

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Самарский национальный исследовательский университет
имени академика С.П. Королева»
Институт информатики, математики и электроники
Факультет информатики
Кафедра технической кибернетики

Отчет по курсу
Дисциплина: «Технологии сетевого программирования»

Выполнили:
Абубекаров Н.Р.
Хромов А.В.

Группа:
6301-010302D

Самара 2025

1 Общая архитектура приложения

1.1 Описание приложения

Данное веб-приложение представляет собой интернет-магазин с каталогом товаров, возможностью авторизации пользователей, оформления заказов, управления корзиной, профилем пользователя и просмотра информации о товарах.

1.2 Состав приложения

Приложение состоит из следующих функциональных частей:

1. Серверная часть приложения.
2. Клиентская часть приложения.
3. База данных.

1.3 Описание функциональных частей

1.3.1 Серверная часть

Серверная часть написана на языке программирования Java с использованием фреймворка Spring Boot. Проект использует Spring Security для авторизации и аутентификации, Spring Web для построения REST API и Spring Data JPA для работы с базой данных. Безопасность реализована на основе JWT с поддержкой обновления токенов. Сборка и управление зависимостями выполняются с помощью Maven.

1.3.2 Клиентская часть

Клиентская часть реализована на языке JavaScript с использованием фреймворка React 19. Библиотека React Router DOM используется для маршрутизации в проекте, React Icons предоставляет популярные иконки, которые были использованы на сайте, Axios для взаимодействия с REST API, а Zustand для управления состоянием приложения. Также были использованы HTML и модульные стили CSS.

1.3.3 База данных

В качестве СУБД используется PostgreSQL. А для работы с ней применяются ORM модели.

Настройка БД:

spring:

datasource:

url:

\${SPRING_DATASOURCE_URL:jdbc:postgresql://localhost:5433/furniture_db}

username: \${SPRING_DATASOURCE_USERNAME:furniture_user}

password: \${SPRING_DATASOURCE_PASSWORD:furniture_pass}

driver-class-name: org.postgresql.Driver

jpa:

hibernate:

ddl-auto: none

show-sql: true

properties:

hibernate:

format_sql: true

flyway:

enabled: true

locations: classpath:db/migration

jwt:

secret: \${JWT_SECRET:mySuperSecretKeyThatIsAtLeast32BytesLong!}

access-expiration-ms: \${JWT_ACCESS_EXPIRATION_MS:15000}

refresh-expiration-ms: \${JWT_REFRESH_EXPIRATION_MS:2592000000}

1.4 Система авторизации и безопасности

Система авторизации реализована через JWT (JSON Web Token). Используются access и refresh токены. Access используется для краткосрочного

доступа (время его действия принято устанавливать не больше 2 часов), а refresh для обновления access токена (время действия от 7 до 30 дней).

1.5 Функционирование приложения

Приложение собрано и запускается с помощью файла docker-compose.yml, находящегося в корне проекта и который объединяет 3 контейнера (бэкенд, фронтенд и базу данных).

Параметры для запуска контейнера с базой данных:

db:

image: postgres:15-alpine

container_name: furniture_store_db

environment:

- POSTGRES_USER=furniture_user

- POSTGRES_PASSWORD=furniture_pass

- POSTGRES_DB=furniture_db

ports:

- "5433:5432"

volumes:

- db_data:/var/lib/postgresql/data

networks:

- app-network

Параметры для запуска контейнера с бэкендом:

backend:

build:

context: ./backend

dockerfile: Dockerfile

container_name: furniture_store_backend

environment:

-

SPRING_DATASOURCE_URL=jdbc:postgresql://db:5432/furniture_db

- *SPRING_DATASOURCE_USERNAME=furniture_user*
- *SPRING_DATASOURCE_PASSWORD=furniture_pass*
- *JWT_SECRET=mySuperSecretKeyThatIsAtLeast32BytesLong!*
- *JWT_ACCESS_EXPIRATION_MS=15000*
- *JWT_REFRESH_EXPIRATION_MS=2592000000*

ports:

- *"8080:8080"*

depends_on:

- *db*

networks:

- *app-network*

Параметры для запуска контейнера с фронтом:

frontend:

build:

context: ./frontend # Путь к папке с Dockerfile фронтенда

dockerfile: Dockerfile

container_name: furniture_store_frontend

ports:

- *"80:80"*

environment:

- *API_BASE_URL=http://localhost:8080*

depends_on:

- *backend*

networks:

- *app-network*

2 Работа приложения

2.1 Регистрация и авторизация

На рисунке 1 представлена страница регистрации, где пользователь вводит свои данные (имя, фамилию, электронную почту, пароль, номер телефона и адрес). Зарегистрировать пользователя с уже имеющейся почтой в БД не выйдет. После успешной регистрации, пользователь может войти в магазин на странице входа, показанной на рисунке 2.

Регистрация

[Зарегистрироваться](#)

[На страницу входа](#)

Рисунок 1 – Страница регистрации

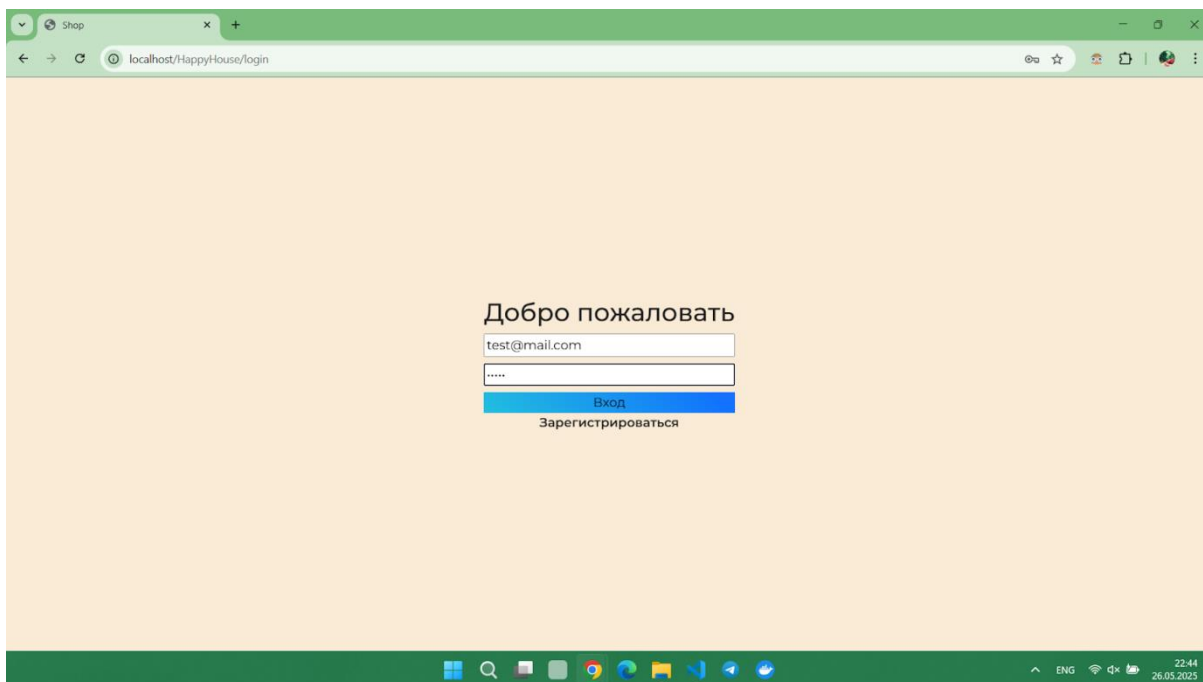


Рисунок 2 – Страница входа

Далее открывается главная страница магазина, изображенная на рисунках 3, 4, на которой можно просматривать товары. Сверху находится панель навигации с главными страницами.

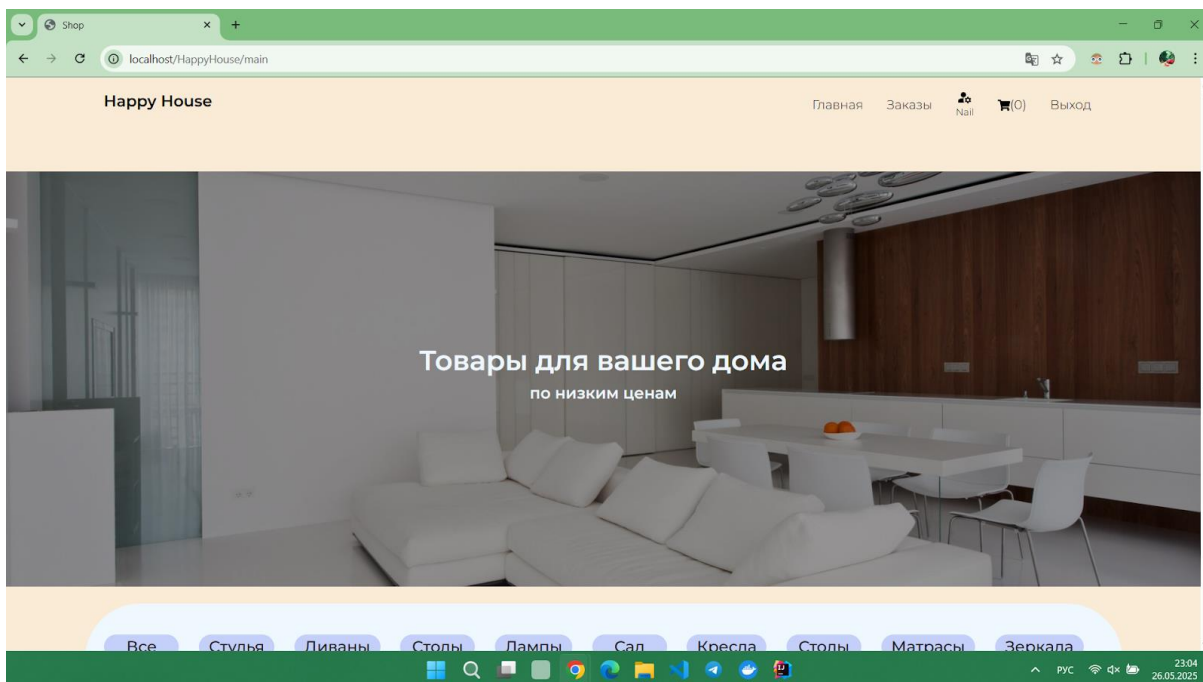


Рисунок 3 – Главная страница

2.1 Изменение корзины

Чтобы добавить товар в корзину пользователь должен нажать на синюю кнопку добавления внутри карточки товара, показанную на рисунке 4. После этого можно увидеть добавленную мебель на странице корзины (рисунок 5). Также в панели навигации обновится число товаров корзины.

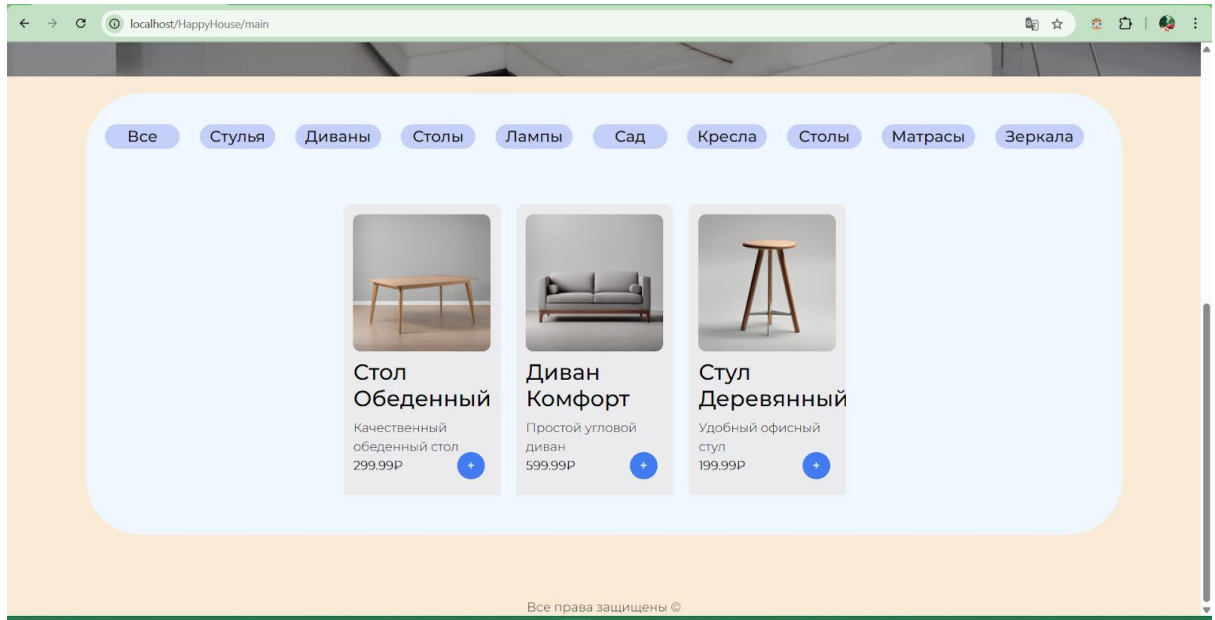


Рисунок 4 – Главная страница, товары

На странице корзины есть возможность изменить количество единиц товара или удалить товар из корзины вовсе.

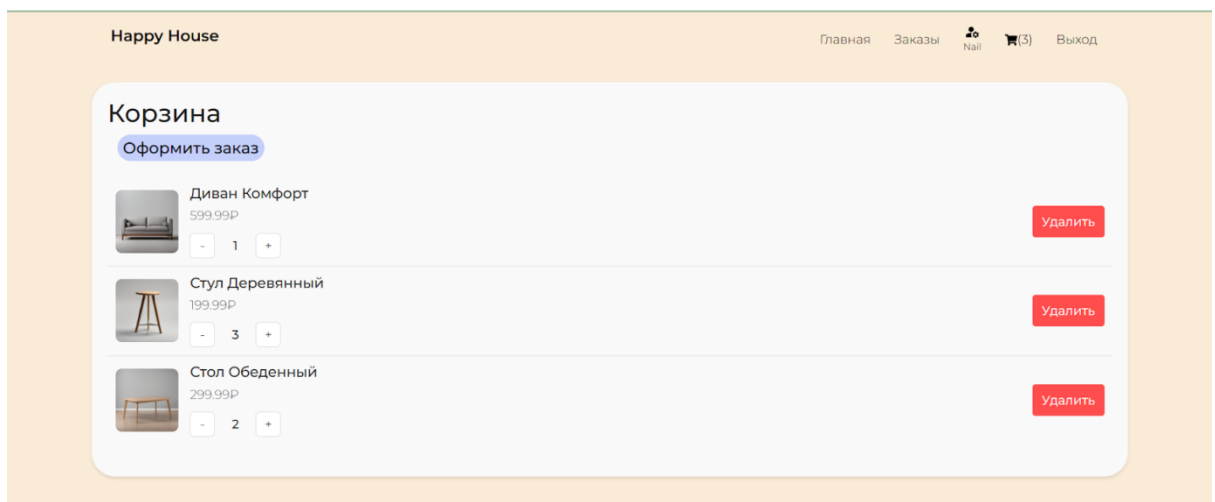


Рисунок 5 – Страница корзины

2.2 Изменение данных пользователя

На странице профиля есть возможность поменять имя, фамилию, адрес и номер телефона, а также пароль (все кроме почты). Для изменения пароля необходимо нажать на соответствующую кнопку, написать актуальный пароль и новый нужно написать 2 раза, при несоответствии паролей, выйдет ошибка.

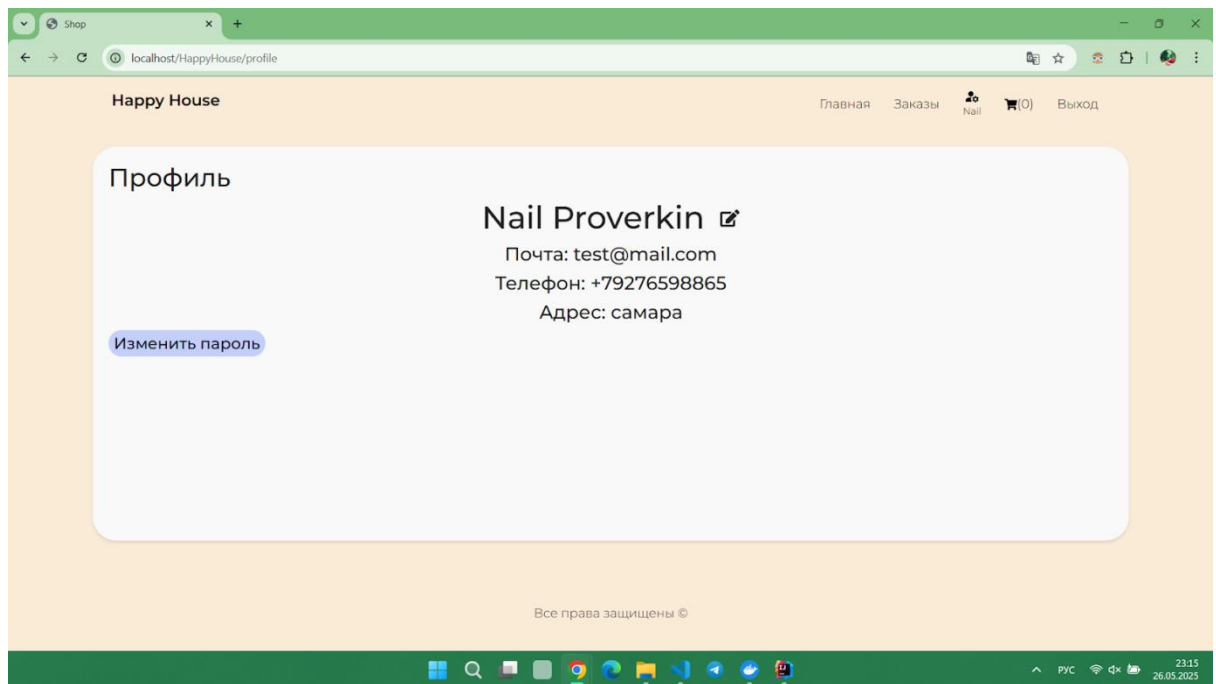


Рисунок 6 – Профиль пользователя

Для изменения оставшихся данных надо нажать на иконку ручки, рядом с фамилией пользователя, как на рисунке 6.

После сохранения данных выходит сообщение об успешности изменений. Поле можно увидеть на рисунке 7.

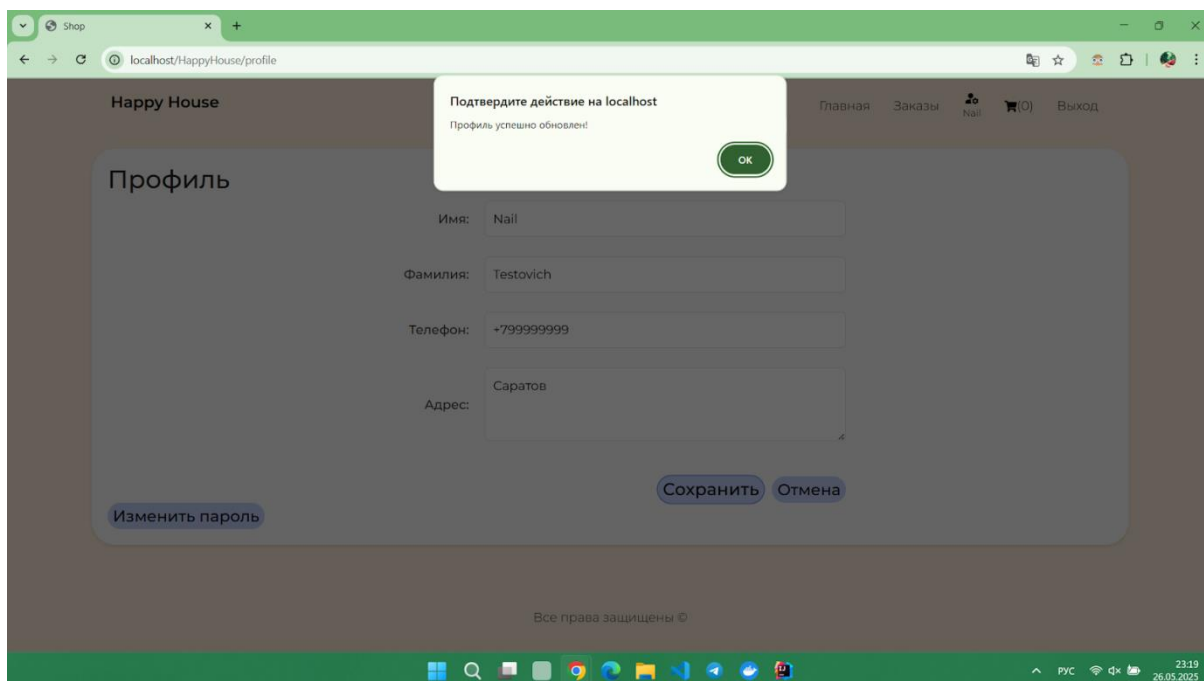


Рисунок 7 – Успешное изменение данных пользователя

2.3 Страница товара и страница заказов

При нажатии на товар на странице корзины или на главной, появляется страница товара, изображенная на рисунке 6, на которой можно более детально рассмотреть изображение или также добавить или удалить товар. При изменении статуса товара меняется стиль кнопки добавления а также сам текст внутри нее.

Находясь на странице корзины пользователь имеет возможность оформить заказ. После чего он добавиться в историю заказов, а корзина очистится. Заказы можно проверить на соответствующей странице. На рисунке 9 можно увидеть данную страницу.

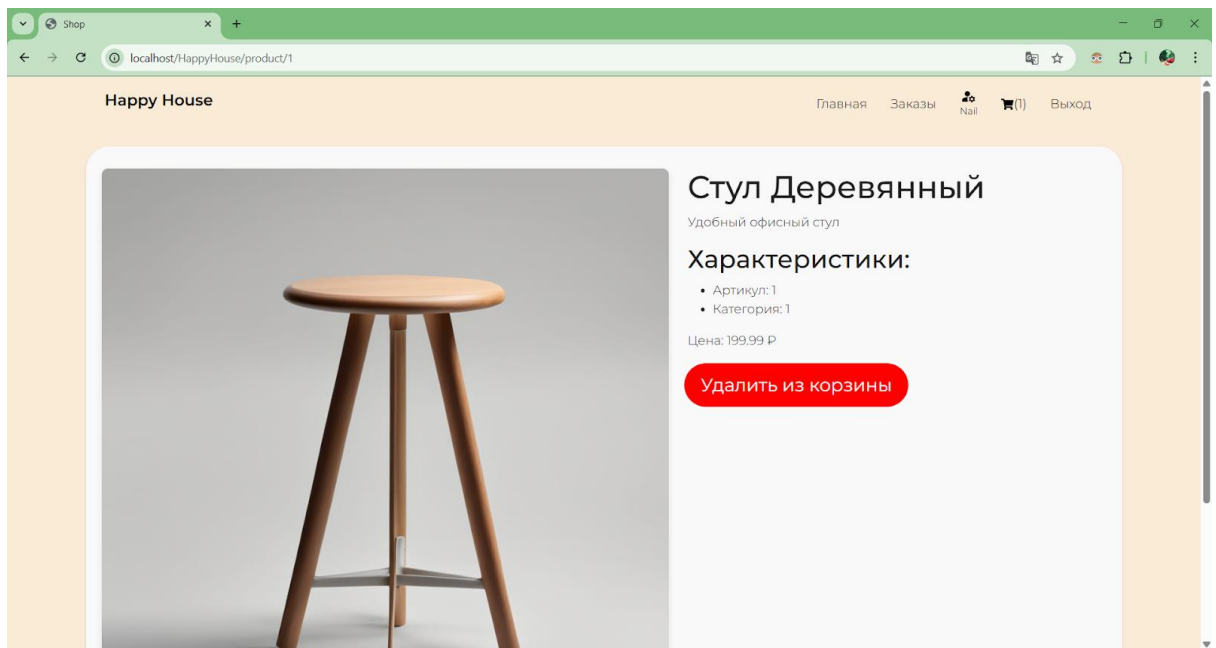


Рисунок 8 - Страница товара

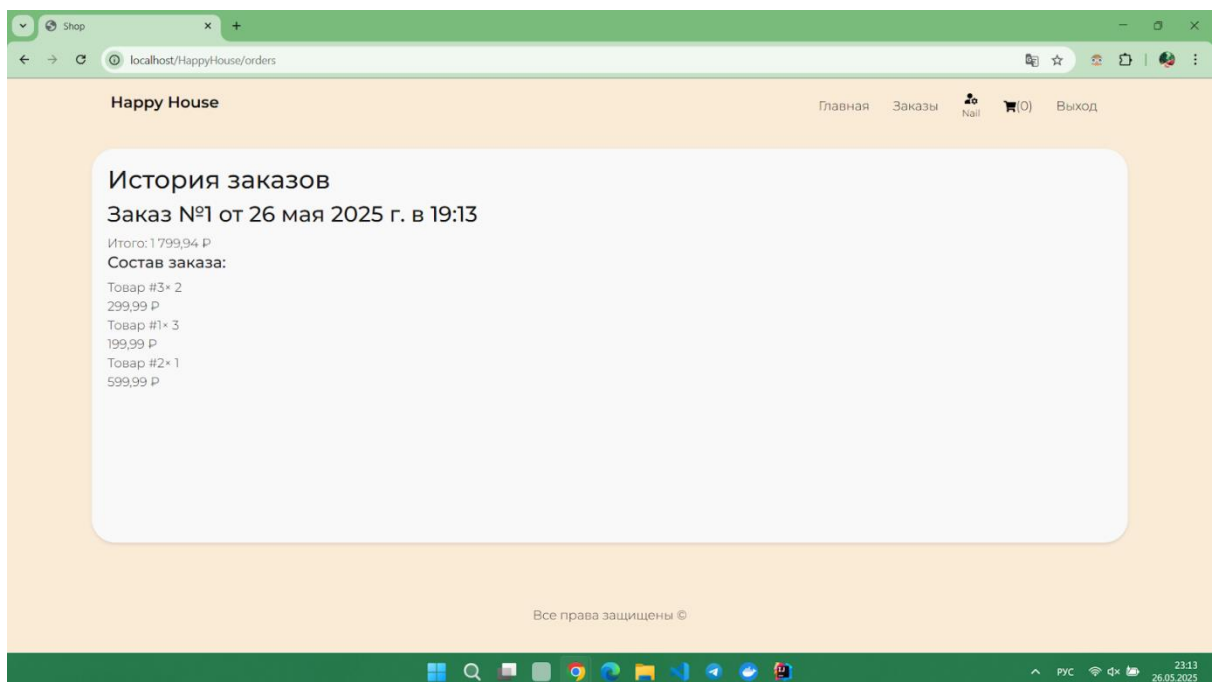


Рисунок 9 – Страница заказов

2.4 Выход из профиля

При нажатии на "Выход", стираются все данные пользователя (срабатывает функция, стирающая данные из localStorage: userId, access/refresh токены) и перед ним открывается страница входа. Если человек захочет переместиться на другую страницу приложения через адресную строку, его обратно перебросит на "вход".

ЗАКЛЮЧЕНИЕ

В ходе работы были освоены ключевые технологии: Spring Security, JWT, React Router, Zustand, ORM. Стала более понятна работа с REST API, контейнеризацией и работа с такими инструментами как Postman, DBeaver.

Были выполнены все основные требования к курсу:

1. Проектирование приложения.
2. Разработка базы данных.
3. Разработка API.
4. Авторизация.
5. Разработка клиентской части
6. Финализация приложения и упаковка в Docker

Что можно улучшить:

1. Добавить тесты (JUnit, Jest).
2. Реализовать ролевую модель (админ/пользователь).