на разных экранах. Используются до трёх начертаний: для основного текста, заголовков и акцентных элементов. Цвет текста контрастирует с фоном, а важные элементы выделяются жирным шрифтом или акцентным цветом.

Приложение должно корректно отображаться и функционировать во всех популярных браузерах (Google Chrome, Mozilla Firefox, Safari, Microsoft Edge), а также адаптироваться под различные разрешения экранов: настольные ПК, планшеты и смартфоны. Это обеспечит универсальный доступ к галерее независимо от устройства пользователя.

Контент платформы должен быть актуальным и структурированным. В карточках изображений отображаются название работы, автор, стоимость, категория, а также визуальный предпросмотр. Каждое изображение сопровождается подробной карточкой с описанием, тегами, комментариями и кнопками взаимодействия (купить, добавить в избранное, оставить отзыв).

Для администраторов предусмотрена система управления контентом (CMS), обеспечивающая добавление, редактирование и удаление изображений, категорий, а также контроль над пользовательской активностью. Это позволяет гибко адаптировать структуру галереи под нужды проекта.

Обязательными являются функции регистрации и авторизации, реализация ролей (пользователь / администратор), безопасное хранение данных и защита личной информации. Навигация должна быть простой и логичной, а все ключевые действия — доступны с любой страницы.

Требования к оформлению интерфейса «Online Gallery»:

**Шрифт** — Montserrat:

* + Light — основной текст;
  + Regular — универсальные элементы интерфейса;
  + SemiBold / Bold — заголовки и акценты.

**Размеры шрифтов**:

* + Заголовки 1 уровня — 24px
  + Заголовки 2 уровня — 20px
  + Заголовки 3 уровня — 18px
  + Основной текст — 16px
  + Подписи и системный текст — 14px

**Цветовая палитра**:

* + Основной текст: чёрный #000000, белый #FFFFFF
  + Фон: светло-серый #F5F5F5
  + Шапка/меню: тёмно-серый с оттенком фиолетового #343a40
  + Фон карточек: белый #FFFFFF, фон блоков — #E8EAE5
  + Акцентные цвета и кнопки: синий #3563E9, фиолетовый #5B6B82

Такая система визуального и функционального оформления позволяет сформировать единый, целостный и визуально привлекательный облик веб-приложения «**Online Gallery»**, обеспечивая положительный пользовательский опыт и лёгкость взаимодействия с системой.

## Структура веб-приложения

Структура веб-приложения «**OnlineGallery»** представляет собой логически упорядоченную схему, отражающую взаимодействие между основными компонентами системы и пользователями. Она демонстрирует, каким образом организована работа всех функциональных блоков, начиная от интерфейса пользователя и заканчивая взаимодействием с базой данных.

Каждый компонент выполняет строго определённую роль в обеспечении работоспособности всей платформы: просмотр изображений, авторизация, добавление в избранное, оформление покупки, управление контентом и т.д. Структура также определяет логику переходов между страницами, обработку событий и взаимодействие между фронтендом и серверной частью приложения.

Пользовательский поток построен по принципу интуитивной навигации: главная страница → каталог изображений → карточка изображения → действия (купить / добавить в избранное / прокомментировать) → оформление → история заказов и личный кабинет. Администратор работает в отдельной части приложения, с доступом к функциям управления контентом и пользователями.

Общая схема структуры веб-приложения «**Online Gallery»** представлена в **Приложении А** на **рисунке А.1** в виде ментальной карты.

## Проектирование макета веб-приложения

Веб-сайт был спроектирован в онлайн-сервисе Figma, который предоставляет мощные инструменты для прототипирования и проектирования интерфейсов. С помощью Figma были созданы интерактивные элементы, включая кнопки, формы, меню и векторные изображения, что позволило реализовать современный и удобный дизайн.

Сайт использует шрифт Montserrat, который обеспечивает отличную читаемость и поддерживает современный стиль дизайна. Шрифт применяется для всех текстов, включая заголовки, описания и подписи к кнопкам.

Дизайн сайта имеет в основном белый, голубой, светло-фиолетовый цвета. Он был спроектирован таким образом, чтобы удовлетворить интересы пользователей. Это улучшает качество пользовательского опыта.

Размер макета был выбран исходя из размеров окна браузера.

Главная страница сайта имеет размер 1920 пикселей по ширине и 4899 по высоте.

Раздел «Каталог товаров» имеет размер 1920 пикселей по ширине и 2940 по высоте.

Раздел «Карточка товара» имеет размер 1920 пикселей по ширине и 2660 по высоте.

Раздел «Избранное» имеет размер 1920 пикселей по ширине и 2020 по высоте.

Раздел «Корзина (с товарами)» имеет размер 1920 пикселей по ширине и 1065 по высоте.

Раздел «Корзина (пустая)» имеет размер 1920 пикселей по ширине и 1080 по высоте.

Раздел «Оформление заказа» имеет размер 1920 пикселей по ширине и 1300 по высоте.

Раздел «Спасибо за заказ» имеет размер 1920 пикселей по ширине и 1080 по высоте.

Раздел «Привязка новой карты» имеет размер 1920 пикселей по ширине и 1080 по высоте.

Раздел «Политика персональных данных» имеет размер 1920 пикселей по ширине и 6070 по высоте.

Форма авторизации имеет размер 1920 пикселей по ширине и 1080 по высоте.

Форма регистрации имеет размер 1920 пикселей по ширине и 1080 по высоте.

Раздел «Профиль» имеет размер 1920 пикселей по ширине и 1280 по высоте.

Раздел «Ошибка 404» имеет размер 1920 пикселей по ширине и 1080 по высоте.

На приложении В на рисунке В.1 представлен макет главной страницы, на которой размещены баннер, категории товаров и блок акций.

На приложении В на рисунке В.2 представлен макет страницы «Каталог товаров», содержащей фильтры и карточки товаров.

На приложении В на рисунке В.3 представлен макет страницы «Карточка товара», содержащей описание, характеристики и отзывы.

На приложении В на рисунке В.4 представлен макет страницы «Избранное», отображающей товары, добавленные пользователем.

На приложении В на рисунке В.5 представлен макет страницы «Корзина (с товарами)», отображающей список товаров и сумму заказа.

На приложении В на рисунке В.6 представлен макет страницы «Корзина (пустая)», которая отображается, если пользователь ещё не добавил товары.

На приложении В на рисунке В.7 представлен макет страницы «Оформление заказа», содержащей формы ввода адреса и выбора способа оплаты.

На приложении В на рисунке В.8 представлен макет страницы «Спасибо за заказ», подтверждающей оформление и номер заказа.

На приложении В на рисунке В.9 представлен макет страницы «Привязка новой карты», содержащей форму для добавления платёжных данных.

На приложении В на рисунке В.10 представлен макет формы авторизации, содержащей поля ввода e-mail и пароля.

На приложении В на рисунке В.11 представлен макет формы регистрации, содержащей поля для заполнения личных данных.

На приложении В на рисунке В.12 представлен макет страницы «Профиль», где отображаются личные данные пользователя.

## Программно-технические средства, необходимые для разработки приложения

Для реализации веб-приложения COOLZY используется современный стек технологий, включающий языки программирования, фреймворки, библиотеки и инструменты разработки, которые обеспечивают высокую производительность, надёжность и адаптивность системы. Приложение ориентировано на клиент-серверную архитектуру с поддержкой сетевого взаимодействия, регистрации пользователей, отображения товаров и обработки заказов в режиме онлайн.

С помощью выбранных технологий решаются следующие задачи в рамках реализации проекта:

* построение серверной логики приложения. Для разработки backend-части используется язык программирования C# с фреймворком ASP.NET Core MVC [5]. Он выбран благодаря своей высокой надёжности, масштабируемости, встроенной поддержке безопасной авторизации и удобной интеграции с базами данных. ASP.NET Core позволяет строить REST API, что идеально подходит для взаимодействия с клиентской частью;
* создание динамичного интерфейса. Для frontend-части применяется библиотека ReactJS на языке JavaScript [9]. React позволяет создавать компонентную структуру, повторно использовать интерфейсные блоки и реализовывать интерактивные элементы. Выбор React обусловлен его популярностью, богатой экосистемой и высокой производительностью при работе с изменяемыми данными;
* реализация структуры и стиля интерфейса. Для вёрстки используется HTML — язык гипертекстовой разметки, задающий структуру контента, а также CSS (в частности, CSS-модули), позволяющие применять адаптивные стили, задавать цветовые схемы, шрифты и элементы дизайна. Такой выбор обеспечивает кроссбраузерность и гибкость в оформлении интерфейса [6];
* взаимодействие между клиентом и сервером. Для отправки HTTP-запросов и получения данных используется библиотека Axios [8]. Она проста в использовании и обеспечивает эффективную работу с API: получение списка товаров, регистрация, оформление заказа и т.д.;
* работа с базой данных. Для хранения данных о пользователях, товарах, заказах и избранном используется SQL Server, один из самых стабильных и производительных реляционных СУБД [3]. С ним работает ORM-фреймворк Entity Framework Core, упрощающий доступ к данным и сокращающий количество SQL-запросов за счёт модели сущностей [10]. Такой подход позволяет работать с БД на уровне C#-объектов;
* организация среды разработки. Для клиентской и сервисной части применяется Visual Studio Code — лёгкий, но функциональный редактор, который идеально подходит для React-проектов [11]. Оба инструмента поддерживают Git, расширения и плагины;
* тестирование API и отладка. Для проверки и документирования API-запросов используется Postman [7]. Это приложение позволяет вручную тестировать маршруты, аутентификацию и отправку данных, обеспечивая уверенность в корректной работе backend-сервиса.

Приложение COOLZY реализует полноценную сетевую архитектуру. Клиенты взаимодействуют с серверной частью через интернет с помощью HTTPS, получая актуальные данные из базы, отправляя запросы и обновляя информацию в реальном времени. Такая архитектура позволяет использовать приложение как на ПК, так и на мобильных устройствах, обеспечивая круглосуточный доступ к функционалу: корзине, оформлению заказа, профилю и избранному.

Выбор всех вышеуказанных технологий обусловлен их стабильностью, широким сообществом, большим количеством готовых решений, а также возможностью быстрой разработки, масштабирования и лёгкой поддержки проекта в будущем. Благодаря такому подходу обеспечивается высокая производительность, безопасность и современный пользовательский опыт веб-приложения COOLZY.

## Защита и сохранность данных

Веб-приложение «**OnlineGallery»** разработано с учётом современных требований к безопасности и конфиденциальности, что особенно важно при работе с персональными данными пользователей, историями покупок и платёжной информацией. Главная цель — обеспечить надёжную защиту информации от несанкционированного доступа и возможных угроз.

В системе реализована регистрация и аутентификация пользователей, основанная на уникальных логинах и зашифрованных паролях. Каждая пользовательская сессия привязана к конкретной записи в базе данных и проходит проверку авторизации, что исключает доступ посторонних лиц к личным данным и истории заказов.

Вся пользовательская информация (данные аккаунта, купленные изображения, избранное и комментарии) хранится в базе данных **SQL Server**, взаимодействие с которой осуществляется только через защищённые серверные API-запросы. Передача данных между клиентской и серверной частью осуществляется по защищённому протоколу **HTTPS**, исключающему возможность перехвата или подмены информации в процессе обмена.

Платформа запрашивает от пользователя только минимально необходимые разрешения:

**доступ к интернету** — требуется для загрузки изображений, регистрации, авторизации, оформления покупки и получения данных из базы;

**хранение токенов авторизации** — используется для безопасного управления сессией и упрощения повторного входа без необходимости постоянного ввода данных;

**доступ к внешним платёжным системам** — применяется при оформлении заказа с оплатой.

Серверная часть реализована на **ASP.NET Core**, с использованием встроенных средств авторизации и шифрования, включая защиту паролей (через хэширование) и проверку подлинности с помощью токенов. Это позволяет надёжно разграничивать уровни доступа и защищать чувствительные данные.

Дополнительно реализована чёткая структура ролей: пользователь и администратор имеют доступ только к разрешённым разделам, что предотвращает несанкционированное вмешательство в административные процессы. Все запросы обрабатываются через валидацию и фильтры доступа.

Архитектура «**OnlineGallery»** спроектирована в соответствии с принципами **«privacy by design»** и **«minimum access»**, что означает изначально встроенные механизмы защиты данных и ограничение доступа только к необходимым частям системы. Такой подход существенно снижает риски утечки информации и обеспечивает высокий уровень безопасности на всех уровнях взаимодействия.

## Организация и ведение информационной базы (модели)

Диаграмма «сущность-связь» представляет логическую модель базы данных веб-приложения по автоматизации работы сети магазинов техники «COOLZY» и отражает основные сущности, их атрибуты и связи между ними. Она используется на этапе проектирования системы для определения структуры хранения данных и логики их взаимодействия.

Сущность «Пользователь» содержит атрибуты UserId, ФИО, Email, номер телефона, пароль, дата регистрации, дата последнего входа, роль, а также используется для идентификации при входе в систему и оформления заказов. Пользователь может добавлять товары в корзину и в избранное, оформлять заказы и оставлять отзывы. Связь с другими таблицами позволяет отслеживать активность и историю заказов конкретного пользователя.

Сущность «Товар» включает атрибуты ProductId, название, бренд, цена, остаток, наличие, характеристики, описание, категория, RAM, память, процессор, а также ссылки на изображения. Это основная сущность, отображающая все товары, доступные в каталоге магазина.

Сущность «Категория» представлена атрибутами CategoryId и Name, и используется для группировки товаров (смартфоны, ноутбуки, телевизоры и др.). Каждый товар принадлежит одной категории, что упрощает навигацию и фильтрацию.

Сущность «Корзина» содержит CartId, UserId и общую стоимость. Каждый пользователь имеет одну корзину, в которую добавляются товары перед оформлением заказа. Сущность CartItems выступает связующей и хранит ProductId, Quantity, CartId.

Сущность «Избранное» (FavoriteItems) содержит UserId и ProductId, обеспечивая возможность добавлять понравившиеся товары в отдельный список для быстрого доступа.

Сущность «Заказ» хранит информацию об оформленных заказах: OrderId, UserId, дата, способ оплаты, ФИО, телефон, адрес. Сущность OrderItems связана с конкретными товарами в заказе и содержит поля ProductId, Quantity, Price.

Сущность «Администратор» представлена отдельной таблицей, связанной с Users, и отвечает за управление товарами, заказами, пользователями и аналитикой.

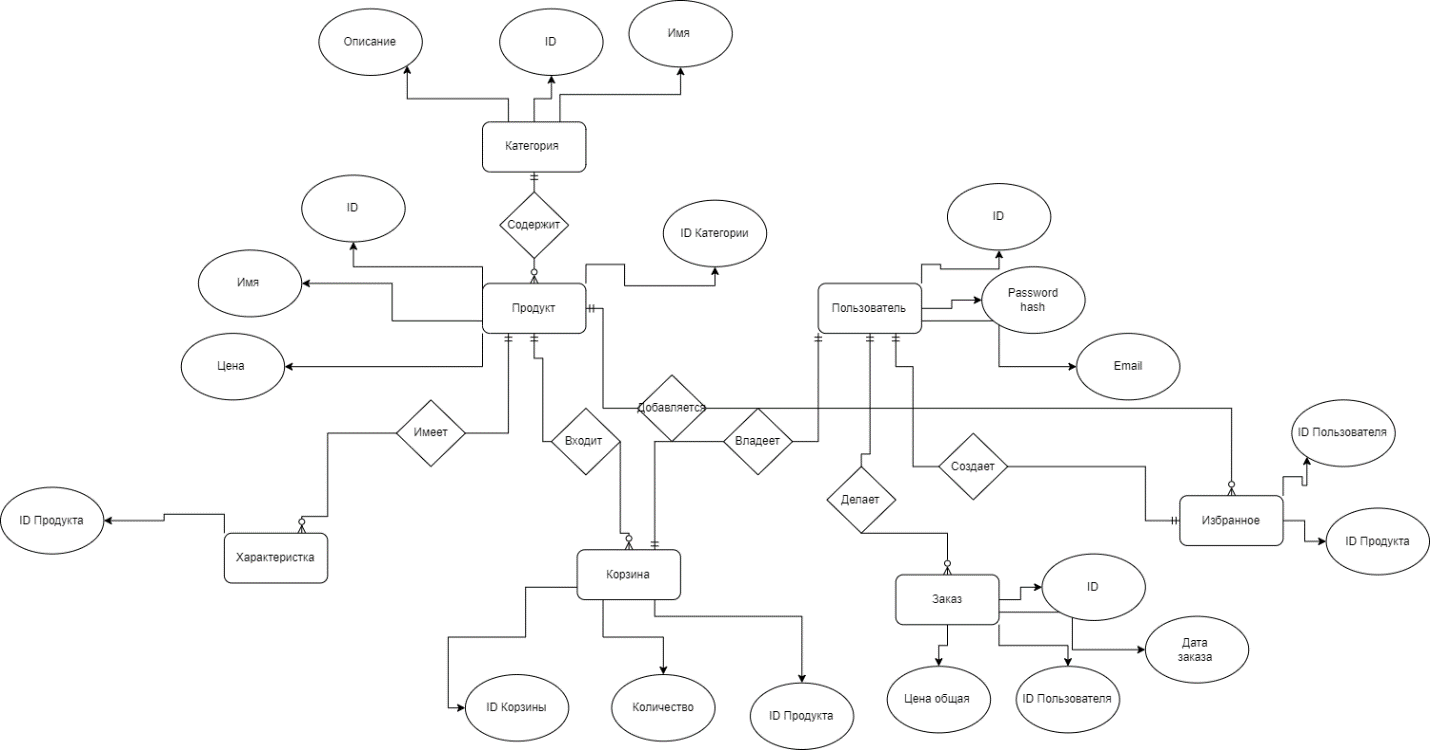
Сущность «Отзывы» содержит ProductId, UserId, оценку, комментарий и дату, обеспечивая возможность пользователям оставлять мнения о товарах.

Сущность «Изображения товаров» (ProductImages) связана с таблицей Products и позволяет каждому товару иметь одно или несколько изображений.

Сущность «Характеристики» (ProductSpecifications) содержит ProductId, группу, название атрибута и значение атрибута, что позволяет динамически описывать спецификации товара (например: экран, процессор, батарея и др.).

Связи между сущностями построены таким образом, чтобы обеспечить логическую целостность данных и правильную маршрутизацию между заказами, товарами, пользователями и администрированием. Один пользователь может иметь множество заказов и отзывов, каждый заказ состоит из нескольких товаров, каждый товар имеет изображения и характеристики, а все действия пользователя логируются в соответствующих таблицах.

Диаграмма «Сущность-связь» представлена на Рисунке 1.



1. – Диаграмма «Сущность-связь»

Приведённые таблицы базы данных были спроектированы на основании ER-диаграммы и отражают физическую реализацию логической структуры, разработанной в рамках предметной области веб-приложения «COOLZY» — системы автоматизации сети магазинов техники. Каждая таблица соответствует отдельной сущности, представленной на диаграмме, а связи между таблицами реализованы с помощью внешних ключей, обеспечивающих целостность и согласованность данных.

Такая структура позволяет эффективно реализовать все заявленные функции: управление пользователями и заказами, обработку корзины, отображение каталога, работу с избранным и отзывами, а также хранение характеристик и изображений товаров. Кроме того, база данных обеспечивает надёжную основу для расширения функциональности приложения в будущем.

Для реализации этих функций была разработана следующая структура таблиц:

Таблица «Пользователи» хранит информацию о зарегистрированных пользователях.

Таблица 1 – Структура таблицы «Users»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип поля | Размер | Описание поля |
| 1 | 2 | 3 | 4 |
| UserId | int | 4 | Идентификатор пользователя |
| Username | nvarchar | MAX | Имя пользователя |
| Email | nvarchar | MAX | Электронная почта (уникальная) |
| PasswordHash | nvarchar | MAX | Хэш пароля |
| Phone | nvarchar | MAX | Номер телефона |
| Gender | nvarchar | MAX | Пол |
| BirthDate | datetime2 | - | Дата рождения |
| RegisteredAt | datetime2 | - | Дата регистрации |
| LastLogin | datetime2 | - | Последний вход |
| IsAdmin | bit | - | Признак администратора |
| FirstName | nvarchar | MAX | Имя |
| LastName | nvarchar | MAX | Фамилия |

Таблица «Продукты» содержит данные о товарах, доступных для покупки.

Таблица 2 – Структура таблицы «Products»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип поля | Размер | Описание поля |
| ProductId | int | 4 | Идентификатор товара |
| Model | nvarchar | MAX | Модель |
| Brand | nvarchar | MAX | Производитель |
| OldPrice | decimal | 18,2 | Старая цена |
| NewPrice | decimal | 18,2 | Актуальная цена |
| Availability | nvarchar | MAX | Наличие |
| Ram | nvarchar | MAX | Объём оперативной памяти |
| Storage | nvarchar | MAX | Объём накопителя |
| CategoryId | int | 4 | Ссылка на категорию |
| Description | nvarchar | MAX | Описание товара |
| Quantity | int | 4 | Количество на складе |

Таблица «Категории» описывает виды техники, к которым относятся товары.

Таблица 3 – Структура таблицы «Categories»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип поля | Размер | Описание поля |
| CategoryId | int | 4 | Идентификатор категории |
| Name | nvarchar | MAX | Название категории |

Таблица «Изображения товаров» содержит ссылки на изображения товаров.

Таблица 4 – Структура таблицы «ProductImages»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип поля | Размер | Описание поля |
| ImageId | int | 4 | Идентификатор изображения |
| ProductId | int | 4 | Ссылка на товар |
| ImageUrl | nvarchar | MAX | Ссылка на изображение |

Таблица «Характеристики товаров» используется для отображения атрибутов.

Таблица 5 – Структура таблицы «ProductSpecifications»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип поля | Размер | Описание поля |
| Id | int | 4 | Идентификатор записи |
| ProductId | int | 4 | Ссылка на товар |
| GroupName | nvarchar | MAX | Группа (напр. «Экран») |
| AttributeName | nvarchar | MAX | Название характеристики |
| AttributeValue | nvarchar | MAX | Значение характеристики |

Таблица «Корзина» хранит информацию о текущей корзине пользователя.

Таблица 6 – Структура таблицы «Carts»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип поля | Размер | Описание поля |
| CartId | int | 4 | Идентификатор корзины |
| UserId | int | 4 | Ссылка на пользователя |
| TotalPrice | decimal | 18,2 | Общая сумма |

Таблица «Элементы корзины» содержит товары, добавленные в корзину.

Таблица 7 – Структура таблицы «CartItems»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип поля | Размер | Описание поля |
| Id | int | 4 | Идентификатор записи |
| ProductId | int | 4 | Ссылка на товар |
| Quantity | int | 4 | Количество |
| CartId | int | 4 | Ссылка на корзину |

Таблица «Избранное» фиксирует товары, добавленные в список избранного.

Таблица 8 – Структура таблицы «FavoriteItems»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип поля | Размер | Описание поля |
| Id | int | 4 | Идентификатор записи |
| UserId | int | 4 | Ссылка на пользователя |
| ProductId | int | 4 | Ссылка на товар |

Таблица «Заказы» содержит сведения об оформленных покупках.

Таблица 9 – Структура таблицы «Orders»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип поля | Размер | Описание поля |
| OrderId | int | 4 | Идентификатор заказа |
| UserId | int | 4 | Ссылка на пользователя |
| CreatedAt | datetime2 | - | Дата оформления заказа |
| Address | nvarchar | MAX | Адрес доставки |
| FullName | nvarchar | MAX | ФИО получателя |
| PaymentMethod | nvarchar | MAX | Способ оплаты |
| Phone | nvarchar | MAX | Телефон |

Таблица «Состав заказа» описывает позиции в каждом заказе.

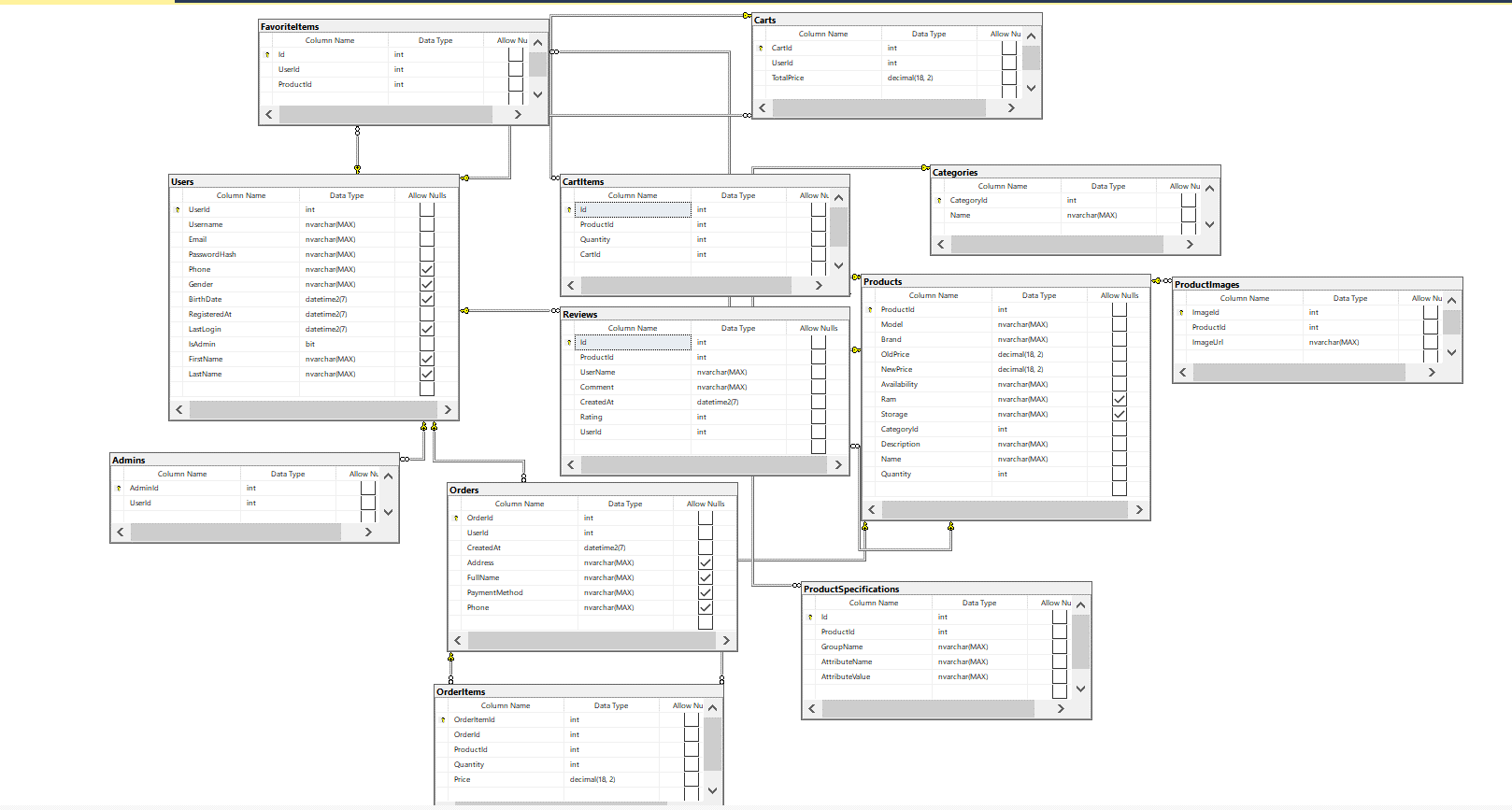
Таблица 10 – Структура таблицы «OrderItems»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип поля | Размер | Описание поля |
| 1 | 2 | 3 | 4 |
| OrderItemId | int | 4 | Идентификатор записи |
| OrderId | int | 4 | Ссылка на заказ |
| ProductId | int | 4 | Ссылка на товар |

Продолжение таблицы 10

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| Quantity | int | 4 | Количество |
| Price | decimal | 18,2 | Цена за единицу на момент заказа |

Диаграмма базы данных представлена на рисунке 2



1. – Диаграмма базы данных

# Реализация веб-приложения

## Описание разделов веб-приложения

Главная страница веб-приложения представляет собой удобный и интуитивно понятный интерфейс, предназначенный для комфортного взаимодействия пользователей с системой автоматизации работы маркетплейса компьютерной техники «COOLZY». В верхней части страницы расположено навигационное меню, включающее ссылки на основные разделы сайта: «Главная», «Каталог», «Корзина», «Сравнение», «Профиль» и «Контакты».

Центральная часть главной страницы отображает рекламные баннеры с акциями и новинками, а также каталог компьютерной и цифровой техники. Каждый товар представлен в виде карточки, содержащей изображение, наименование, цену и кнопку «Добавить в корзину» или «Избранное». Также предусмотрена возможность перейти на страницу с подробной информацией о товаре. Там отображаются характеристики, описание, отзывы пользователей и рейтинг. Для удобства реализована форма поиска, позволяющая быстро найти нужный товар по названию.

Слева от каталога размещён блок фильтрации, позволяющий отсортировать товары по категориям, брендам и ценовому диапазону. Также доступны опции сортировки по возрастанию/убыванию цены или по алфавиту. Это упрощает ориентацию в большом ассортименте товаров и помогает быстро найти нужную позицию.

Страница «Корзина» отображает добавленные пользователем товары, их количество, цену за единицу и общую стоимость заказа. Пользователь может изменить количество товаров, удалить позиции из корзины, а также перейти к оформлению заказа. В процессе оформления указываются контактные данные, способ оплаты и тип доставки (самовывоз или доставка на адрес). После подтверждения оформления появляется сообщение об успешном заказе, а также доступна возможность скачивания гарантийного талона в формате Word.

Раздел «Профиль» содержит личную информацию пользователя (имя, электронная почта, телефон, дата рождения), а также отображает историю заказов и список избранных товаров. Пользователь может отредактировать данные и выйти из системы. Интерфейс профиля адаптирован под мобильные и десктопные устройства.

Раздел «Сравнение» позволяет пользователю добавлять несколько товаров и визуально сравнивать их характеристики. В шапке сайта отображается счётчик товаров в сравнении, предусмотрена кнопка для очистки списка. Функция полезна при выборе между схожими моделями.

Раздел «Контакты» содержит подробную информацию о компании, включая название, номер телефона, адрес электронной почты и физический адрес. Также реализована интеграция с картой, позволяющая пользователю быстро определить местоположение офиса и построить маршрут.

Административная часть сайта доступна только после авторизации администратора через форму входа. В разделе «Управление товарами» можно добавлять новые позиции, редактировать существующие, изменять цену и количество, а также загружать изображения товаров. Также предусмотрена возможность экспорта данных о заказах и пользователях в формат Excel для последующего анализа. Раздел «Управление заказами» позволяет отслеживать все поступившие заявки и изменять их статус. Раздел «Управление пользователями» включает просмотр и редактирование данных зарегистрированных пользователей.

Редактирование информации о товарах осуществляется через отдельную страницу, где администратор может изменить такие данные, как название, описание, цена и количество товара, что обеспечивает удобное управление каталогом продукции.

## Разработка административной части приложения

Административная часть веб-приложения «COOLZY» представляет собой совокупность логических и физических компонентов, обеспечивающих централизованное управление товарами, заказами, пользователями и отчетностью.

Логическая структура включает несколько основных модулей: модуль авторизации, модуль управления товарами, модуль управления заказами, модуль управления пользователями, а также модуль формирования отчётов. Административная панель служит центральной точкой взаимодействия, предоставляя доступ ко всем функциям приложения.

Модуль авторизации обеспечивает доступ только для администраторов, выполняя контроль входа и разграничение прав. Модуль управления товарами позволяет добавлять, редактировать, удалять и просматривать информацию о товарах, включая изображения, цены и описание.

Модуль управления заказами реализует просмотр, обработку и изменение статуса поступивших заказов. Модуль управления пользователями предоставляет функциональность по просмотру зарегистрированных пользователей, их ролей и активности.

Модуль формирования отчётов используется для анализа ключевых показателей: количества заказов, активности пользователей и продаж за определённые периоды. Взаимодействие между модулями осуществляется через общую базу данных, что обеспечивает целостность бизнес-логики и актуальность информации.

Логическая структура административной части представлена на приложение Б рисунок Б.1.

Физическая структура административной части веб-приложения «COOLZY» включает клиентскую, серверную и базовую компоненты. Клиентская часть реализована в виде веб-интерфейса, доступного через браузер администратора, где осуществляется взаимодействие с системой.

Серверная часть обеспечивает обработку запросов от клиента, реализацию логики управления товарами, заказами, пользователями и отчётностью. Взаимодействие между компонентами происходит с использованием REST API и архитектуры клиент-сервер.

Базовая часть представлена системой управления базами данных, в которой хранится структурированная информация о пользователях, товарах, заказах, категориях и другой бизнес-логике административной панели.

При необходимости возможна адаптация архитектуры, включая масштабирование системы за счёт разделения клиентской и серверной части, внедрения облачных решений и выделения отдельных сервисов.

Таким образом, физическая структура административной части в совокупности с логической формирует надёжную, масштабируемую и удобную платформу для администрирования сервиса интернет-магазина «COOLZY».

Логическая структура административной части представлена на приложение Б рисунок Б.2.

## Разработка клиентской части приложения

Клиентская часть веб-приложения «COOLZY» предназначена для обеспечения удобного взаимодействия пользователей с системой. Логическая структура клиентской части включает несколько взаимосвязанных компонентов, каждый из которых выполняет определённые функции.

Логическая структура веб-приложения состоит из следующих ключевых компонентов. Регистрация и вход обеспечивают доступ пользователей к личному кабинету и сохраняют индивидуальные данные. Редактирование профиля позволяет изменять личную информацию и просматривать историю заказов.

Поиск и фильтрация используются для удобной навигации по каталогу товаров и позволяют быстро находить нужные позиции по категориям, брендам или ключевым словам. Просмотр каталога предоставляет пользователю доступ ко всем доступным товарам с возможностью перехода на подробную карточку товара.

Добавление в корзину позволяет сформировать список покупок, а оформление заказа — завершить покупку, выбрав способ доставки и оплаты. Компонент «Избранное» даёт возможность сохранять понравившиеся товары для быстрого доступа в будущем. Также реализован просмотр заказов с детализацией по каждому из них.

Навигационные элементы (меню, кнопки и маршруты) обеспечивают удобное перемещение по разделам.

Логическая структура клиентской части представлена на приложение Б на рисунке Б.3

Физическая структура клиентской части веб-приложения «COOLZY» представляет собой описание размещения физических компонентов, организованных по поддиректориям в рамках основной директории проекта. Эти компоненты обеспечивают отображение интерфейса, маршрутизацию, обработку пользовательских действий и взаимодействие с серверной частью системы.

Структура включает разделы, отвечающие за навигацию между страницами, подключение к базе данных, отображение данных, а также взаимодействие с моделью приложения. Каждый раздел содержит компоненты, необходимые для реализации отдельных функций клиентской части, таких как просмотр каталога, работа с корзиной, оформление заказов и управление профилем пользователя.

Физическая структура клиентской части представлена на приложение Б на рисунке Б.4.

## Описание используемых функций и процедур

Для реализации графического интерфейса веб-приложения «COOLZY» использовались технологии C#, ASP.NET Core MVC, React и SQL Server. Такой стек обеспечивает современный внешний вид, кроссбраузерную совместимость и надёжное хранение данных. Архитектура построена по принципу клиент–серверного взаимодействия с использованием REST API, что позволяет эффективно масштабировать проект и разделять логику между фронтендом и бэкендом.

Функция в программировании — это отдельный блок кода, выполняющий конкретную задачу и вызываемый из разных частей программы. Использование функций помогает структурировать код, делает его более понятным, удобным для повторного использования и сопровождения. Функции могут принимать входные параметры и возвращать результат своей работы.

Кроссплатформенность и адаптивность обеспечиваются за счёт использования React: все компоненты интерфейса автоматически адаптируются под размер экрана устройства. Для стилизации применяются кастомные CSS-решения и адаптивная вёрстка, что делает сайт удобным для использования как на ПК, так и на мобильных устройствах.

Локальная база данных реализована с использованием SQL Server и ORM-библиотеки Entity Framework Core, что обеспечивает надёжную и эффективную работу с таблицами пользователей, товаров, заказов, корзины, изображений, категорий и отзывов.

Таким образом, веб-приложение «COOLZY» представляет собой полноценный интернет-магазин, предлагающий пользователю удобную навигацию, авторизацию, возможность оформления заказов, ведения истории покупок и управления профилем.

Код функции регистрации пользователя представлен ниже:

[HttpPost]

public async Task<IActionResult> Register(User user)

{

if (\_context.Users.Any(u => u.Email == user.Email))

return BadRequest("Email уже используется");

\_context.Users.Add(user);

await \_context.SaveChangesAsync();

return Ok(user);

}

Код функции авторизации пользователя:

[HttpPost]

public async Task<IActionResult> Login(LoginModel model)

{

var user = await \_context.Users

.FirstOrDefaultAsync(u => u.Email == model.Email && u.Password == model.Password);

if (user == null)

return Unauthorized();

HttpContext.Session.SetInt32("UserId", user.Id);

return Ok(user);

}

Код функции добавления товара в корзину:

[HttpPost]

public async Task<IActionResult> AddToCart(CartItem item)

{

var existing = await \_context.CartItems

.FirstOrDefaultAsync(ci => ci.UserId == item.UserId && ci.ProductId == item.ProductId);

if (existing != null)

existing.Quantity += item.Quantity;

else

\_context.CartItems.Add(item);

await \_context.SaveChangesAsync();

return Ok();

}

Код функции оформления заказа:

[HttpPost]

public async Task<IActionResult> CreateOrder(Order order)

{

if (order.Items == null || !order.Items.Any())

return BadRequest("Корзина пуста");

\_context.Orders.Add(order);

await \_context.SaveChangesAsync();

return Ok(order.Id);

}

Код функции добавления отзыва на товар:

[HttpPost]

public async Task<IActionResult> AddReview(Review review)

{

review.CreatedAt = DateTime.UtcNow;

\_context.Reviews.Add(review);

await \_context.SaveChangesAsync();

return Ok();

}

Код функции добавления товара в избранное:

[HttpPost]

public async Task<IActionResult> AddToFavorites(FavoriteItem fav)

{

if (!\_context.FavoriteItems.Any(f => f.UserId == fav.UserId && f.ProductId == fav.ProductId))

{

\_context.FavoriteItems.Add(fav);

await \_context.SaveChangesAsync();

}

return Ok();

}

Код функции удаления товара из избранного:

[HttpDelete("{userId}/{productId}")]

public async Task<IActionResult> RemoveFromFavorites(int userId, int productId)

{

var item = await \_context.FavoriteItems

.FirstOrDefaultAsync(f => f.UserId == userId && f.ProductId == productId);

if (item != null)

{

\_context.FavoriteItems.Remove(item);

await \_context.SaveChangesAsync();

}

return Ok();

}

Код функции редактирования профиля пользователя:

[HttpPut("{id}")]

public async Task<IActionResult> UpdateProfile(int id, User updatedUser)

{

var user = await \_context.Users.FindAsync(id);

if (user == null) return NotFound();

user.FirstName = updatedUser.FirstName;

user.LastName = updatedUser.LastName;

user.Phone = updatedUser.Phone;

user.BirthDate = updatedUser.BirthDate;

await \_context.SaveChangesAsync();

return Ok(user);

}

## Функциональное тестирование

Функциональное тестирование представляет собой процесс проверки программного обеспечения на соответствие заявленным требованиям, при котором оценивается корректность выполнения системой всех предусмотренных функций. Основная задача данного вида тестирования — удостовериться, что программный продукт надёжно выполняет заданные действия как при корректных, так и при ошибочных вводах.

Проверка будет осуществляться на основе заранее подготовленных тест-кейсов, каждый из которых описывает определённый сценарий использования, включает входные данные, ожидаемый результат и фактический результат выполнения. Тест-кейсы охватывают основные модули приложения: регистрацию, вход в систему, просмотр каталога рецептов, добавление рецептов в избранное, автоматический перерасчёт ингредиентов, формирование списка покупок и добавление пользовательских рецептов. Отдельное внимание будет уделено обработке некорректных данных, таких как пустые поля, неправильный формат электронной почты, неверные учетные данные и т. д.

Результаты функционального тестирования будут документированы и дополнены скриншотами пользовательского интерфейса, подтверждающими корректность работы функций или наличие выявленных ошибок.

Таблица 11 – Тест - кейсы

|  |  |  |  |
| --- | --- | --- | --- |
| Модуль / Функция | Шаги выполнения | Ожидаемый результат | Фактический результат |
| 1 | 2 | 3 | 4 |
| Регистрация | 1. Открыть сайт 2. Нажать иконку «Личного кабинета» 3. Нажать кнопку «Нет аккаунта? Создайте его» 4. Заполнить данные | Переход на форму «Личный кабинет» | Успешный переход на страницу «Личный кабинет».  Результат представлен на приложении Д на рисунке Д.1 |

Продолжение таблицы

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| Авторизация | 1. Нажать иконку «Личного кабинета» 2. Нажать кнопку «Нет аккаунта? Создайте его» 3. Заполнить данные | Переход на форму «Личный кабинет» | Успешный переход на страницу «Личный кабинет».  Результат представлен на приложении Д на рисунке Д.2 |
| Добавление товара в избранные | 1. Выбрать товар 2. Нажать на иконку «Сердечко (добавление в избранное)» | Добавление товара в избранные | Успешное добавление товара в избранные.  Результат представлен на приложении Д на рисунке Д.3 |
| Добавление товара в сравнение | 1. Выбрать товар 2. Нажать на кнопку «Сравнить» | Добавление товара в сравнение | Успешное добавление товара в сравнение.  Результат представлен на приложении Д на рисунке Д.4 |
| Поиск | 1. На странице «Главная» нажать на поле «Поиск» 2. Заполнить поле данные 3. Нажать кнопку «Поиск» | Показ результата поиска | Успешный поиск и вывод.  Результат представлен на приложении Д на рисунке Д.5 |
| Добавление товара в корзину | 1. Выбрать товар 2. Нажать кнопку «В корзину» | Добавление товара в корзину | Успешное добавление товара в корзину  Результат представлен на приложении Д на рисунке Д.6 |
| Оформление заказа | 1. Перейти в корзину 2. Нажать на кнопку оформление заказа 3. Заполнить данные 4. Нажать на кнопку «Подтвердить заказ» | Успешное оформление заказа. | Успешное оформление заказа  Результат представлен на приложении Д на рисунке Д.7 |

# Применение

## Назначение веб-приложения

Данное веб-приложение разработано для автоматизации процессов в сети магазинов бытовой техники «COOLZY». Оно предоставляет удобные инструменты для управления ассортиментом товаров, обработки заказов, ведения корзины, а также регистрации и обслуживания пользователей. Благодаря системе упрощается работа персонала и улучшается взаимодействие с клиентами.

Для установки приложения необходимо следовать инструкциям разработчика. После размещения на сервере требуется выполнить настройку окружения: установить необходимые библиотеки, подключить базу данных SQL Server и выполнить начальную миграцию.

После завершения установки приложение можно запустить с помощью соответствующих команд для проверки его корректной работы и готовности к использованию.

Веб-приложение ориентировано на автоматизацию торговли и повышения качества обслуживания в сети магазинов «COOLZY». Оно позволяет клиентам просматривать каталог техники, добавлять товары в корзину, оформлять заказы, а также управлять личным кабинетом. Администраторы получают доступ к отдельной панели для управления товарами, категориями, заказами и пользователями.

Ограничения области применения включают:

* доступ к оформлению заказов и управлению данными доступен только авторизованным пользователям;
* административный функционал доступен исключительно для сотрудников с соответствующими правами доступа;
* система специализируется на продаже бытовой техники и ориентирована на магазины, работающие по модели электронной коммерции.

## Программно-аппаратное обеспечение сервера и клиента

Разрабатываемое веб-приложение для маркетплейса «COOLZY» функционирует на сервере под управлением операционной системы Windows 10.

Минимальные системные требования для сервера:

* процессор с четырьмя и более физическими ядрами;
* оперативная память объёмом не менее 4 ГБ;
* свободное место на жёстком диске от 2 ГБ.

Для запуска и корректной работы программного обеспечения необходимо установить следующие компоненты:

* веб-сервер Apache HTTP Server — сервер с открытым исходным кодом, обрабатывающий HTTP-запросы от клиентов и предоставляющий доступ к статическому и динамическому контенту (HTML, CSS, JavaScript). Используется для хостинга клиентской части и API;
* система управления базами данных Microsoft SQL Server — реляционная СУБД, применяемая для хранения информации о товарах, пользователях, заказах, отзывах и других элементах бизнес-логики маркетплейса;
* платформа разработки .NET SDK (ASP.NET Core) — набор средств разработки для реализации серверной логики приложения на языке C# с архитектурой MVC. Поддерживает построение REST API и безопасную работу с базой данных;
* среда выполнения JavaScript Node.js — используется для запуска JavaScript-кода на сервере, применяясь в задачах интеграции с внешними сервисами, сборки фронтенда и отправки уведомлений (по необходимости).

Клиентская часть веб-приложения создаётся с использованием следующих инструментов:

* javaScript-библиотека React — используется для создания интерактивного интерфейса пользователя, включая каталог товаров, корзину, оформление заказов, профиль и админ-панель;
* ведактор кода Visual Studio Code — основной инструмент для разработки клиентской части, поддерживающий HTML, CSS, JavaScript и React, а также интеграцию с системами контроля версий;
* веб-браузер (Google Chrome, Firefox, Microsoft Edge и др.) — необходим для запуска клиентской части, предоставляющей доступ к веб-интерфейсу приложения.

Развёртывание и запуск приложения:

Развёртывание базы данных осуществляется с помощью SQL Server Management Studio (SSMS), где создаётся база данных CoolzyDB со всеми необходимыми таблицами.  
Серверная часть запускается через Visual Studio, а клиентская — при помощи команды npm start в командной строке Node.js.

После запуска веб-приложение становится доступным локально: API функционирует на http://localhost:5085, клиентская часть — на http://localhost:3000. Клиент взаимодействует с сервером через HTTP-запросы, обрабатываемые ASP.NET Core и SQL Server.

# Заключение

В рамках выполнения проекта на тему «Разработка веб-приложения интернет-магазина COOLZY» было создано полнофункциональное веб-приложение, предназначенное для покупки техники и аксессуаров. Приложение предоставляет пользователям возможность просматривать каталог товаров, фильтровать и искать позиции по категориям, оформлять заказы, регистрироваться в системе, а также управлять корзиной и личным кабинетом.

Для достижения целей проектирования были выполнены следующие задачи:

* разработка веб-приложения для удобного взаимодействия с пользователями и товарными позициями;
* проектирование пользовательских сценариев и ролей (покупатель, администратор);
* создание интерфейсов в React с адаптивной вёрсткой для поддержки мобильных и десктопных устройств;
* реализация серверной части с использованием ASP.NET Core и архитектуры REST API;
* подключение базы данных SQL Server и построение схемы данных с использованием Entity Framework Core;
* описание структуры приложения через диаграммы: прецедентов, классов и развёртывания;
* разработка административной панели для управления товарами, заказами и пользователями;
* проведение тестирования основных функций: регистрация, поиск, оформление заказа, управление каталогом.

Приложение обладает удобным, современным и адаптивным интерфейсом. Пользователь может быстро зарегистрироваться, подобрать нужный товар с помощью фильтрации, добавить его в корзину и оформить заказ, выбрав подходящий способ доставки и оплаты. Благодаря использованию компонентов React интерфейс динамично реагирует на действия пользователя и сохраняет целостность на всех типах устройств.

Ключевыми преимуществами проекта являются модульная архитектура, масштабируемость, безопасность и простота обслуживания. В ходе разработки были закреплены практические навыки по построению клиент-серверных приложений, проектированию базы данных, реализации бизнес-логики и созданию адаптивного фронтенда.

Приложение COOLZY готово к использованию в реальной среде и может применяться в качестве платформы для электронной коммерции в сфере продажи техники и аксессуаров.

# Список информационных источников

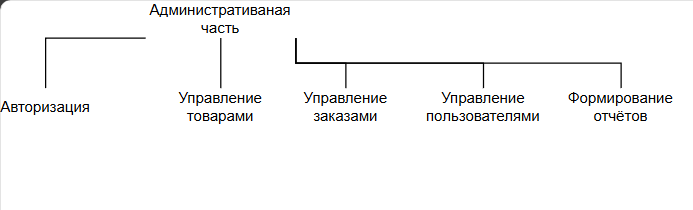
1. Багласова, Т.Г. Методические указания по выполнению курсового проекта для учащихся по специальности 2-40 01 01 «Программное обеспечение технологий» / Т.Г. Багласова. – Минск : КБП, 2017. – 30 c.
2. Багласова, Т.Г. Методические указания по оформлению курсовых и дипломных проектов / Т.Г. Багласова, К.О. Якимович. – Минск : КБП, 2013. – 29 c.
3. Бондарь, А.Г. Microsoft SQL Server 2012 / А.Г. Бондарь. – СПб. : БХВ-Петербург, 2013. – 608 с.
4. Тепляков, С. Паттерны проектирования на платформе .NET / С.Тепляков. – СПб. : Питер, 2015. – 320 с
5. Руководство по программированию на C# [Электронный ресурс]. – Microsoft, 2020. – Режим доступа : http://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/. – Дата доступа : 25.04.2020.
   * HTML и CSS. Полный справочник [Электронный ресурс] / J. Duckett. — Режим доступа: <https://developer.mozilla.org> — Дата доступа: 13.06.2025.
   * Постман. Документация и загрузка [Электронный ресурс]. – Режим доступа: <https://www.postman.com> – Дата доступа: 13.05.2025.
   * Axios: HTTP-библиотека для JavaScript [Электронный ресурс]. – Режим доступа: <https://axios-http.com> – Дата доступа: 13.05.2025.
   * React – библиотека JavaScript [Электронный ресурс]. – Режим доступа: <https://react.dev> – Дата доступа: 13.05.2025.
   * Entity Framework Core: официальная документация [Электронный ресурс]. – Microsoft, 2024. – Режим доступа: <https://learn.microsoft.com/ef/core> – Дата доступа: 13.05.2025.
   * Visual Studio Code – редактор исходного кода [Электронный ресурс]. – Режим доступа: <https://code.visualstudio.com> – Дата доступа: 13.05.2025.

Приложения А  
Ментальная карта

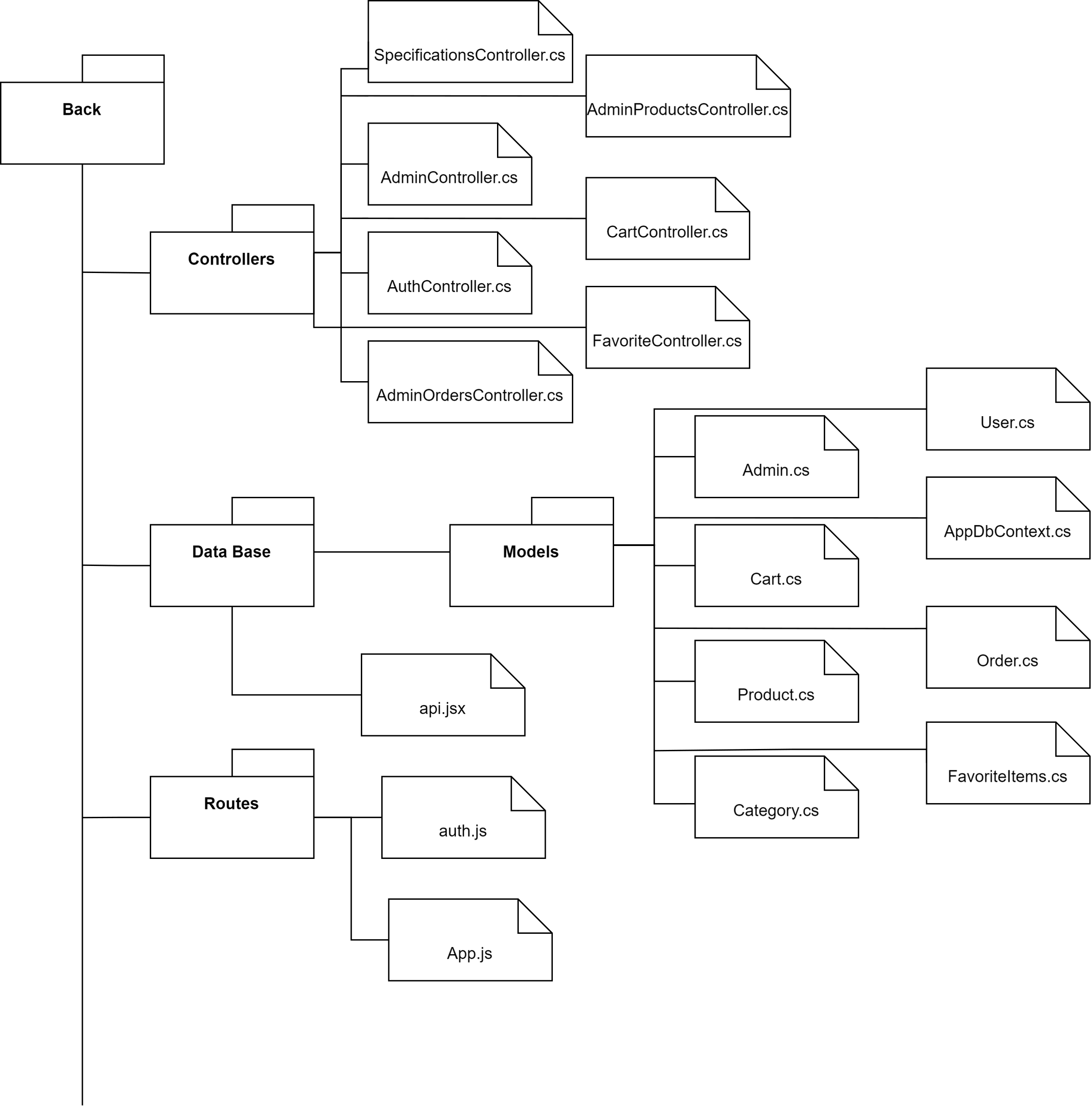


Рисунок А.1 – Ментальная карта

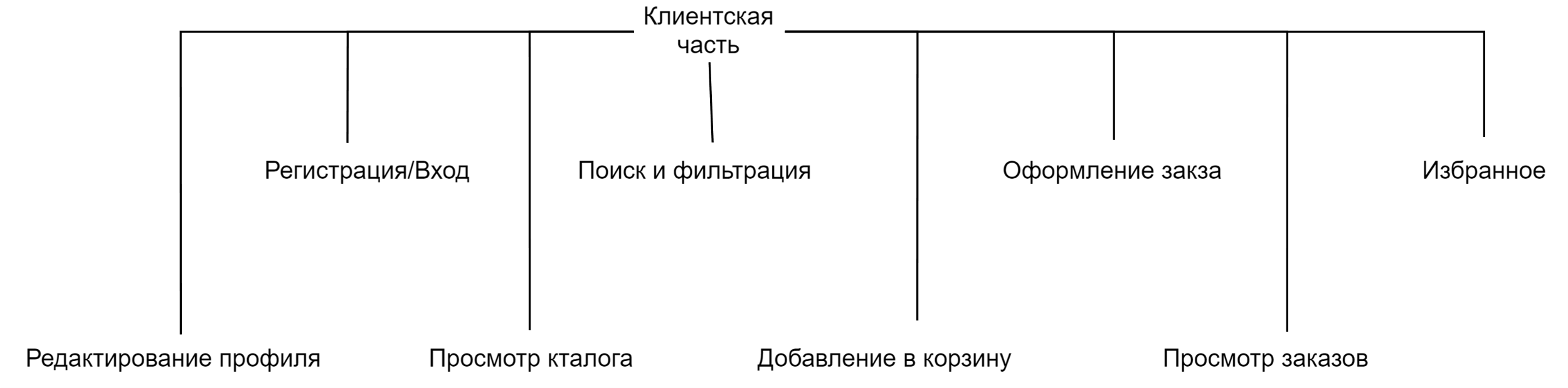
Приложение Б  
Логическая и физическая структура



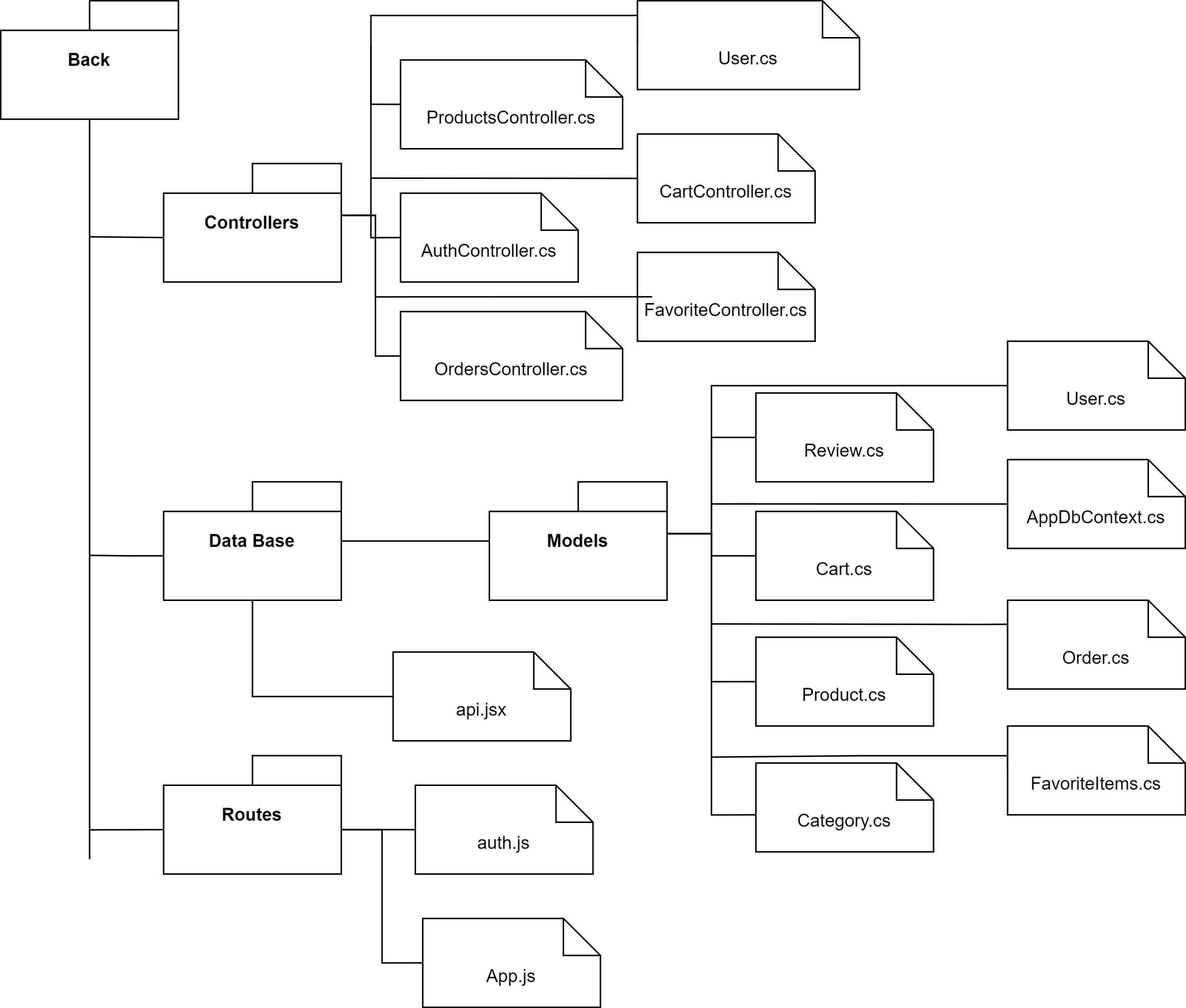
1. Логическая структура административной части веб -приложения



1. Физическая структура административной части веб – приложения

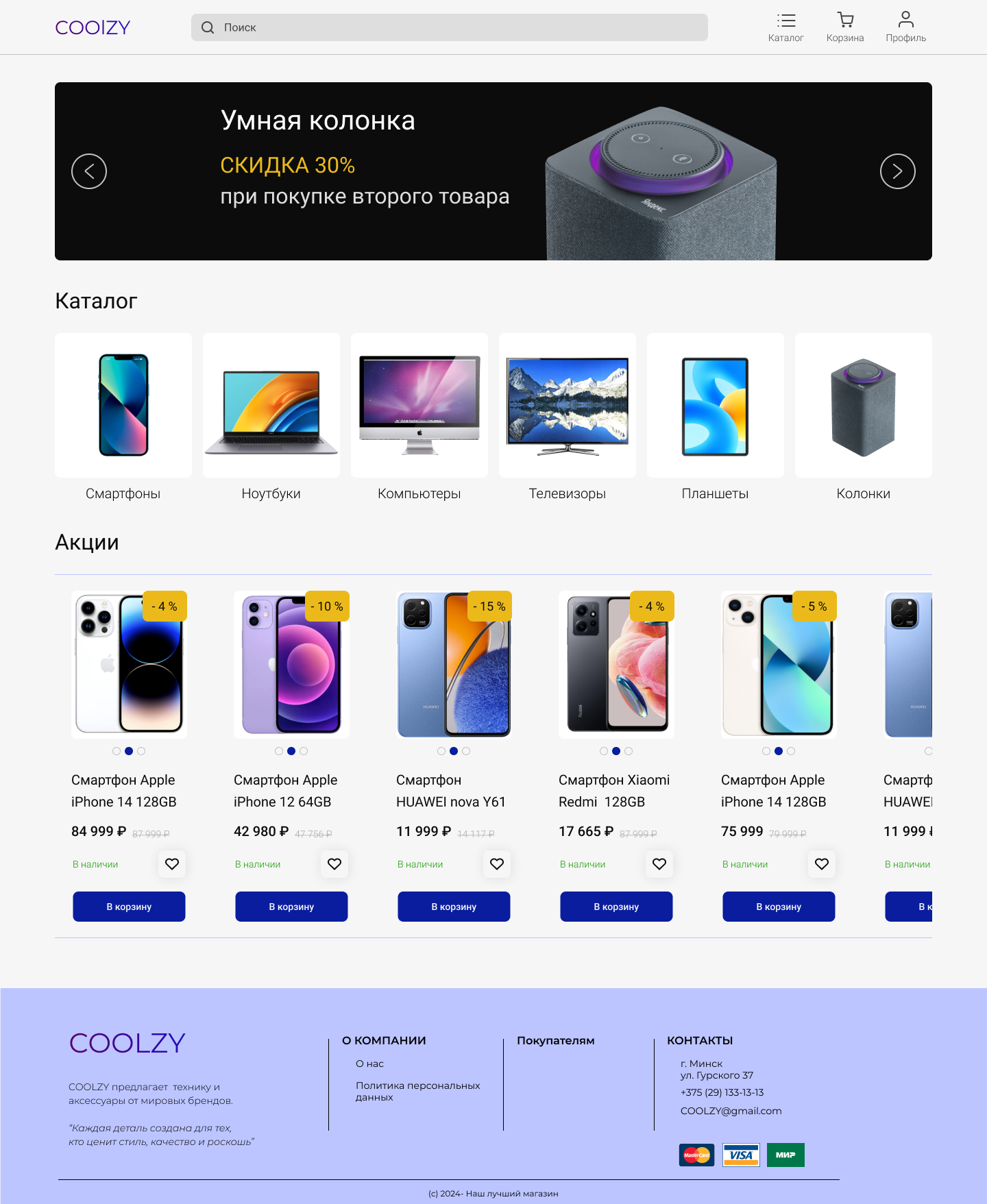


1. Логическая структура клиентской части веб – приложения

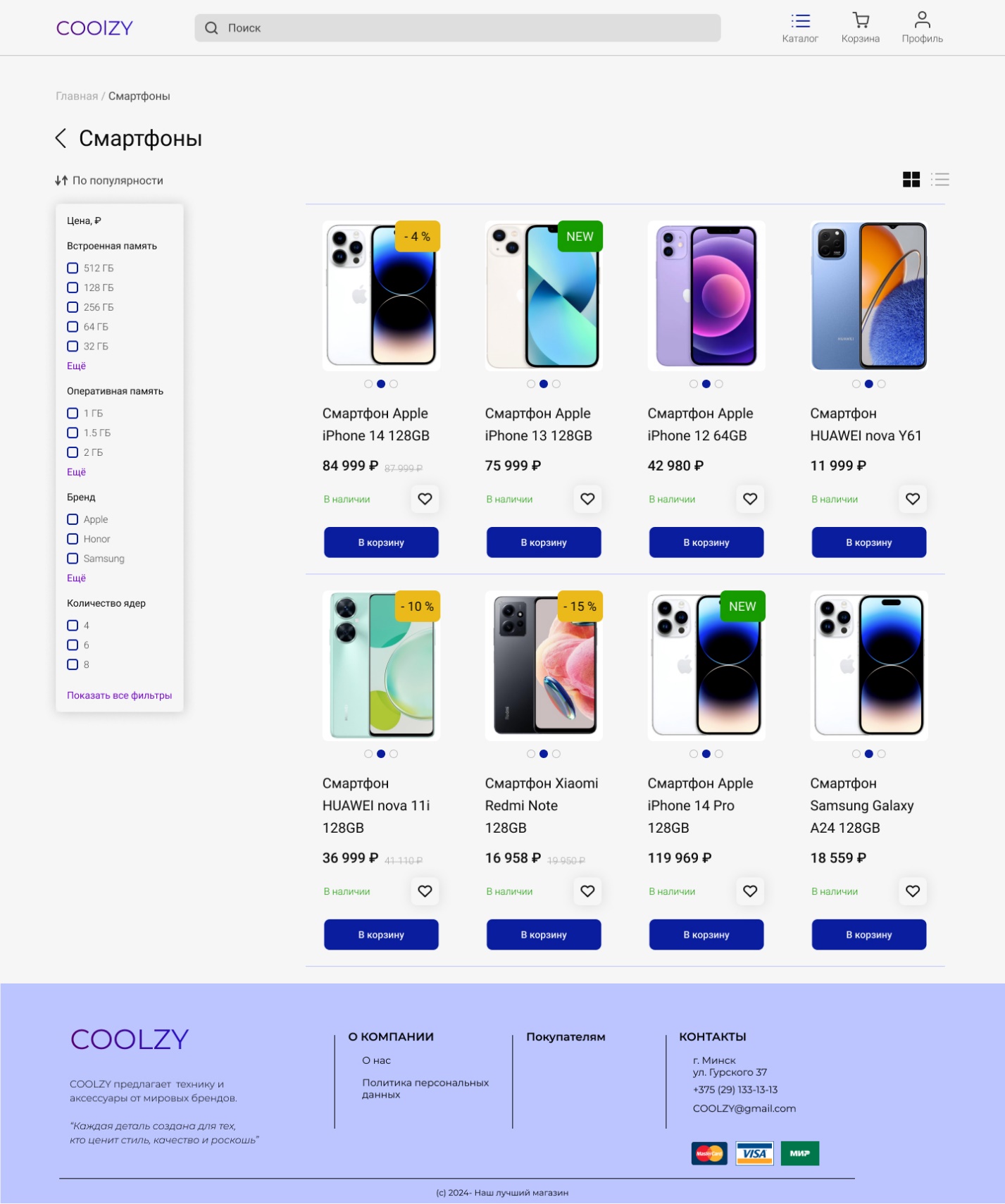


1. Физическая структура клиентской части веб – приложения

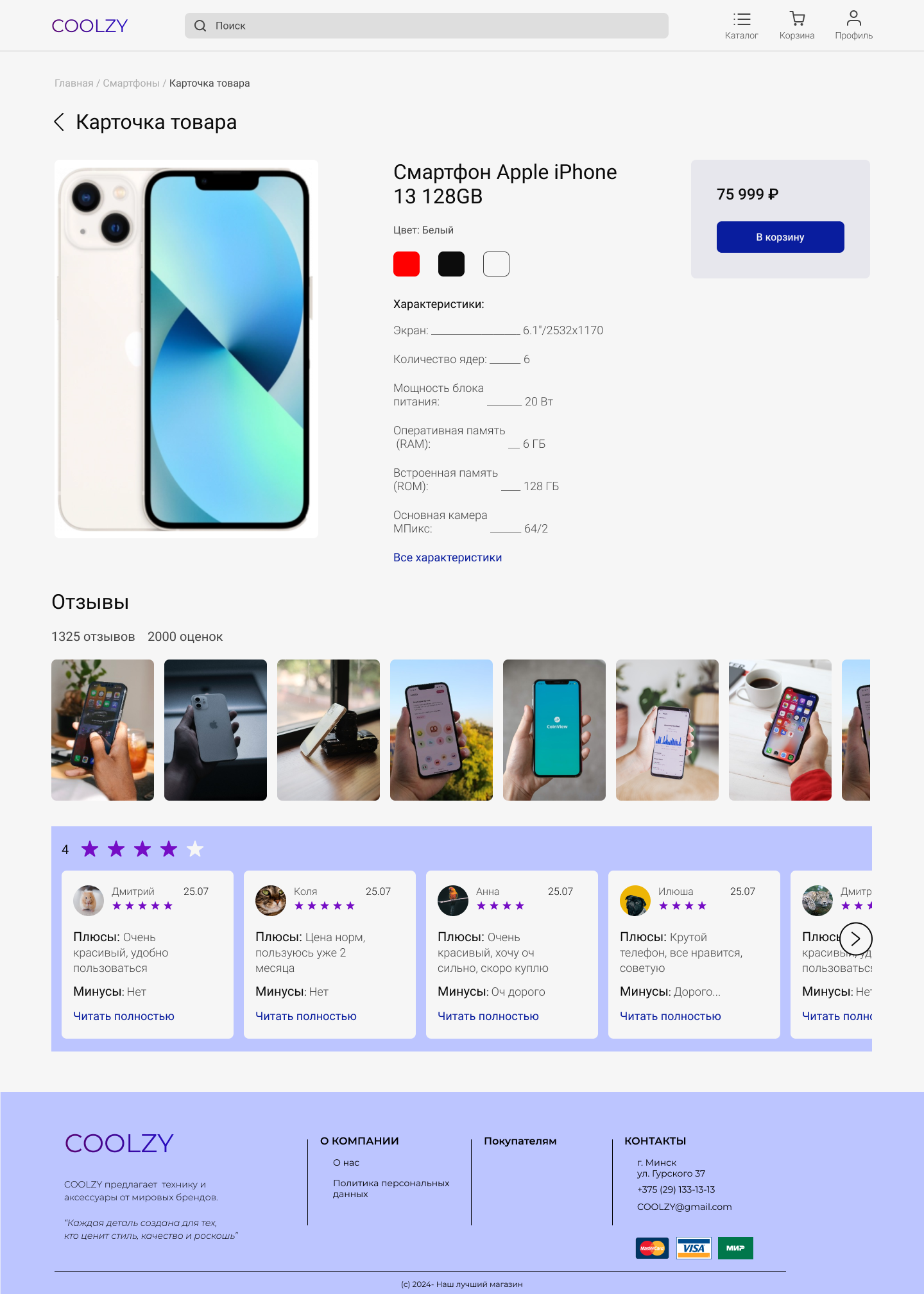
Приложение В  
Макеты страниц веб – приложения



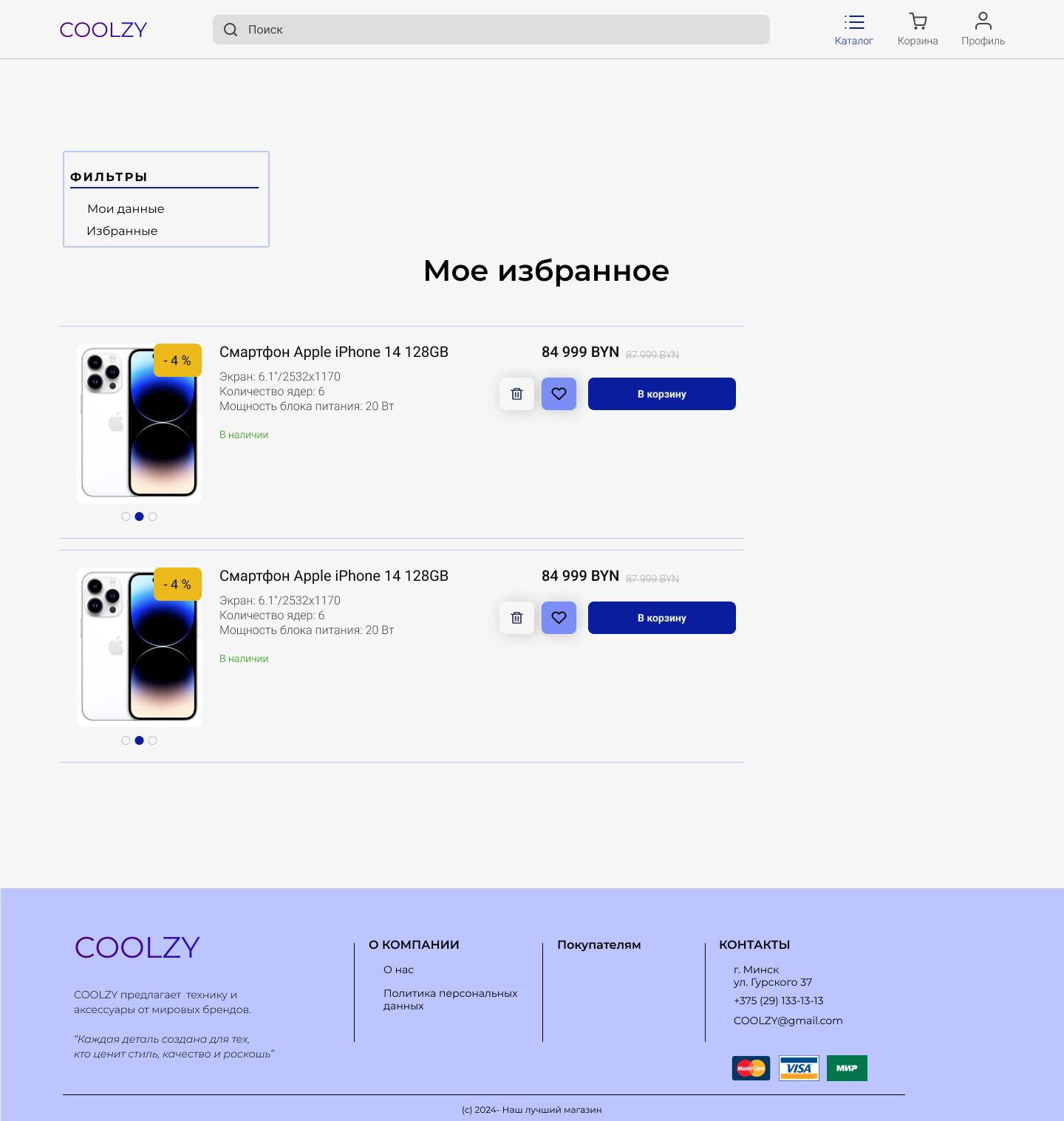
1. Макет страницы «Главная



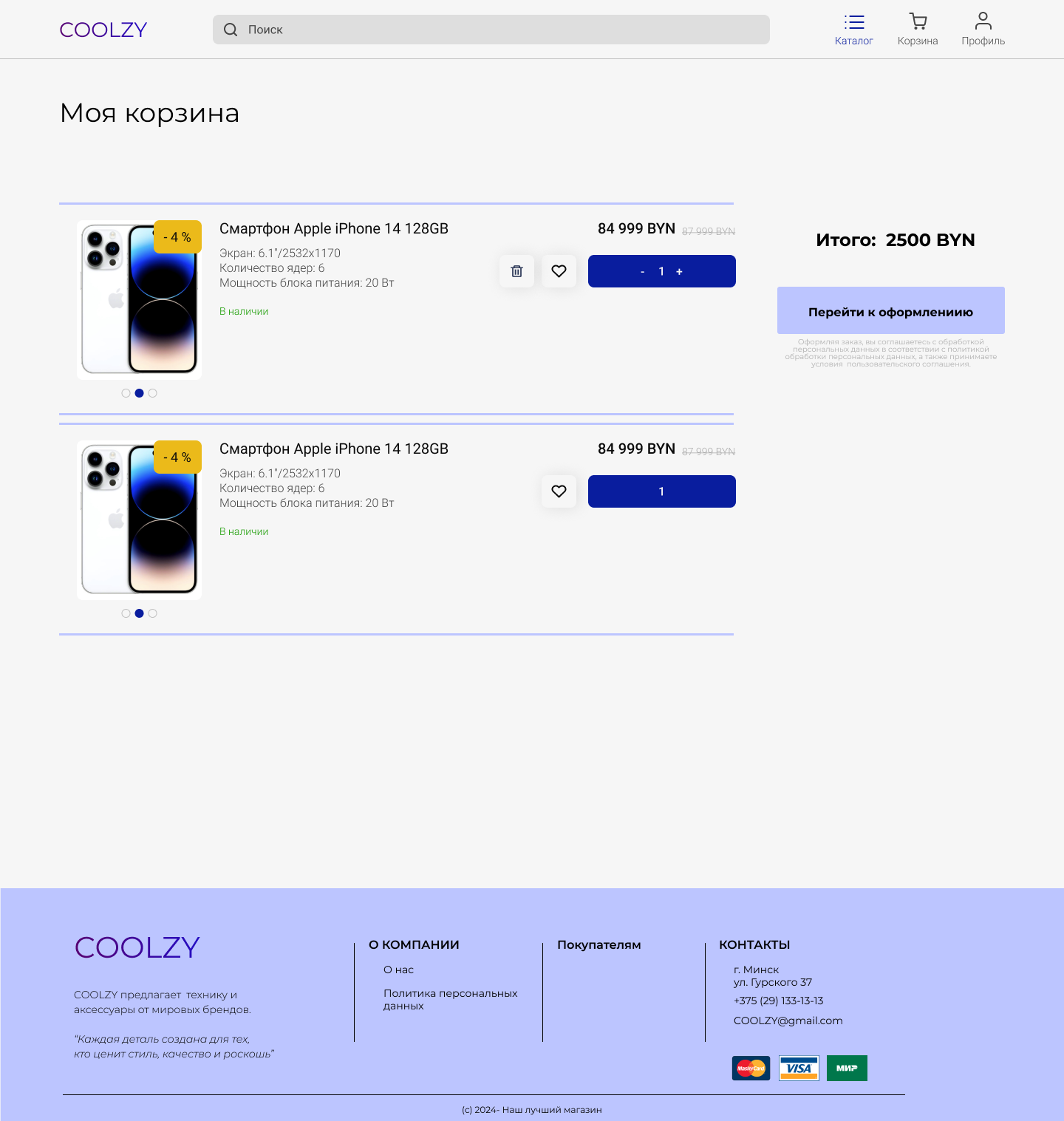
1. Макет страницы «Каталог товаров»



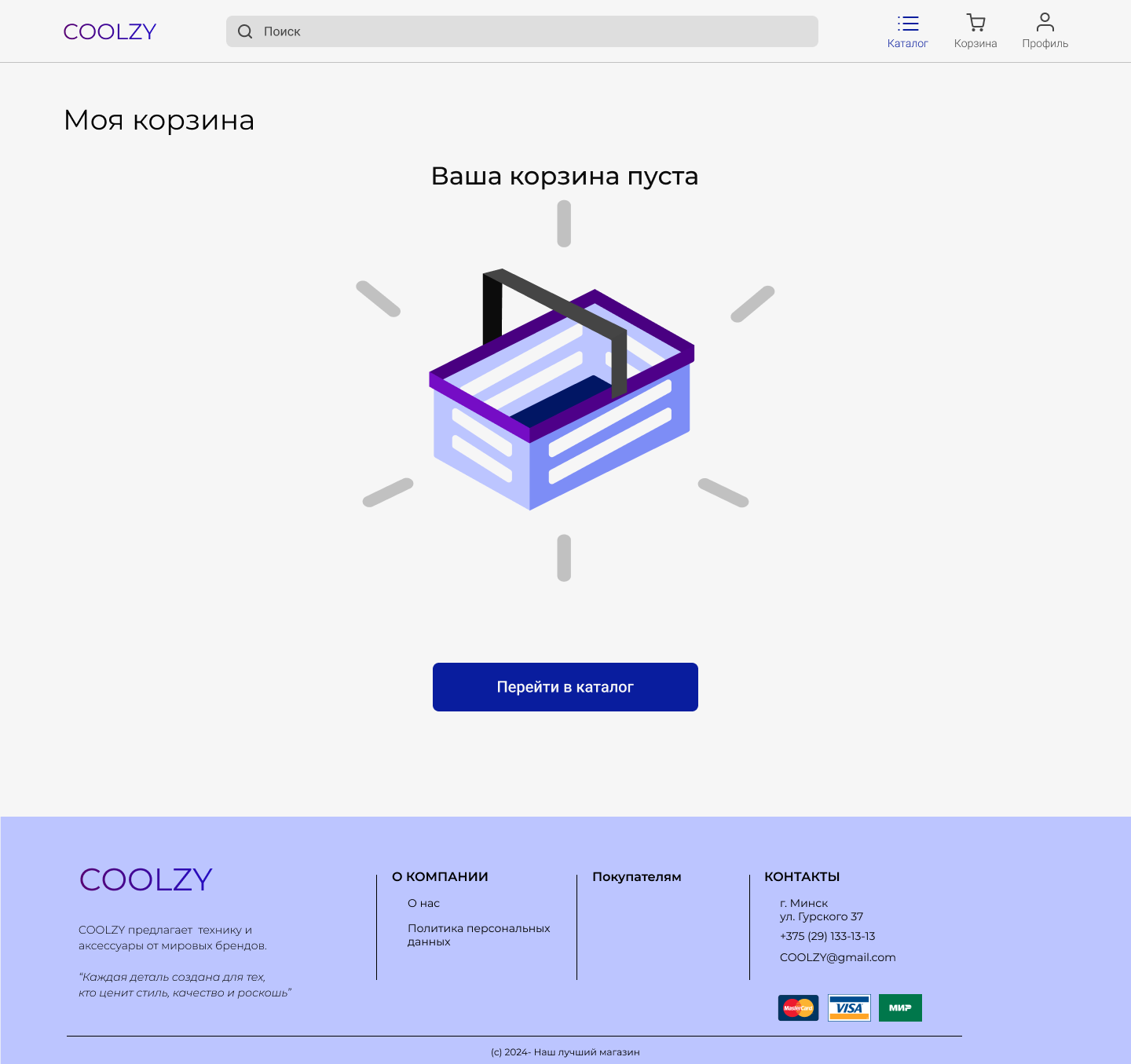
1. Макет страницы «Карточка товара»



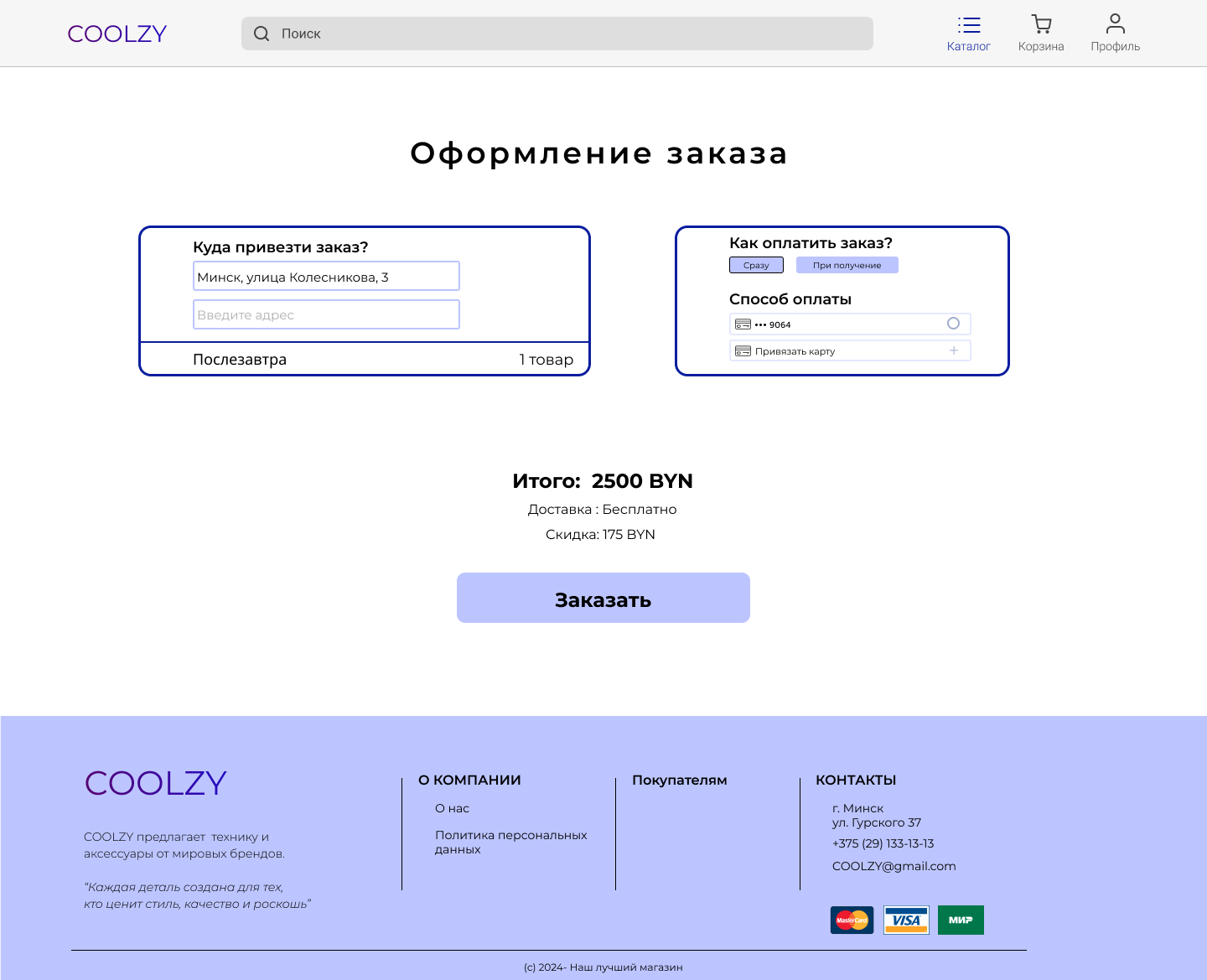
1. Макет страницы «Избранное»



1. Макет страницы «Корзина(с товарами)»



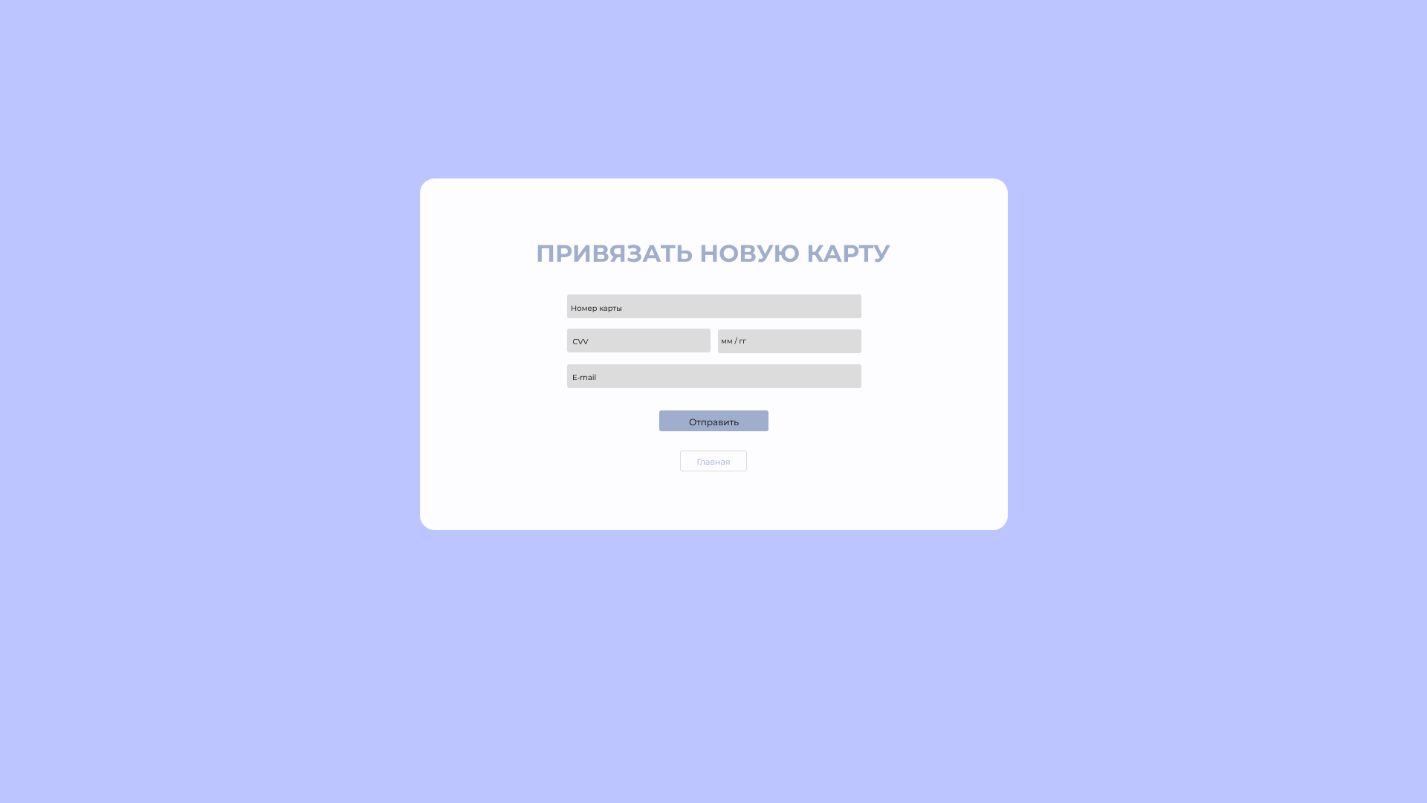
1. Макет страницы «Корзина (пустая)



1. Макет страницы «Оформления заказа»



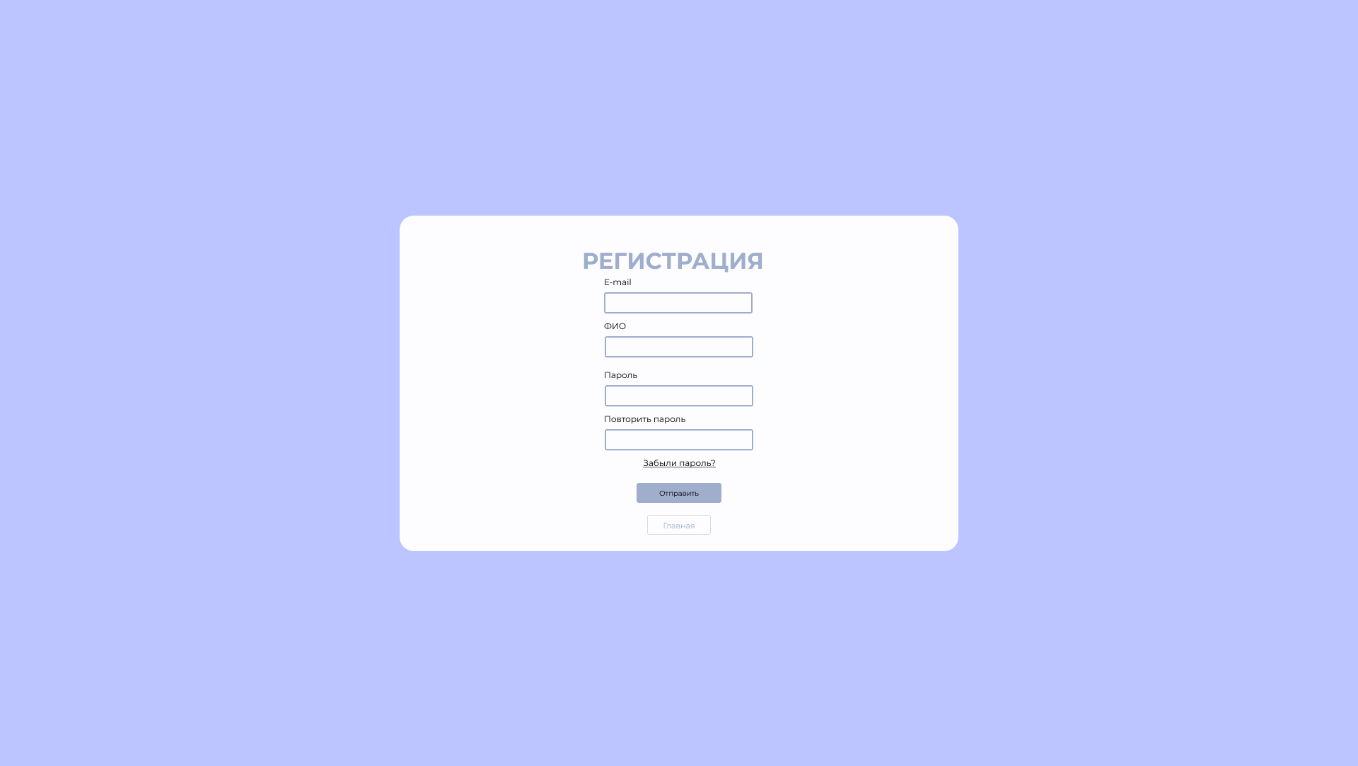
1. Макет страницы «Спасибо за заказ»



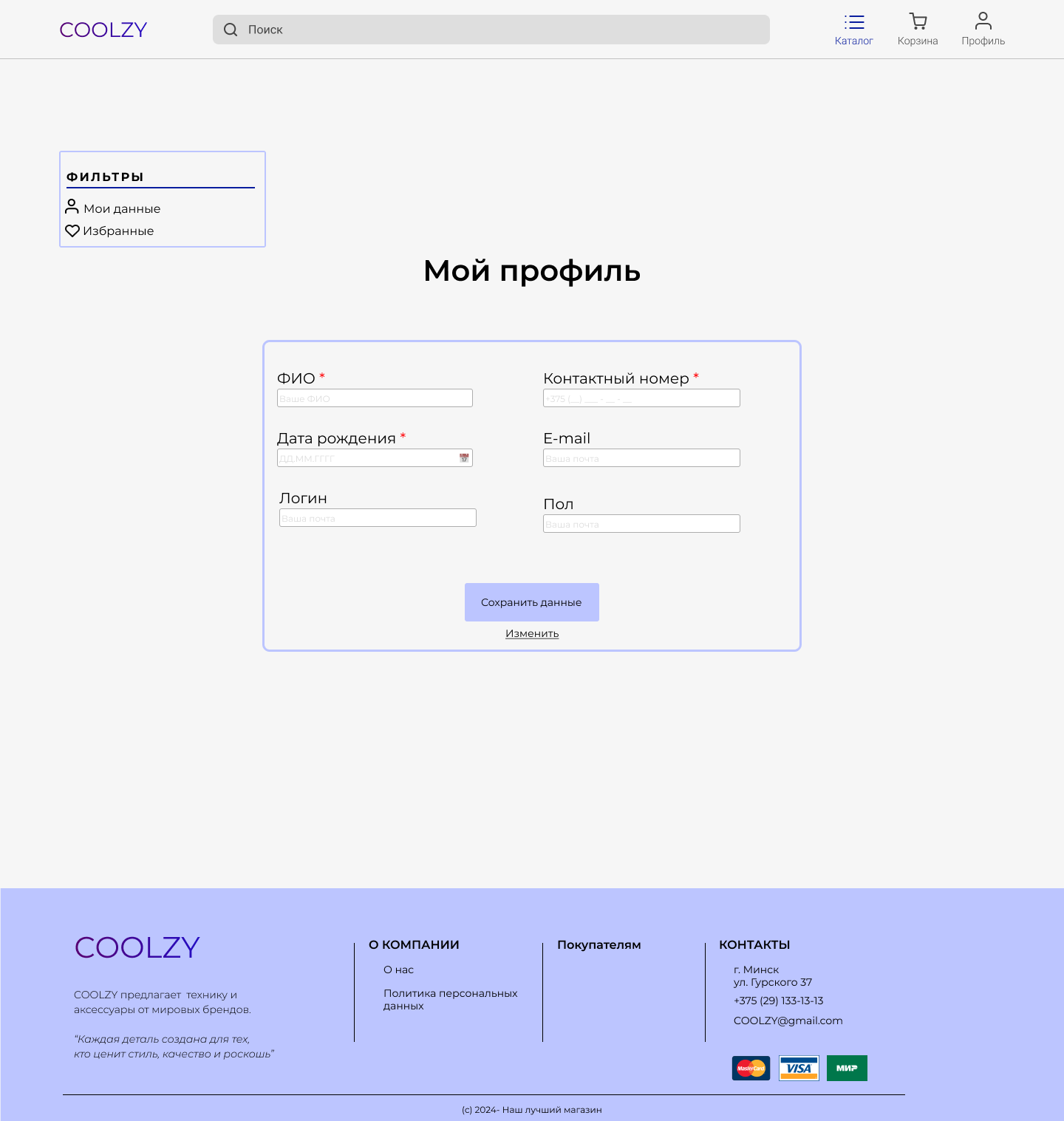
1. Макет страницы «Привязать новую карту»



1. Макет формы «Авторизации»



1. Макет формы «Регистрации»



1. Макет страницы «Профиль»

Приложение Г  
Код модулей веб – приложения

Admin.cs

using System.ComponentModel.DataAnnotations;

namespace TechnicalStoreBackEnd.Models

{

    public class Admin

    {

        [Key]

        public int AdminId { get; set; }

        public int UserId { get; set; }

        public User User { get; set; }

    }

}

AppDbContext.cs

using Microsoft.EntityFrameworkCore;

using TechnicalStoreBackEnd.Models;

namespace TechnicalStoreBackEnd.Models

{

    public class AppDbContext : DbContext

    {

        public AppDbContext(DbContextOptions<AppDbContext> options) : base(options) { }

      public DbSet<Review> Reviews { get; set; }

        public DbSet<User> Users { get; set; }

        public DbSet<Admin> Admins { get; set; }

        public DbSet<Category> Categories { get; set; }

        public DbSet<Product> Products { get; set; }

        public DbSet<ProductImage> ProductImages { get; set; }

        public DbSet<CartItem> CartItems { get; set; }

        public DbSet<Cart> Carts { get; set; }

        public DbSet<Order> Orders { get; set; }

        public DbSet<OrderItem> OrderItems { get; set; }

        public DbSet<FavoriteItem> FavoriteItems { get; set; }

        public DbSet<ProductSpecification> ProductSpecifications { get; set; }

        protected override void OnModelCreating(ModelBuilder modelBuilder)

        {

            base.OnModelCreating(modelBuilder);

            // Определение связей

            modelBuilder.Entity<FavoriteItem>().HasKey(f => f.Id);

            modelBuilder.Entity<FavoriteItem>().HasOne(f => f.User).WithMany().HasForeignKey(f => f.UserId);

            modelBuilder.Entity<FavoriteItem>().HasOne(f => f.Product).WithMany().HasForeignKey(f => f.ProductId);

            modelBuilder.Entity<Product>().HasOne(p => p.Category).WithMany(c => c.Products).HasForeignKey(p => p.CategoryId);

            modelBuilder.Entity<ProductImage>()

    .HasOne(p => p.Product)

    .WithMany(p => p.Images)

    .HasForeignKey(p => p.ProductId);

            modelBuilder.Entity<Admin>().HasOne(a => a.User).WithMany().HasForeignKey(a => a.UserId);

            modelBuilder.Entity<Order>().HasOne(o => o.User).WithMany().HasForeignKey(o => o.UserId);

            modelBuilder.Entity<OrderItem>().HasOne(oi => oi.Order).WithMany(o => o.Items).HasForeignKey(oi => oi.OrderId);

            modelBuilder.Entity<OrderItem>().HasOne(oi => oi.Product).WithMany().HasForeignKey(oi => oi.ProductId);

            modelBuilder.Entity<Cart>()

        .HasOne(c => c.User)

        .WithMany()

        .HasForeignKey(c => c.UserId);

            modelBuilder.Entity<CartItem>()

                .HasOne(ci => ci.Cart)

                .WithMany(c => c.Items)

                .HasForeignKey(ci => ci.CartId);

            modelBuilder.Entity<CartItem>()

                .HasOne(ci => ci.Product)

                .WithMany()

                .HasForeignKey(ci => ci.ProductId);

                // Связь отзывов с продуктами и пользователями

modelBuilder.Entity<Review>()

    .HasOne(r => r.Product)

    .WithMany()

    .HasForeignKey(r => r.ProductId)

    .OnDelete(DeleteBehavior.Cascade);

modelBuilder.Entity<Review>()

    .HasOne(r => r.User)

    .WithMany()

    .HasForeignKey(r => r.UserId)

    .OnDelete(DeleteBehavior.Cascade);

            modelBuilder.Entity<Category>().HasData(

     new Category { CategoryId = 1, Name = "Смартфоны" },

     new Category { CategoryId = 2, Name = "Телевизоры" },

     new Category { CategoryId = 3, Name = "Ноутбуки" },

     new Category { CategoryId = 4, Name = "Колонки" },

     new Category { CategoryId = 5, Name = "Планшеты" }

 );

            // Сидинг данных для Product и ProductImage

            modelBuilder.Entity<Product>().HasData(

                // Смартфоны (Категория 1)

                new Product

                {

                    ProductId = 100,

                    Model = "iPhone 14",

                    Brand = "Apple",

                    OldPrice = 1000m,

                    NewPrice = 900m,

                    Availability = "В наличии",

                    Ram = "8GB",

                    Storage = "128GB",

                    CategoryId = 1

                },

                new Product

                {

                    ProductId = 101,

                    Model = "Galaxy S23",

                    Brand = "Samsung",

                    OldPrice = 1100m,

                    NewPrice = 1000m,

                    Availability = "В наличии",

                    Ram = "8GB",

                    Storage = "128GB",

                    CategoryId = 1

                },

                new Product

                {

                    ProductId = 102,

                    Model = "Pixel 7",

                    Brand = "Google",

                    OldPrice = 900m,

                    NewPrice = 850m,

                    Availability = "В наличии",

                    Ram = "8GB",

                    Storage = "256GB",

                    CategoryId = 1

                },

                new Product

                {

                    ProductId = 103,

                    Model = "Xperia 1 IV",

                    Brand = "Sony",

                    OldPrice = 1200m,

                    NewPrice = 1100m,

                    Availability = "В наличии",

                    Ram = "12GB",

                    Storage = "256GB",

                    CategoryId = 1

                },

                new Product

                {

                    ProductId = 104,

                    Model = "OnePlus 11",

                    Brand = "OnePlus",

                    OldPrice = 950m,

                    NewPrice = 900m,

                    Availability = "В наличии",

                    Ram = "16GB",

                    Storage = "256GB",

                    CategoryId = 1

                },

                new Product

                {

                    ProductId = 105,

                    Model = "Redmi Note 12",

                    Brand = "Xiaomi",

                    OldPrice = 600m,

                    NewPrice = 550m,

                    Availability = "В наличии",

                    Ram = "6GB",

                    Storage = "128GB",

                    CategoryId = 1

                },

                new Product

                {

                    ProductId = 106,

                    Model = "Moto G Power",

                    Brand = "Motorola",

                    OldPrice = 500m,

                    NewPrice = 450m,

                    Availability = "В наличии",

                    Ram = "4GB",

                    Storage = "64GB",

                    CategoryId = 1

                },

                new Product

                {

                    ProductId = 107,

                    Model = "Find X5",

                    Brand = "Oppo",

                    OldPrice = 1000m,

                    NewPrice = 950m,

                    Availability = "В наличии",

                    Ram = "8GB",

                    Storage = "256GB",

                    CategoryId = 1

                },

                new Product

                {

                    ProductId = 108,

                    Model = "Zenfone 9",

                    Brand = "Asus",

                    OldPrice = 800m,

                    NewPrice = 750m,

                    Availability = "В наличии",

                    Ram = "8GB",

                    Storage = "128GB",

                    CategoryId = 1

                },

                new Product

                {

                    ProductId = 109,

                    Model = "Nord 3",

                    Brand = "OnePlus",

                    OldPrice = 700m,

                    NewPrice = 650m,

                    Availability = "В наличии",

                    Ram = "8GB",

                    Storage = "128GB",

                    CategoryId = 1

                },

                // Телевизоры (Категория 2)

                new Product

                {

                    ProductId = 200,

                    Model = "OLED55C2",

                    Brand = "LG",

                    OldPrice = 1500m,

                    NewPrice = 1400m,

                    Availability = "В наличии",

                    CategoryId = 2

                },

                new Product

                {

                    ProductId = 201,

                    Model = "QLED Q80B",

                    Brand = "Samsung",

                    OldPrice = 1600m,

                    NewPrice = 1500m,

                    Availability = "В наличии",

                    CategoryId = 2

                },

                new Product

                {

                    ProductId = 202,

                    Model = "Bravia XR",

                    Brand = "Sony",

                    OldPrice = 1700m,

                    NewPrice = 1600m,

                    Availability = "В наличии",

                    CategoryId = 2

                },

                new Product

                {

                    ProductId = 203,

                    Model = "Neo QLED",

                    Brand = "Samsung",

                    OldPrice = 2000m,

                    NewPrice = 1900m,

                    Availability = "В наличии",

                    CategoryId = 2

                },

                new Product

                {

                    ProductId = 204,

                    Model = "U8H",

                    Brand = "Hisense",

                    OldPrice = 1200m,

                    NewPrice = 1100m,

                    Availability = "В наличии",

                    CategoryId = 2

                },

                new Product

                {

                    ProductId = 205,

                    Model = "A90J",

                    Brand = "Sony",

                    OldPrice = 2500m,

                    NewPrice = 2400m,

                    Availability = "В наличии",

                    CategoryId = 2

                },

                new Product

                {

                    ProductId = 206,

                    Model = "C835",

                    Brand = "TCL",

                    OldPrice = 1300m,

                    NewPrice = 1200m,

                    Availability = "В наличии",

                    CategoryId = 2

                },

                new Product

                {

                    ProductId = 207,

                    Model = "QN85B",

                    Brand = "Samsung",

                    OldPrice = 1800m,

                    NewPrice = 1700m,

                    Availability = "В наличии",

                    CategoryId = 2

                },

                new Product

                {

                    ProductId = 208,

                    Model = "G2 OLED",

                    Brand = "LG",

                    OldPrice = 2200m,

                    NewPrice = 2100m,

                    Availability = "В наличии",

                    CategoryId = 2

                },

                new Product

                {

                    ProductId = 209,

                    Model = "R635",

                    Brand = "TCL",

                    OldPrice = 1000m,

                    NewPrice = 950m,

                    Availability = "В наличии",

                    CategoryId = 2

                },

                // Ноутбуки (Категория 3)

                new Product

                {

                    ProductId = 300,

                    Model = "MacBook Air M2",

                    Brand = "Apple",

                    OldPrice = 1300m,

                    NewPrice = 1200m,

                    Availability = "В наличии",

                    Ram = "8GB",

                    Storage = "256GB",

                    CategoryId = 3

                },

                new Product

                {

                    ProductId = 301,

                    Model = "XPS 13",

                    Brand = "Dell",

                    OldPrice = 1400m,

                    NewPrice = 1300m,

                    Availability = "В наличии",

                    Ram = "16GB",

                    Storage = "512GB",

                    CategoryId = 3

                },

                new Product

                {

                    ProductId = 302,

                    Model = "ThinkPad X1",

                    Brand = "Lenovo",

                    OldPrice = 1500m,

                    NewPrice = 1400m,

                    Availability = "В наличии",

                    Ram = "16GB",

                    Storage = "512GB",

                    CategoryId = 3

                },

                new Product

                {

                    ProductId = 303,

                    Model = "Spectre x360",

                    Brand = "HP",

                    OldPrice = 1600m,

                    NewPrice = 1500m,

                    Availability = "В наличии",

                    Ram = "16GB",

                    Storage = "1TB",

                    CategoryId = 3

                },

                new Product

                {

                    ProductId = 304,

                    Model = "ZenBook 14",

                    Brand = "Asus",

                    OldPrice = 1200m,

                    NewPrice = 1100m,

                    Availability = "В наличии",

                    Ram = "8GB",

                    Storage = "512GB",

                    CategoryId = 3

                },

                new Product

                {

                    ProductId = 305,

                    Model = "Surface Laptop 5",

                    Brand = "Microsoft",

                    OldPrice = 1400m,

                    NewPrice = 1300m,

                    Availability = "В наличии",

                    Ram = "8GB",

                    Storage = "256GB",

                    CategoryId = 3

                },

                new Product

                {

                    ProductId = 306,

                    Model = "Inspiron 14",

                    Brand = "Dell",

                    OldPrice = 1000m,

                    NewPrice = 950m,

                    Availability = "В наличии",

                    Ram = "8GB",

                    Storage = "256GB",

                    CategoryId = 3

                },

                new Product

                {

                    ProductId = 307,

                    Model = "Legion 5",

                    Brand = "Lenovo",

                    OldPrice = 1500m,

                    NewPrice = 1400m,

                    Availability = "В наличии",

                    Ram = "16GB",

                    Storage = "1TB",

                    CategoryId = 3

                },

                new Product

                {

                    ProductId = 308,

                    Model = "ROG Zephyrus",

                    Brand = "Asus",

                    OldPrice = 1800m,

                    NewPrice = 1700m,

                    Availability = "В наличии",

                    Ram = "16GB",

                    Storage = "1TB",

                    CategoryId = 3

                },

                new Product

                {

                    ProductId = 309,

                    Model = "Pavilion 15",

                    Brand = "HP",

                    OldPrice = 1100m,

                    NewPrice = 1000m,

                    Availability = "В наличии",

                    Ram = "8GB",

                    Storage = "512GB",

                    CategoryId = 3

                },

                // Колонки (Категория 4)

                new Product

                {

                    ProductId = 400,

                    Model = "HomePod Mini",

                    Brand = "Apple",

                    OldPrice = 100m,

                    NewPrice = 90m,

                    Availability = "В наличии",

                    CategoryId = 4

                },

                new Product

                {

                    ProductId = 401,

                    Model = "Echo Dot 5",

                    Brand = "Amazon",

                    OldPrice = 60m,

                    NewPrice = 50m,

                    Availability = "В наличии",

                    CategoryId = 4

                },

                new Product

                {

                    ProductId = 402,

                    Model = "SoundLink",

                    Brand = "Bose",

                    OldPrice = 150m,

                    NewPrice = 140m,

                    Availability = "В наличии",

                    CategoryId = 4

                },

                new Product

                {

                    ProductId = 403,

                    Model = "Nest Audio",

                    Brand = "Google",

                    OldPrice = 100m,

                    NewPrice = 90m,

                    Availability = "В наличии",

                    CategoryId = 4

                },

                new Product

                {

                    ProductId = 404,

                    Model = "Charge 5",

                    Brand = "JBL",

                    OldPrice = 180m,

                    NewPrice = 170m,

                    Availability = "В наличии",

                    CategoryId = 4

                },

                new Product

                {

                    ProductId = 405,

                    Model = "PartyBox 110",

                    Brand = "JBL",

                    OldPrice = 400m,

                    NewPrice = 380m,

                    Availability = "В наличии",

                    CategoryId = 4

                },

                new Product

                {

                    ProductId = 406,

                    Model = "Soundbar 300",

                    Brand = "Bose",

                    OldPrice = 450m,

                    NewPrice = 420m,

                    Availability = "В наличии",

                    CategoryId = 4

                },

                new Product

                {

                    ProductId = 407,

                    Model = "Roam",

                    Brand = "Sonos",

                    OldPrice = 180m,

                    NewPrice = 170m,

                    Availability = "В наличии",

                    CategoryId = 4

                },

                new Product

                {

                    ProductId = 408,

                    Model = "Flip 6",

                    Brand = "JBL",

                    OldPrice = 130m,

                    NewPrice = 120m,

                    Availability = "В наличии",

                    CategoryId = 4

                },

                new Product

                {

                    ProductId = 409,

                    Model = "One SL",

                    Brand = "Sonos",

                    OldPrice = 200m,

                    NewPrice = 190m,

                    Availability = "В наличии",

                    CategoryId = 4

                },

                // Планшеты (Категория 5)

                new Product

                {

                    ProductId = 500,

                    Model = "iPad Air 5",

                    Brand = "Apple",

                    OldPrice = 600m,

                    NewPrice = 550m,

                    Availability = "В наличии",

                    Ram = "8GB",

                    Storage = "64GB",

                    CategoryId = 5

                },

                new Product

                {

                    ProductId = 501,

                    Model = "Galaxy Tab S8",

                    Brand = "Samsung",

                    OldPrice = 700m,

                    NewPrice = 650m,

                    Availability = "В наличии",

                    Ram = "8GB",

                    Storage = "128GB",

                    CategoryId = 5

                },

                new Product

                {

                    ProductId = 502,

                    Model = "Surface Go 3",

                    Brand = "Microsoft",

                    OldPrice = 550m,

                    NewPrice = 500m,

                    Availability = "В наличии",

                    Ram = "4GB",

                    Storage = "64GB",

                    CategoryId = 5

                },

                new Product

                {

                    ProductId = 503,

                    Model = "MatePad 11",

                    Brand = "Huawei",

                    OldPrice = 500m,

                    NewPrice = 450m,

                    Availability = "В наличии",

                    Ram = "6GB",

                    Storage = "128GB",

                    CategoryId = 5

                },

                new Product

                {

                    ProductId = 504,

                    Model = "Tab P11 Pro",

                    Brand = "Lenovo",

                    OldPrice = 600m,

                    NewPrice = 550m,

                    Availability = "В наличии",

                    Ram = "6GB",

                    Storage = "128GB",

                    CategoryId = 5

                },

                new Product

                {

                    ProductId = 505,

                    Model = "iPad Pro 11",

                    Brand = "Apple",

                    OldPrice = 900m,

                    NewPrice = 850m,

                    Availability = "В наличии",

                    Ram = "8GB",

                    Storage = "256GB",

                    CategoryId = 5

                },

                new Product

                {

                    ProductId = 506,

                    Model = "Tab A8",

                    Brand = "Samsung",

                    OldPrice = 300m,

                    NewPrice = 280m,

                    Availability = "В наличии",

                    Ram = "4GB",

                    Storage = "64GB",

                    CategoryId = 5

                },

                new Product

                {

                    ProductId = 507,

                    Model = "Fire HD 10",

                    Brand = "Amazon",

                    OldPrice = 200m,

                    NewPrice = 180m,

                    Availability = "В наличии",

                    Ram = "3GB",

                    Storage = "32GB",

                    CategoryId = 5

                },

                new Product

                {

                    ProductId = 508,

                    Model = "Yoga Tab 13",

                    Brand = "Lenovo",

                    OldPrice = 700m,

                    NewPrice = 650m,

                    Availability = "В наличии",

                    Ram = "8GB",

                    Storage = "128GB",

                    CategoryId = 5

                },

                new Product

                {

                    ProductId = 509,

                    Model = "iPad Mini 6",

                    Brand = "Apple",

                    OldPrice = 500m,

                    NewPrice = 450m,

                    Availability = "В наличии",

                    Ram = "4GB",

                    Storage = "64GB",

                    CategoryId = 5

                }

            );

            // Сидинг данных для ProductImage

            modelBuilder.Entity<ProductImage>().HasData(

                // Смартфоны

                new ProductImage { ImageId = 1, ProductId = 100, ImageUrl = "/img/tovar/iphone14.jpg" },

                new ProductImage { ImageId = 2, ProductId = 101, ImageUrl = "/img/tovar/galaxys23.jpg" },

                new ProductImage { ImageId = 3, ProductId = 102, ImageUrl = "/img/tovar/pixel7.jpg" },

                new ProductImage { ImageId = 4, ProductId = 103, ImageUrl = "/img/tovar/xperia1iv.jpg" },

                new ProductImage { ImageId = 5, ProductId = 104, ImageUrl = "/img/tovar/oneplus11.jpg" },

                new ProductImage { ImageId = 6, ProductId = 105, ImageUrl = "/img/tovar/redminote12.jpg" },

                new ProductImage { ImageId = 7, ProductId = 106, ImageUrl = "/img/tovar/motogpower.jpg" },

                new ProductImage { ImageId = 8, ProductId = 107, ImageUrl = "/img/tovar/findx5.jpg" },

                new ProductImage { ImageId = 9, ProductId = 108, ImageUrl = "/img/tovar/zenfone9.jpg" },

                new ProductImage { ImageId = 10, ProductId = 109, ImageUrl = "/img/tovar/nord3.jpg" },

                // Телевизоры

                new ProductImage { ImageId = 11, ProductId = 200, ImageUrl = "/img/tovar/oled55c2.jpg" },

                new ProductImage { ImageId = 12, ProductId = 201, ImageUrl = "/img/tovar/qledq80b.jpg" },

                new ProductImage { ImageId = 13, ProductId = 202, ImageUrl = "/img/tovar/braviaxr.jpg" },

                new ProductImage { ImageId = 14, ProductId = 203, ImageUrl = "/img/tovar/neoqled.jpg" },

                new ProductImage { ImageId = 15, ProductId = 204, ImageUrl = "/img/tovar/u8h.jpg" },

                new ProductImage { ImageId = 16, ProductId = 205, ImageUrl = "/img/tovar/a90j.jpg" },

                new ProductImage { ImageId = 17, ProductId = 206, ImageUrl = "/img/tovar/c835.jpg" },

                new ProductImage { ImageId = 18, ProductId = 207, ImageUrl = "/img/tovar/qn85b.jpg" },

                new ProductImage { ImageId = 19, ProductId = 208, ImageUrl = "/img/tovar/g2oled.jpg" },

                new ProductImage { ImageId = 20, ProductId = 209, ImageUrl = "/img/tovar/r635.jpg" },

                // Ноутбуки

                new ProductImage { ImageId = 21, ProductId = 300, ImageUrl = "/img/tovar/macbookairm2.jpg" },

                new ProductImage { ImageId = 22, ProductId = 301, ImageUrl = "/img/tovar/xps13.jpg" },

                new ProductImage { ImageId = 23, ProductId = 302, ImageUrl = "/img/tovar/thinkpadx1.jpg" },

                new ProductImage { ImageId = 24, ProductId = 303, ImageUrl = "/img/tovar/spectrex360.jpg" },

                new ProductImage { ImageId = 25, ProductId = 304, ImageUrl = "/img/tovar/zenbook14.jpg" },

                new ProductImage { ImageId = 26, ProductId = 305, ImageUrl = "/img/tovar/surfacelaptop5.jpg" },

                new ProductImage { ImageId = 27, ProductId = 306, ImageUrl = "/img/tovar/inspiron14.jpg" },

                new ProductImage { ImageId = 28, ProductId = 307, ImageUrl = "/img/tovar/legion5.jpg" },

                new ProductImage { ImageId = 29, ProductId = 308, ImageUrl = "/img/tovar/rogzephyrus.jpg" },

                new ProductImage { ImageId = 30, ProductId = 309, ImageUrl = "/img/tovar/pavilion15.jpg" },

                // Колонки

                new ProductImage { ImageId = 31, ProductId = 400, ImageUrl = "/img/tovar/homepodmini.jpg" },

                new ProductImage { ImageId = 32, ProductId = 401, ImageUrl = "/img/tovar/echodot5.jpg" },

                new ProductImage { ImageId = 33, ProductId = 402, ImageUrl = "/img/tovar/soundlink.jpg" },

                new ProductImage { ImageId = 34, ProductId = 403, ImageUrl = "/img/tovar/nestaudio.jpg" },

                new ProductImage { ImageId = 35, ProductId = 404, ImageUrl = "/img/tovar/charge5.jpg" },

                new ProductImage { ImageId = 36, ProductId = 405, ImageUrl = "/img/tovar/partybox110.jpg" },

                new ProductImage { ImageId = 37, ProductId = 406, ImageUrl = "/img/tovar/soundbar300.jpg" },

                new ProductImage { ImageId = 38, ProductId = 407, ImageUrl = "/img/tovar/roam.jpg" },

                new ProductImage { ImageId = 39, ProductId = 408, ImageUrl = "/img/tovar/flip6.jpg" },

                new ProductImage { ImageId = 40, ProductId = 409, ImageUrl = "/img/tovar/onesl.jpg" },

                // Планшеты

                new ProductImage { ImageId = 41, ProductId = 500, ImageUrl = "/img/tovar/ipadair5.jpg" },

                new ProductImage { ImageId = 42, ProductId = 501, ImageUrl = "/img/tovar/galaxytabs8.jpg" },

                new ProductImage { ImageId = 43, ProductId = 502, ImageUrl = "/img/tovar/surfacego3.jpg" },

                new ProductImage { ImageId = 44, ProductId = 503, ImageUrl = "/img/tovar/matepad11.jpg" },

                new ProductImage { ImageId = 45, ProductId = 504, ImageUrl = "/img/tovar/tabp11pro.jpg" },

                new ProductImage { ImageId = 46, ProductId = 505, ImageUrl = "/img/tovar/ipadpro11.jpg" },

                new ProductImage { ImageId = 47, ProductId = 506, ImageUrl = "/img/tovar/taba8.jpg" },

                new ProductImage { ImageId = 48, ProductId = 507, ImageUrl = "/img/tovar/firehd10.jpg" },

                new ProductImage { ImageId = 49, ProductId = 508, ImageUrl = "/img/tovar/yogatab13.jpg" },

                new ProductImage { ImageId = 50, ProductId = 509, ImageUrl = "/img/tovar/ipadmini6.jpg" }

            );

        }

    }

}

Cart.cs

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

namespace TechnicalStoreBackEnd.Models

{

    public class Cart

    {

        [Key]

        public int CartId { get; set; }

        public int UserId { get; set; }

        public User User { get; set; }

        public ICollection<CartItem> Items { get; set; } = new List<CartItem>();

        [Column(TypeName = "decimal(18,2)")]

        public decimal TotalPrice { get; set; }

    }

}

CartItem.cs

namespace TechnicalStoreBackEnd.Models

{

    public class CartItem

    {

        public int Id { get; set; }

        public int CartId { get; set; } // Добавлено для связи с Cart

        public int ProductId { get; set; }

        public int Quantity { get; set; }

        public Cart Cart { get; set; }

        public Product Product { get; set; }

    }

}

Category.cs

namespace TechnicalStoreBackEnd.Models

{

    public class Category

    {

        public int CategoryId { get; set; }

        public string Name { get; set; } = string.Empty; // 🔹 для поиска

        public ICollection<Product> Products { get; set; } = new List<Product>();

    }

}

ErrorViewModel.cs

namespace TechnicalStoreBackEnd.Models;

public class ErrorViewModel

{

    public string? RequestId { get; set; }

    public bool ShowRequestId => !string.IsNullOrEmpty(RequestId);

}

FavoriteItem.cs

using TechnicalStoreBackEnd.Models;

public class FavoriteItem

{

    public int Id { get; set; }

    public int UserId { get; set; }

    public User User { get; set; }

    public int ProductId { get; set; }

    public Product Product { get; set; }

}

Order.cs

using System.Collections.Generic;

using TechnicalStoreBackEnd.Models; // ← это ОБЯЗАТЕЛЬНО

using System.Linq;

namespace TechnicalStoreBackEnd.Models

{

    public class Order

    {

        public int OrderId { get; set; }

        public int UserId { get; set; }

        public DateTime CreatedAt { get; set; }

        public User? User { get; set; }

        public string? FullName { get; set; }

        public string? Address { get; set; }

        public string? Phone { get; set; }

        public string? PaymentMethod { get; set; }

        public List<OrderItem> Items { get; set; } = new();

        public decimal TotalAmount => Items?.Sum(i => i.Price \* i.Quantity) ?? 0m;

    }

}

OrderItem.cs

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

namespace TechnicalStoreBackEnd.Models

{ OrderItem

    public class OrderItem

    {

        [Key]

        public int OrderItemId { get; set; }

        public int OrderId { get; set; }

        public Order? Order { get; set; }

        public int ProductId { get; set; }

        public Product? Product { get; set; }

        public int Quantity { get; set; }

        [Column(TypeName = "decimal(18,2)")]

        public decimal Price { get; set; } // фиксированная цена на момент покупки

    }

}

Product.cs

using System.ComponentModel.DataAnnotations;

namespace TechnicalStoreBackEnd.Models

{

    public class Product

    {

        [Key]

        public int ProductId { get; set; }

        public int Quantity { get; set; }

        public string Name { get; set; } = string.Empty;            // 🔹 Название товара

        public string Description { get; set; } = string.Empty;     // 🔹 Описание

        public string Model { get; set; } = string.Empty;

        public string Brand { get; set; } = string.Empty;

        public decimal OldPrice { get; set; }

        public decimal NewPrice { get; set; }

        public string Availability { get; set; } = string.Empty;

        public string? Ram { get; set; }

        public string? Storage { get; set; }

        public int CategoryId { get; set; }

        public Category Category { get; set; } = default!;

        public ICollection<ProductImage> Images { get; set; } = new List<ProductImage>();

    }

}

ProductImage.cs

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

namespace TechnicalStoreBackEnd.Models

{

    public class ProductImage

    {

        [Key]

        public int ImageId { get; set; }

        public int ProductId { get; set; }

        public string ImageUrl { get; set; } = string.Empty;

        [ForeignKey("ProductId")]

        public Product Product { get; set; } = default!;

    }

}

ProductSpecification.cs

namespace TechnicalStoreBackEnd.Models

{

    public class ProductSpecification

    {

        public int Id { get; set; }

        public int ProductId { get; set; }

        public string GroupName { get; set; }     // "Основные", "Экран" и т.д.

        public string AttributeName { get; set; } // "Бренд", "Тип телефона"

        public string AttributeValue { get; set; } // "Xiaomi", "AMOLED"

        public Product Product { get; set; }

    }

}

RegisterRequest.cs

public class RegisterRequest

{

    public string Username { get; set; }

    public string Email { get; set; }

    public string Password { get; set; }

}

Review.cs

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

namespace TechnicalStoreBackEnd.Models

{

    public class Review

    {

        [Key]

        public int Id { get; set; }

        [Required]

        public int ProductId { get; set; }

        [ForeignKey("ProductId")]

        public Product Product { get; set; }

        [Required]

        public int UserId { get; set; }

        [ForeignKey("UserId")]

        public User User { get; set; }

        public string UserName { get; set; } = string.Empty;

        [Range(1, 5)]

        public int Rating { get; set; }

        [Required]

        public string Comment { get; set; } = string.Empty;

        public DateTime CreatedAt { get; set; } = DateTime.UtcNow;

    }

}

User

using System.ComponentModel.DataAnnotations;

namespace TechnicalStoreBackEnd.Models

{

    public class User

    {

        [Key]

        public int UserId { get; set; }

        public string Username { get; set; }

        public string Email { get; set; }

        public string PasswordHash { get; set; }

        public string? Phone { get; set; }

        public string? Gender { get; set; }

        public DateTime? BirthDate { get; set; }

        public DateTime RegisteredAt { get; set; } = DateTime.UtcNow;

        public DateTime? LastLogin { get; set; }

        public bool IsAdmin { get; set; } = false;

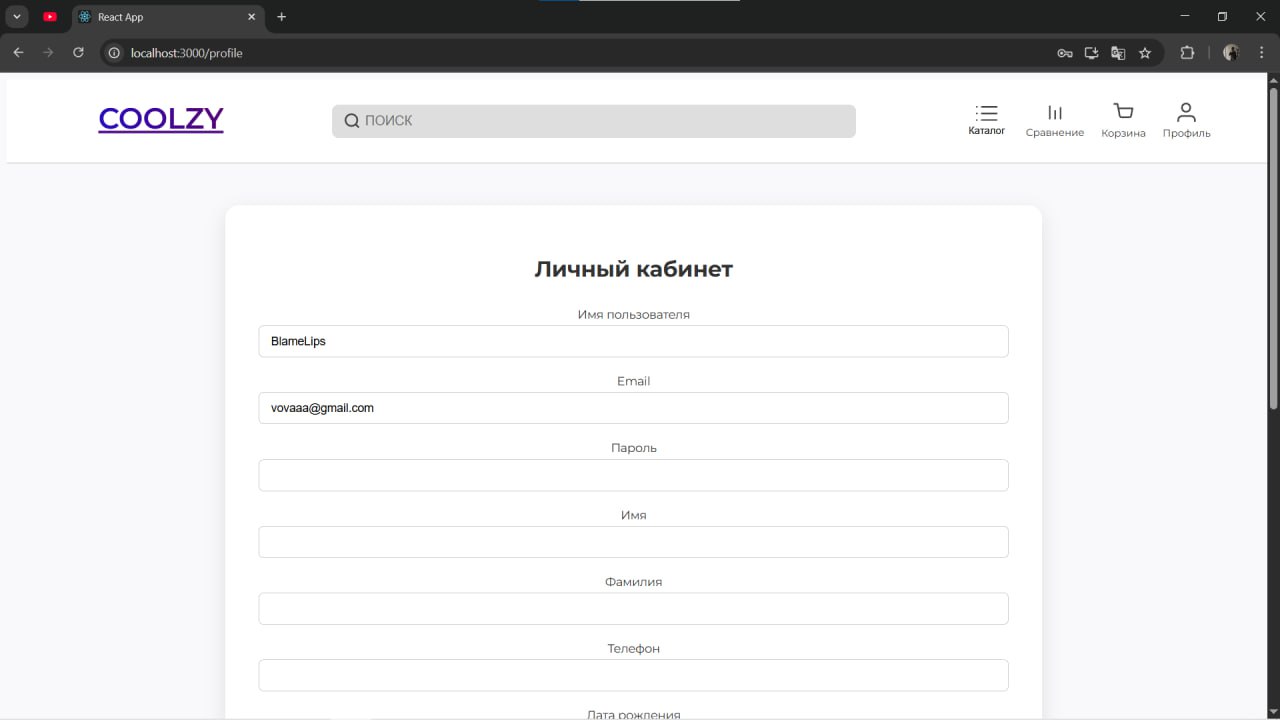
        public string? FirstName { get; set; }

        public string? LastName { get; set; }

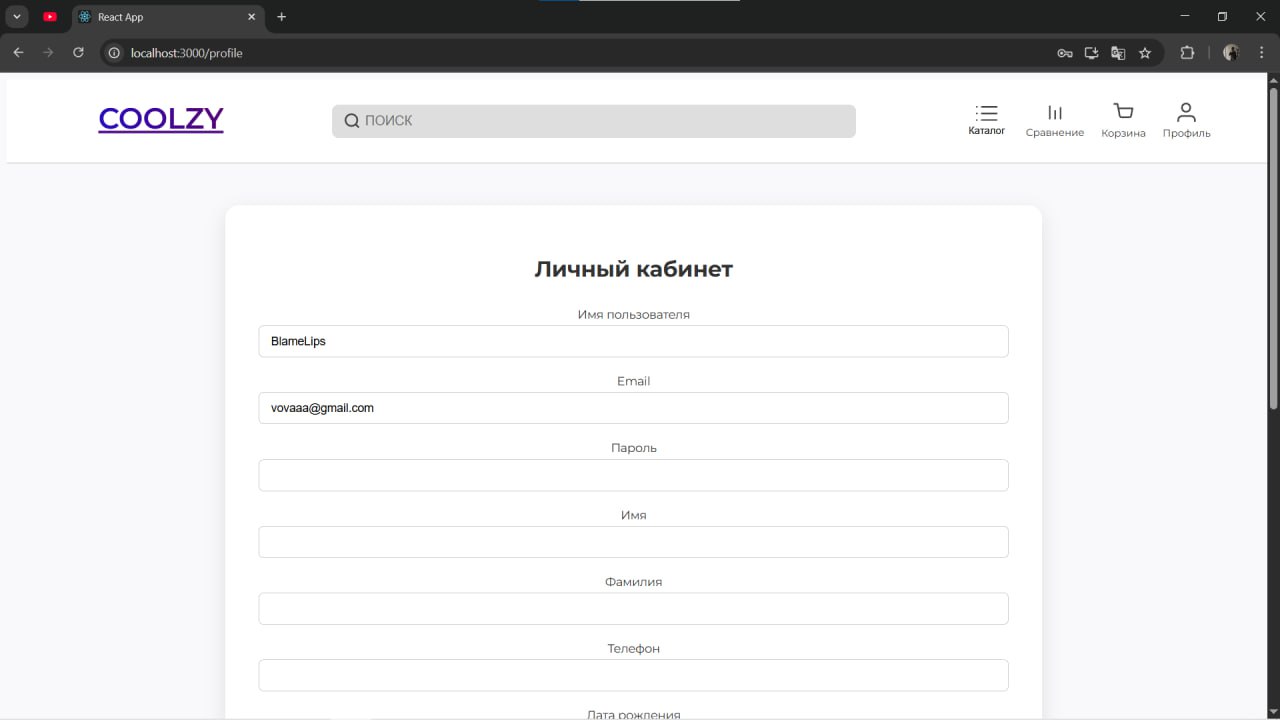
    }

}

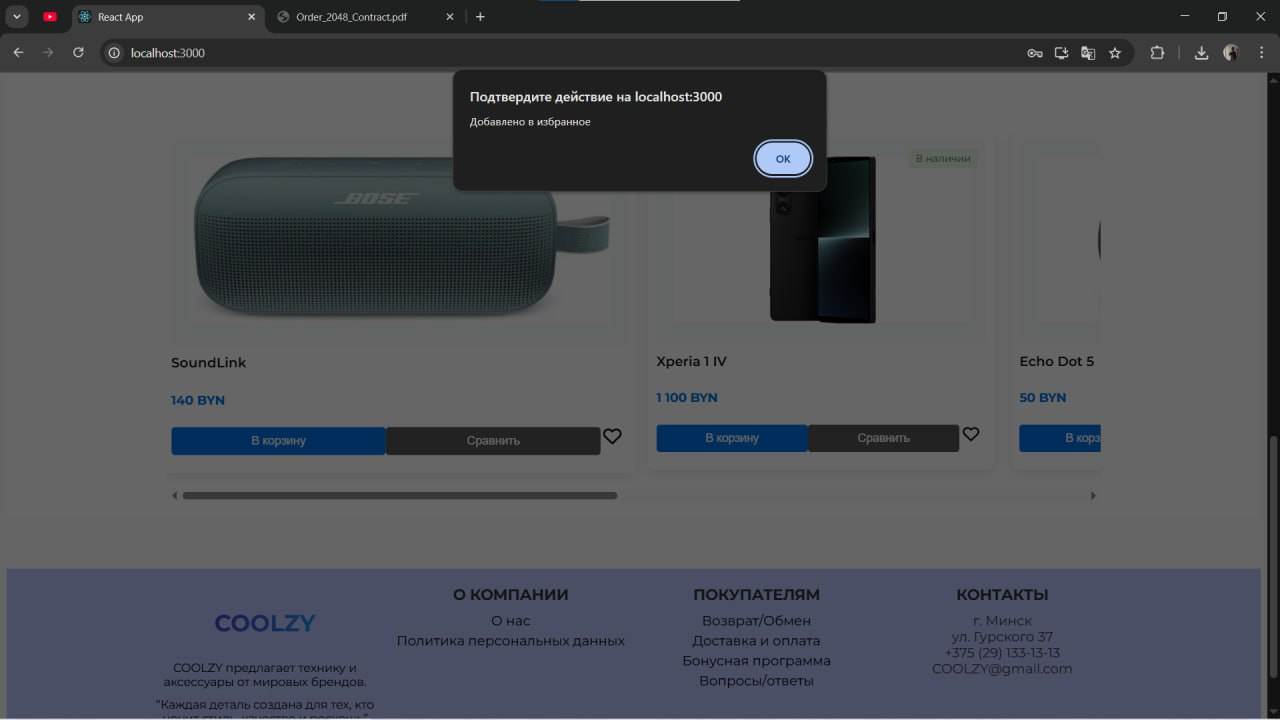
Приложение Д  
Результаты работы веб – приложения



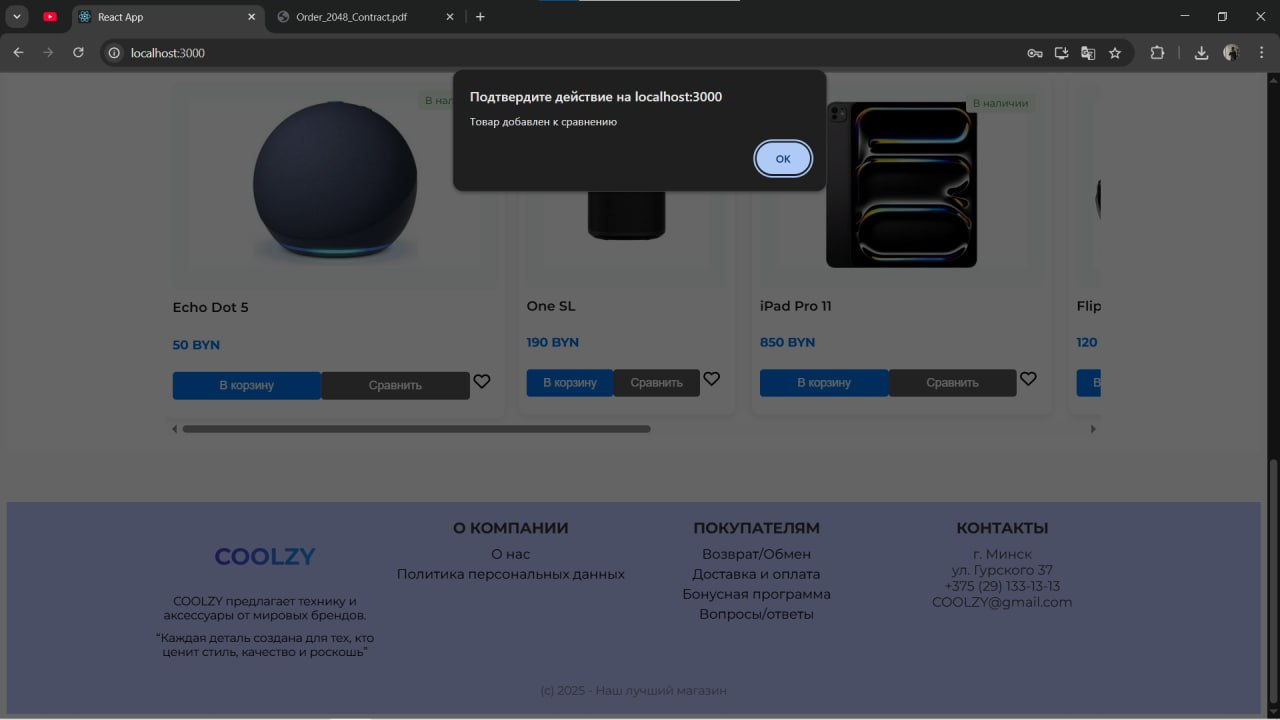
1. ‒ Результат тестирование «Регистрация»



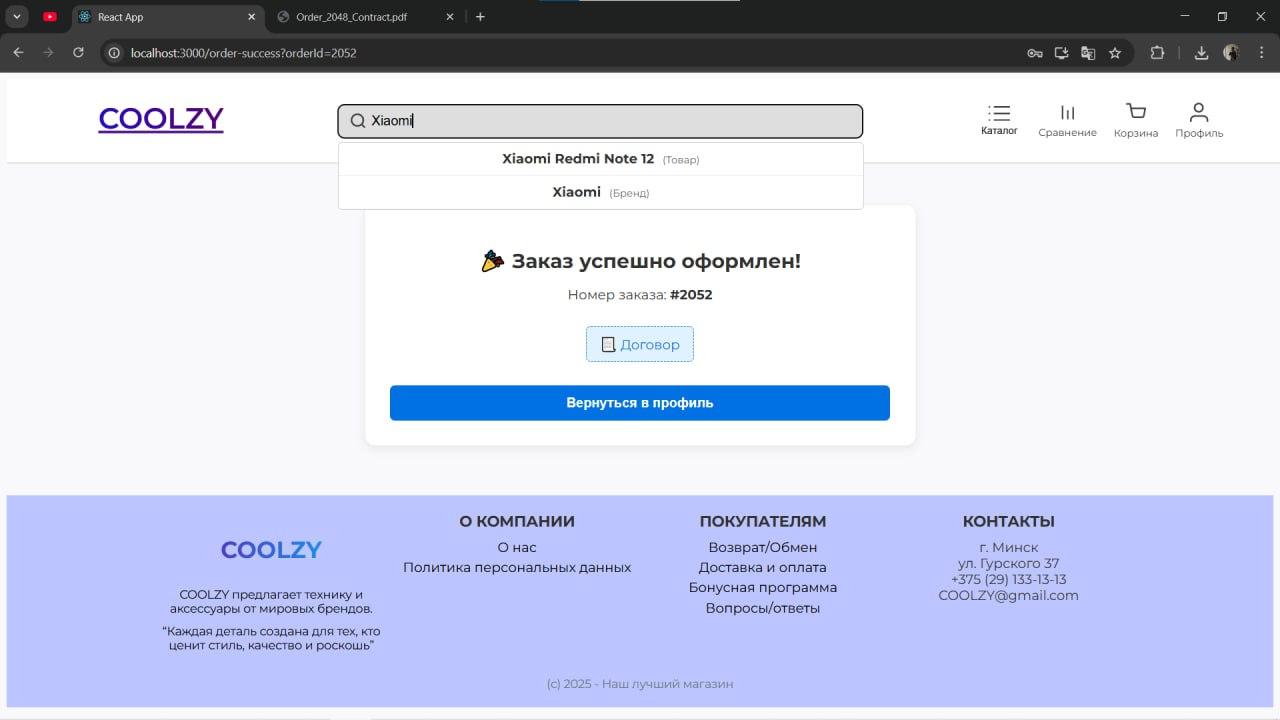
1. ‒ Результат тестирование «Авторизация»



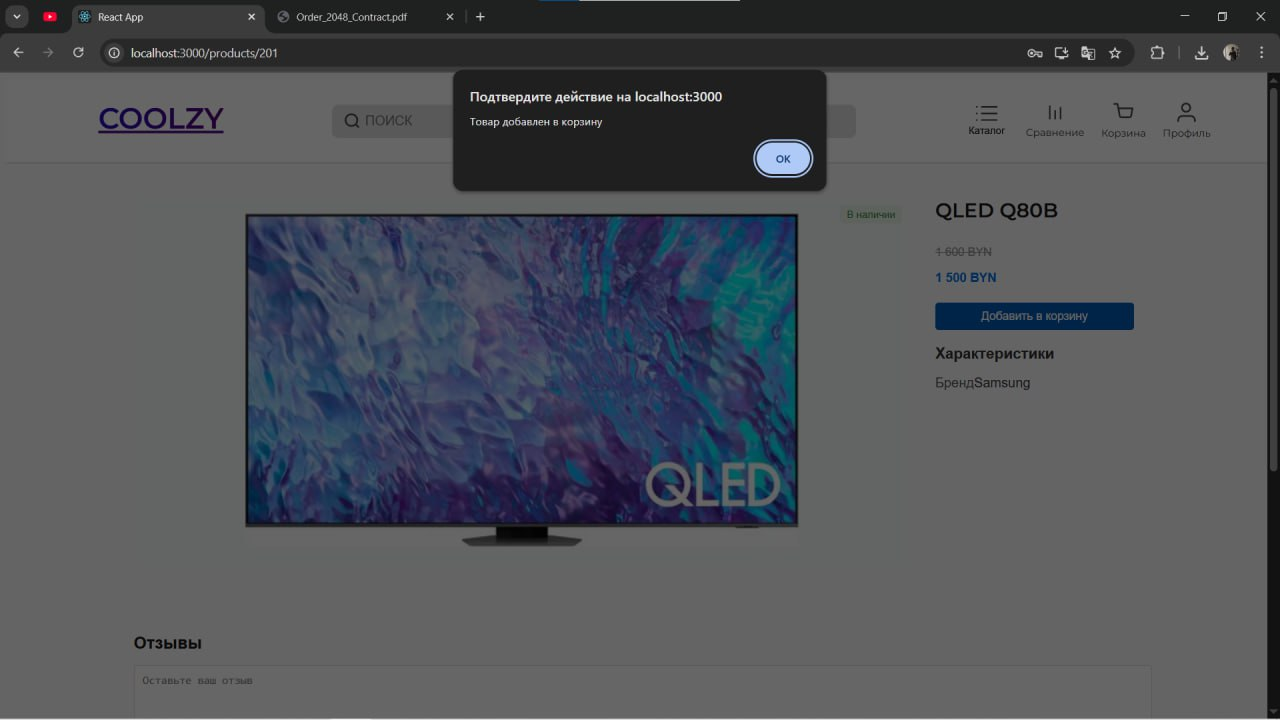
1. ‒ Результат тестирование «Добавление товара в избранные»



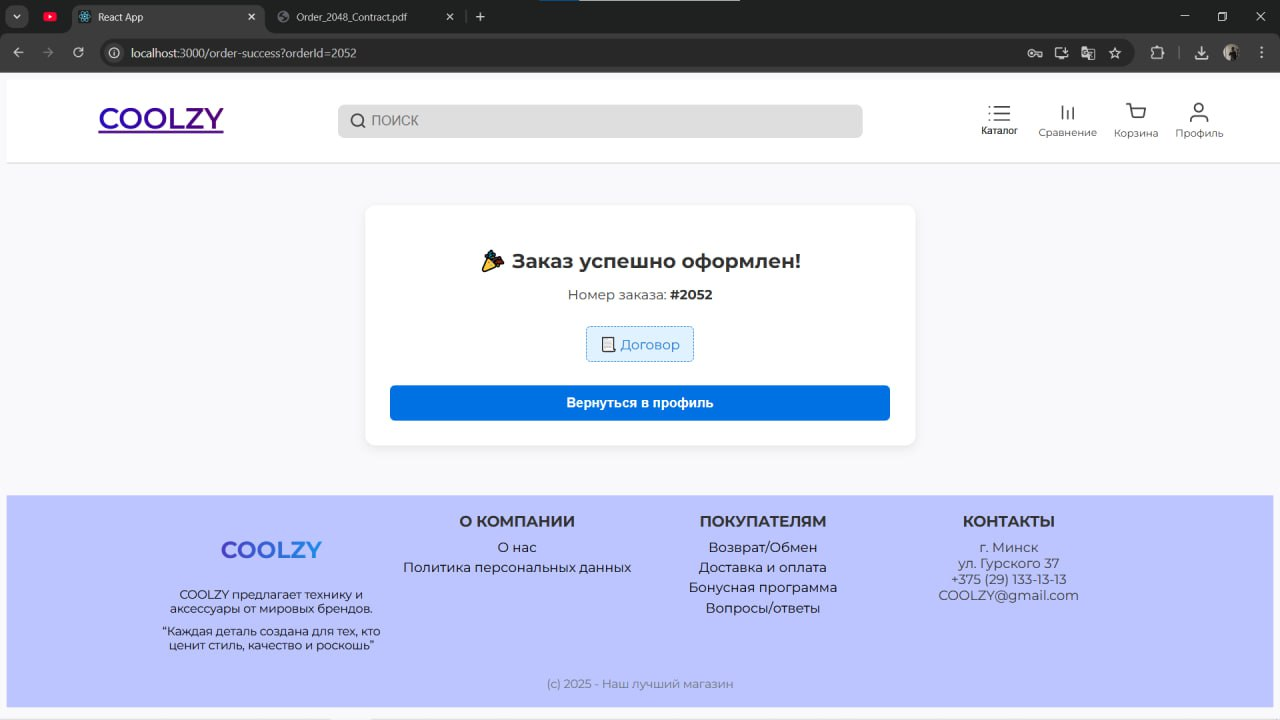
1. ‒ Результат тестирование «Добавление товара в сравнение»



1. ‒ Результат тестирование «Поиск»



1. ‒ Результат тестирование «Добавление товара в корзину»



1. ‒ Результат тестирование «Оформление заказа»

КП Т.319003.401 ГЧ

КБП

У

Т. Контр.

Разраб. Разраб.

Григоренко В.Р.

Провер. Провер.

Кривошеина А.В.

Реценз.

Утверд.

Н. Контр.

Диаграмма прецедентов

№ докум.

Подпись

Дата

Изм.

Лист

Масса

Лит.

Масштаб

Лист 1

Листов 3

Инв.№подл.

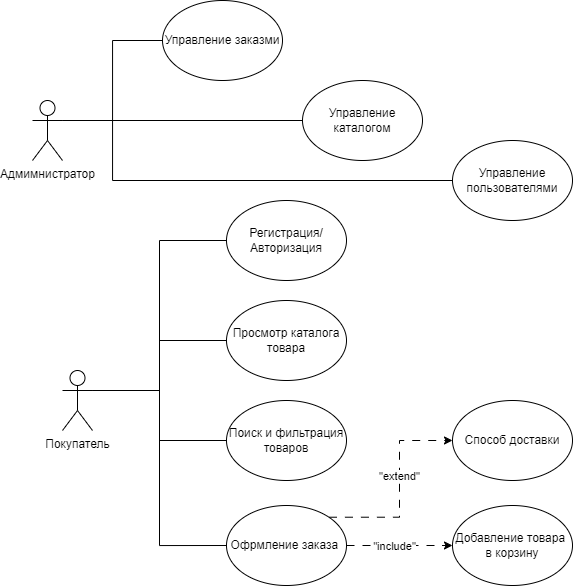
Подп. и дата

Взам.инв.№

Инв.№дубл.

Подп. и дата

КП Т.319003.401 ГЧ Т.ХХХХХХ.401 ГЧ



*Разработка веб-приложения для автоматизации работы маркетплейса компьютерной техники «COOLZY»*

*Разработка веб-приложения по автоматизации работы сети магазинов техники «COOLZY»*

КП Т.319003.401 ГЧ

КБП

У

Т. Контр.

Разраб. Разраб.

Григоренко В.Р.

Провер. Провер.

Кривошеина А.В.

Реценз.

Утверд.

Н. Контр.

№ докум.

Подпись

Дата

Изм.

Лист

Масса

Лит.

Масштаб

Лист 2

Листов 3

Инв.№подл.

Подп. и дата

Взам.инв.№

Инв.№дубл.

Подп. и дата

КП Т.319003.401 ГЧ Т.ХХХХХХ.401 ГЧ

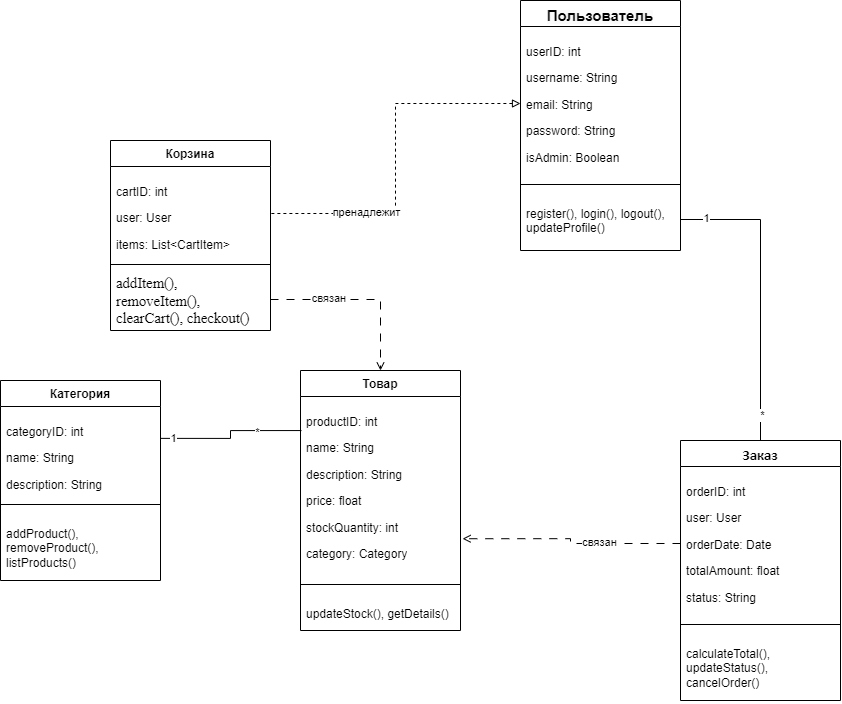


Диаграмма классов

*Разработка веб-приложения для автоматизации работы маркетплейса компьютерной техники «COOLZY»*

КП Т.319003.401 ГЧ

КБП

У

Т. Контр.

Разраб. Разраб.

Григоренко В.Р.

Провер. Провер.

Кривошеина А.В.

Реценз.

Утверд.

Н. Контр.

№ докум.

Подпись

Дата

Изм.

Лист

Масса

Лит.

Масштаб

Лист 3

Листов 3

Инв.№подл.

Подп. и дата

Взам.инв.№

Инв.№дубл.

Подп. и дата

КП Т.095000.401 ГЧ Т.ХХХХХХ.401 ГЧ

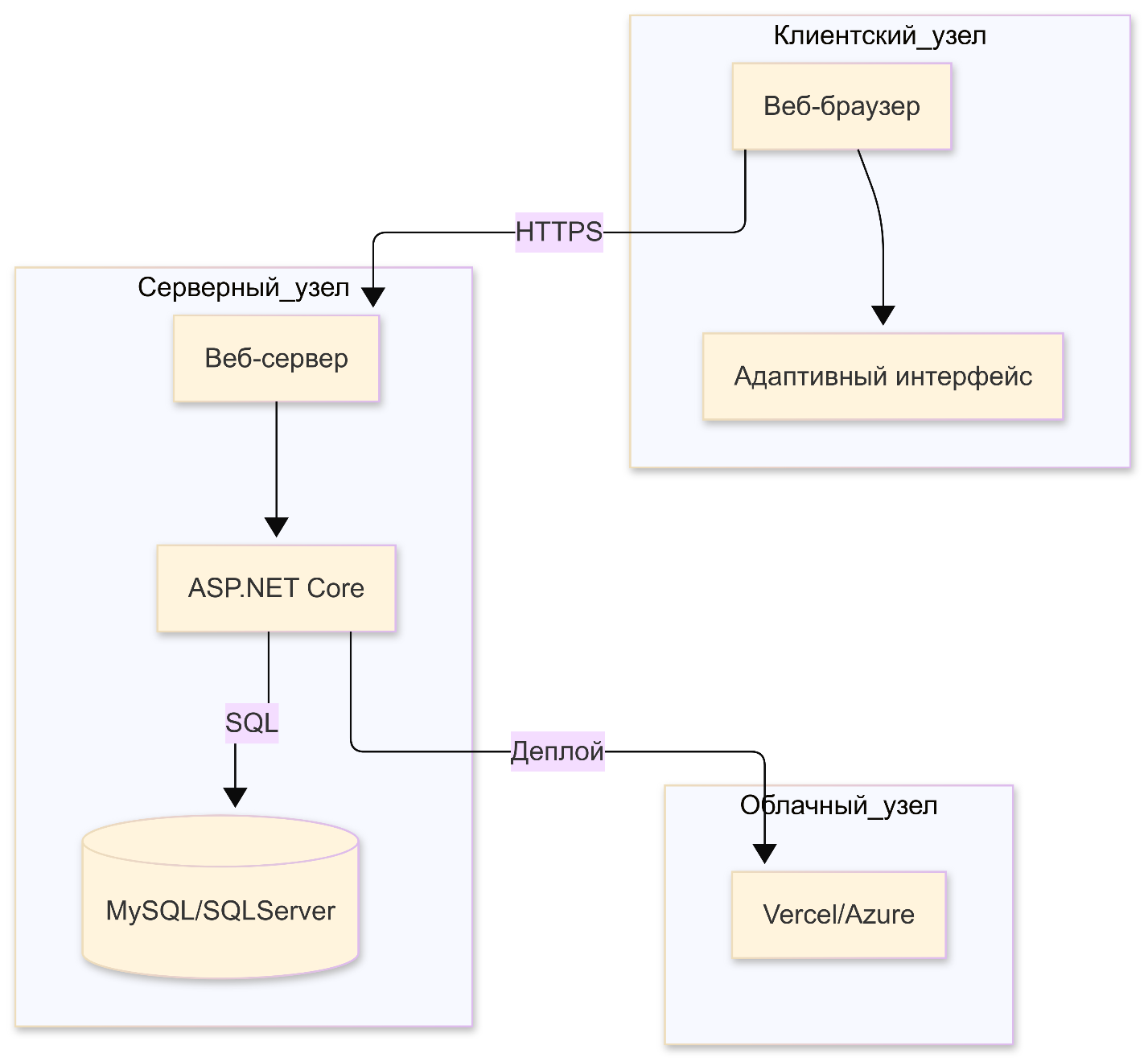


Диаграмма развертывания

*Разработка веб-приложения для автоматизации работы маркетплейса компьютерной техники «COOLZY»*

**Удостоверяющий лист**

электронного документа – курсовой проект

Тема КПРазработка веб-приложения по автоматизации работы сети магазинов техники «COOLZY»

Обозначение КП Т.319003.401

Разработчик Григоренко В.Р. Руководитель Кривошеина А.В.

(Ф.И.О.) (Ф.И.О.)

Подписи лиц, ответственных за разработку электронного документа

|  |  |  |
| --- | --- | --- |
| Состав электронного документа | Разработчик | Руководитель |
| Пояснительная записка (на бумажном носителе формата А4), ПЗ\_Григоренко\_т–319.docх |  |  |
| ГЧ, Диаграммы.docx |  |  |
| Папка с проектом «kursach» |  |  |
| Файл базы данных «technical\_store» |  |  |
| Тип носителя: USB – флешка |  |  |

**Этикетка**

для курсового проекта

**Курсовой проект**

Тема «Разработка веб-приложения для автоматизации работы маркетплейса компьютерной техники «COOLZY»

КП Т.319003.401

Разработан 01.06.2006

Разработчик: Григоренко В.Р.

Руководитель: Кривошеина А.В.

Технические средства: Персональный компьютер с процессором Intel Core i5 или выше, не менее 8 ГБ оперативной памяти, жёсткий диск объёмом от 100 ГБ, монитор с разрешением не ниже 1920×1080, клавиатура, мышь.

Программные средства: Операционная система Windows 10/11, среда разработки Microsoft Visual Studio 2022, платформа .NET 6/7 с использованием ASP.NET Core MVC, система управления базами данных Microsoft SQL Server, браузер Google Chrome, фреймворк React для клиентской части

**Состав документа:**

Пояснительная записка – ПЗ\_Григоренко\_т-319.docх

Графическая часть –Диаграммы.docx

Папка с проектом – kursach

Файл базы данных – technical\_store