

Министерство науки и высшего образования Российской Федерации  
Федеральное Государственное бюджетное образовательное учреждение высшего образования  
«Московский авиационный институт» (национальный исследовательский университет)

Учебный курс:  
**Python: Продвинутый уровень**

**Разработка онлайн магазина товаров  
с предиктивной аналитикой**

Работу выполнили:  
студенты уч. группы М8О-109СВ-24  
- Сорокин Никита  
- Ефанов Кирилл  
- Панкин Андрей  
- Бутаков Георгий  
- Дрёмов Александр

Преподаватель:  
Ахмед Самир Халид

*Москва, 2025*

## **Описание задачи:**

Онлайн-магазины все любят. Для того чтобы там покупать, никуда ходить не надо – лишь бы деньги и желание купить было.

## **Продавцу хочется:**

- знать продуктовую воронку,
- понимать, насколько перспективен продукт,
- прогнозировать, за сколько его продавать,
- рассчитывать, какой доход он сможет получить.

## **Ваша задача:**

Разработать систему онлайн-магазина с предиктивным функционалом, продуктовыми корзинами и рекомендациями для пользователей. Также предоставьте пользователям возможность:

- покупать продукцию,
- получать чеки.

## **Дополнительно:**

- Подумайте, какие признаки поспособствуют лучшей предиктивной способности вашей модели.
- Реализуйте процесс сбора данных.
- Оптимизируйте модель для наилучших результатов.

## Стек технологий:

- **Vue.js** обеспечивает создание динамичного и удобного интерфейса для пользователей;
- **FastAPI** позволяет быстро разрабатывать бэкенд с поддержкой аналитических функций и рекомендаций;
- **PostgreSQL** обеспечивает надежное хранение данных с возможностью гибкой аналитики;
- **Docker** и **docker compose** обеспечивают изоляцию, универсальность и удобство развертывания всех компонентов приложения.

За основу была взята статья [«Developing a Single Page App with FastAPI and Vue.js»](#) с сайта TestDriven.io, которая представляет пошаговое руководство по созданию приложения с использованием FastAPI для бэкенда и Vue.js для фронтенда.

В процессе разработки автор демонстрирует, как настроить окружение, создать RESTful API с FastAPI, разработать интерфейс с Vue.js, реализовать аутентификацию на основе токенов и объединить фронтенд с бэкендом.

Дополнительно рассматривается контейнеризация приложения с помощью Docker и использование PostgreSQL в качестве базы данных.

## Структура проекта

### **assets/**

Содержит статические ресурсы, такие как GIF-анимации, демонстрирующие различные аспекты приложения (админ, рекомендации, пользователь).

### **services/**

Разделен на два основных сервиса:

#### **backend/**

Содержит серверную часть проекта, включая Dockerfile, зависимости Python, исходный код с модулями для маршрутов, работы с базой данных и утилит.

#### **frontend/**

Включает клиентскую часть проекта, с Dockerfile, конфигурациями для Vue.js, зависимостями, исходным кодом компонентов, маршрутизации и представлений.

### **docker\_compose.yml**

Конфигурационный файл Docker Compose для запуска всех сервисов проекта в контейнерах.

## **backend/**

Содержит серверную часть проекта:

**Dockerfile** — Описание Docker-образа для бэкенда.

**postgres\_data/** — Данные PostgreSQL.

**requirements.txt** — Список Python-зависимостей.

**src/** — Исходный код:

**main.py** — Основной файл для запуска сервера.

**routes/** — Маршруты API:

**auth.py** — Авторизация.

**cart.py** — Работа с корзиной.

**products.py** — Управление продуктами.

**recommendations.py** — Рекомендации.

**user.py** — Пользователи.

**database/** — Работа с базой данных:

**db\_init.py, db\_models.py, db\_startup.py** — Инициализация, модели и запуск базы данных.

**csvs/, test\_csvs/** — CSV-файлы для работы с базой данных.

**utils/** — Утилиты:

**data\_processing.py, features.py, models.py** — Обработка данных и вспомогательные модули.

## **frontend/**

Содержит клиентскую часть проекта:

**Dockerfile** — Описание Docker-образа для фронтенда.

**babel.config.js, jsconfig.json, package.json** — Конфигурации и зависимости Vue.js.

**public/index.html** — Основной HTML-шаблон.

**src/** — Исходный код:

**App.vue** — Основной компонент приложения.

**components/** — Дополнительные компоненты, например NavBarComp.vue.

**router/** — Настройка маршрутов.

**store/** — Управление состоянием.

**views/** — Различные страницы приложения.

**assets/** — Логотипы.

## Датасет

**Instacart Market Basket Analysis** — это популярный набор данных, предоставленный платформой Instacart, которая специализируется на онлайн-продаже продуктов питания в США и Канаде. Этот набор данных широко используется для исследований в области анализа корзины покупок, построения рекомендательных систем и предсказания потребительского поведения.

Основные характеристики набора данных:

- **Заказы (Orders):** информация о заказах пользователей, включая время заказа, идентификаторы пользователей и сессий;
- **Пользователи (Users):** детали о пользователях, такие как демографические данные;
- **Продукты (Products):** информация о продуктах, включая названия, идентификаторы, категории и подразделы;
- **Категории (Departments и Aisles):** структура категорий продуктов, помогающая в классификации и анализе;
- **История заказов (Order Products):** детали о продуктах, включенных в каждый заказ, включая метки, указывающие, были ли они повторными покупками.

Применение: **рекомендательные системы, предсказание поведения, анализ корзины, кластеризация пользователей.**



## Предиктивная модель

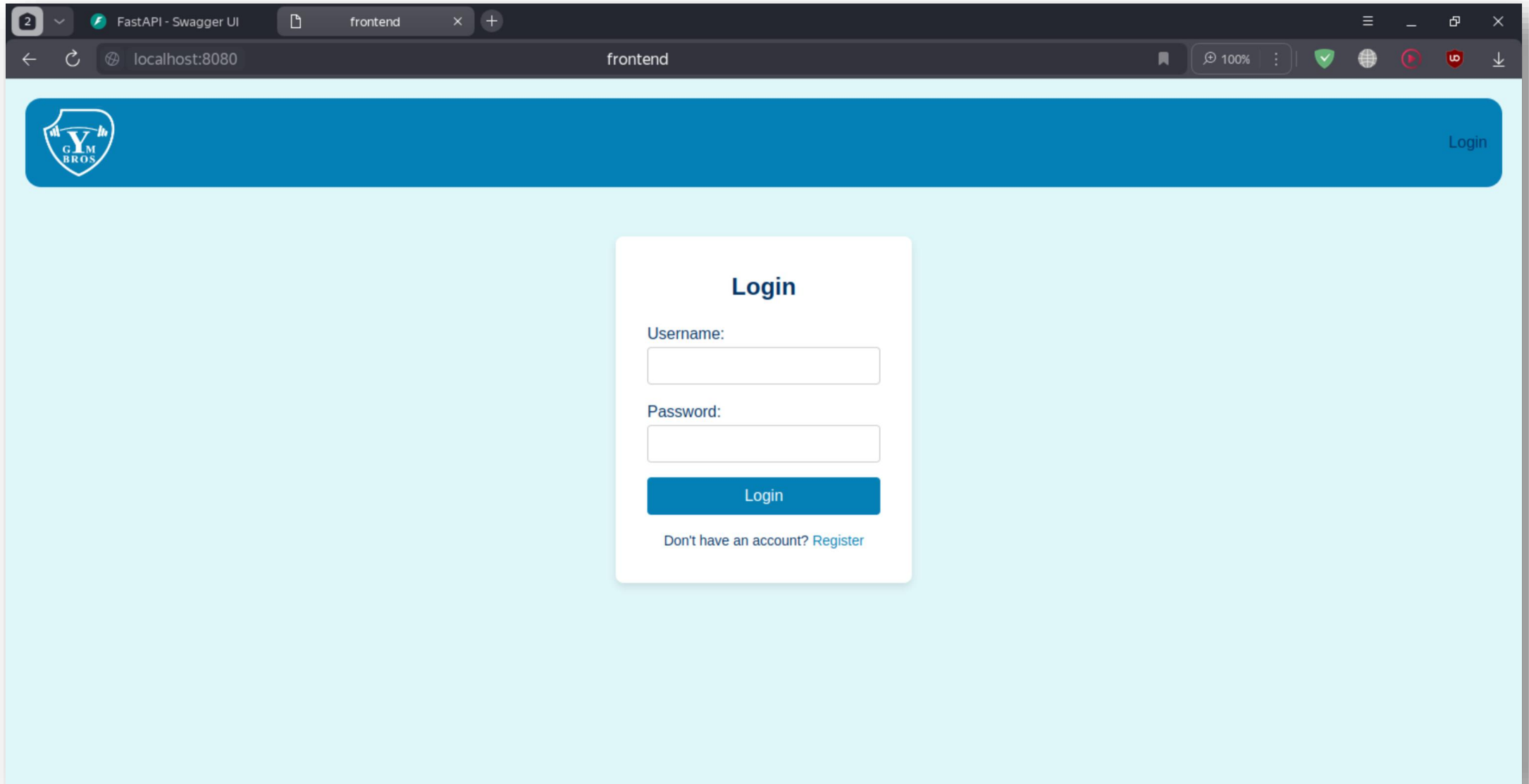
Была реализована рекомендательная система, которая предсказывает вероятность заказа конкретных продуктов пользователями.

Сначала загружаются исторические данные о заказах, включая предыдущие заказы пользователей, тренировочные данные, информацию о самих заказах и продуктах. Затем проводится обширная инженерия признаков: создаются характеристики продуктов (например, общее количество заказов и коэффициент повторных покупок), пользовательские признаки (такие как среднее время между заказами, общее количество заказов и уникальных товаров) и признаки взаимодействия пользователь-продукт (включая количество заказов конкретного продукта пользователем и последний заказ, в котором этот продукт был включён).

LightGBM обучается на этом наборе данных для моделирования вероятности заказа продукта пользователем. Для генерации рекомендаций модель предсказывает вероятности для всех возможных пар заказ-продукт в тестовой выборке и выбирает топ-10 продуктов с наивысшими предсказанными вероятностями для каждого пользователя.

Подробнее: <https://github.com/GymBros-Marketplace/service-recommendation/tree/master/src>

# Login page



The screenshot shows a web browser window with the address bar at `localhost:8080` and the page title `frontend`. The browser tabs include `FastAPI - Swagger UI` and `frontend`. The page features a blue header with the **GYM BROS** logo on the left and a **Login** link on the right. The main content area is light blue and contains a white login form centered on the page. The form has the title **Login**, followed by input fields for **Username:** and **Password:**, a blue **Login** button, and a link for [Register](#) with the text "Don't have an account?".

**GYM BROS**

Login

## Login

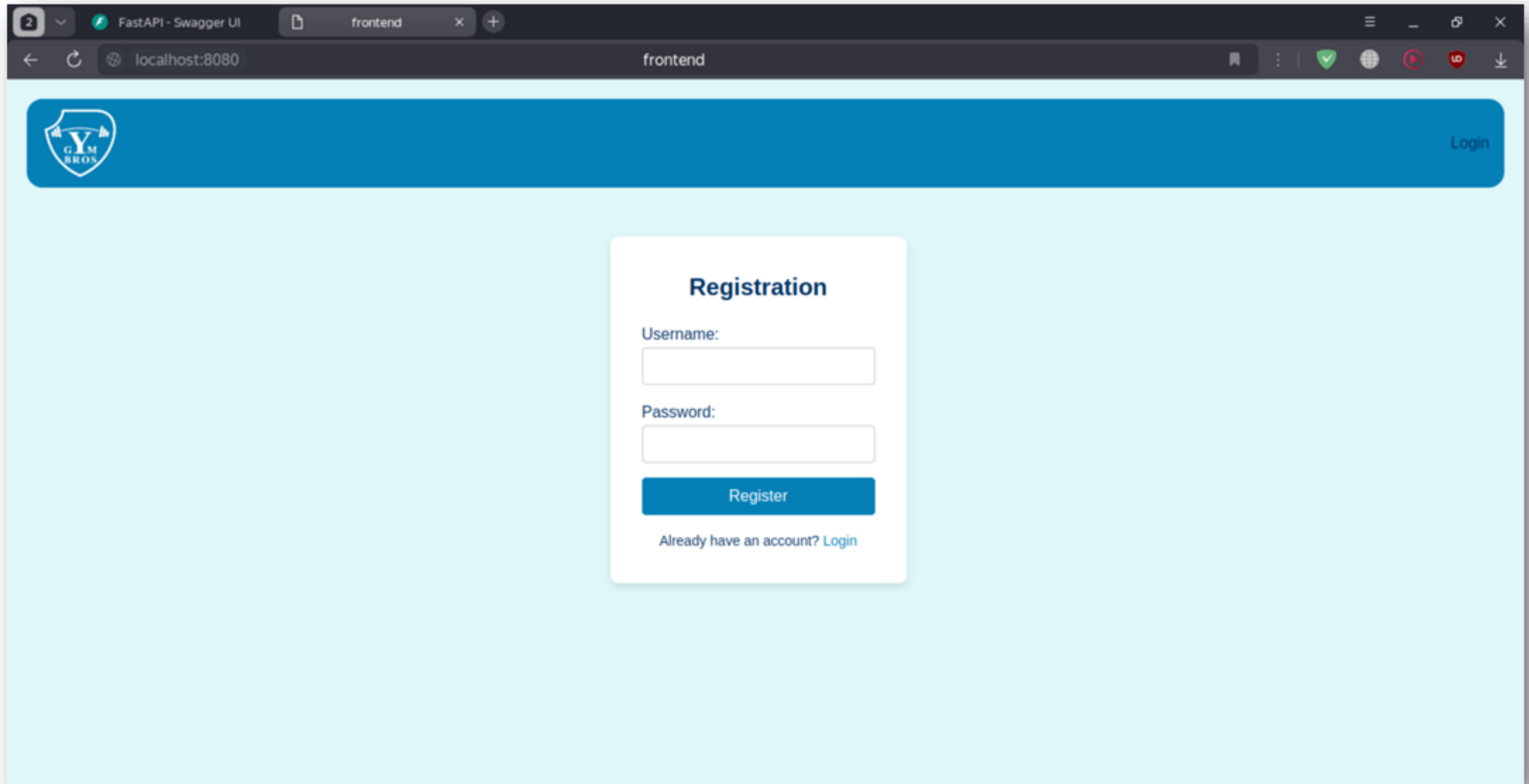
Username:

Password:

Login

Don't have an account? [Register](#)


## Registration page



The screenshot shows a web browser window with the address bar displaying 'localhost:8080' and the page title 'frontend'. The browser tabs include 'FastAPI - Swagger UI' and 'frontend'. The page features a blue header with the 'GYM BROS' logo on the left and a 'Login' link on the right. The main content area is light blue and contains a white registration form. The form has a title 'Registration', followed by 'Username:' and a text input field, then 'Password:' and another text input field. Below these is a blue 'Register' button. At the bottom of the form, it says 'Already have an account? [Login](#)'.

2 FastAPI - Swagger UI frontend

localhost:8080 frontend

 Login

### Registration

Username:

Password:

Register

Already have an account? [Login](#)

## Add/edit/delete product page

2

FastAPI - Swagger UI

frontend

localhost:8080

frontend

Logout

### Product List

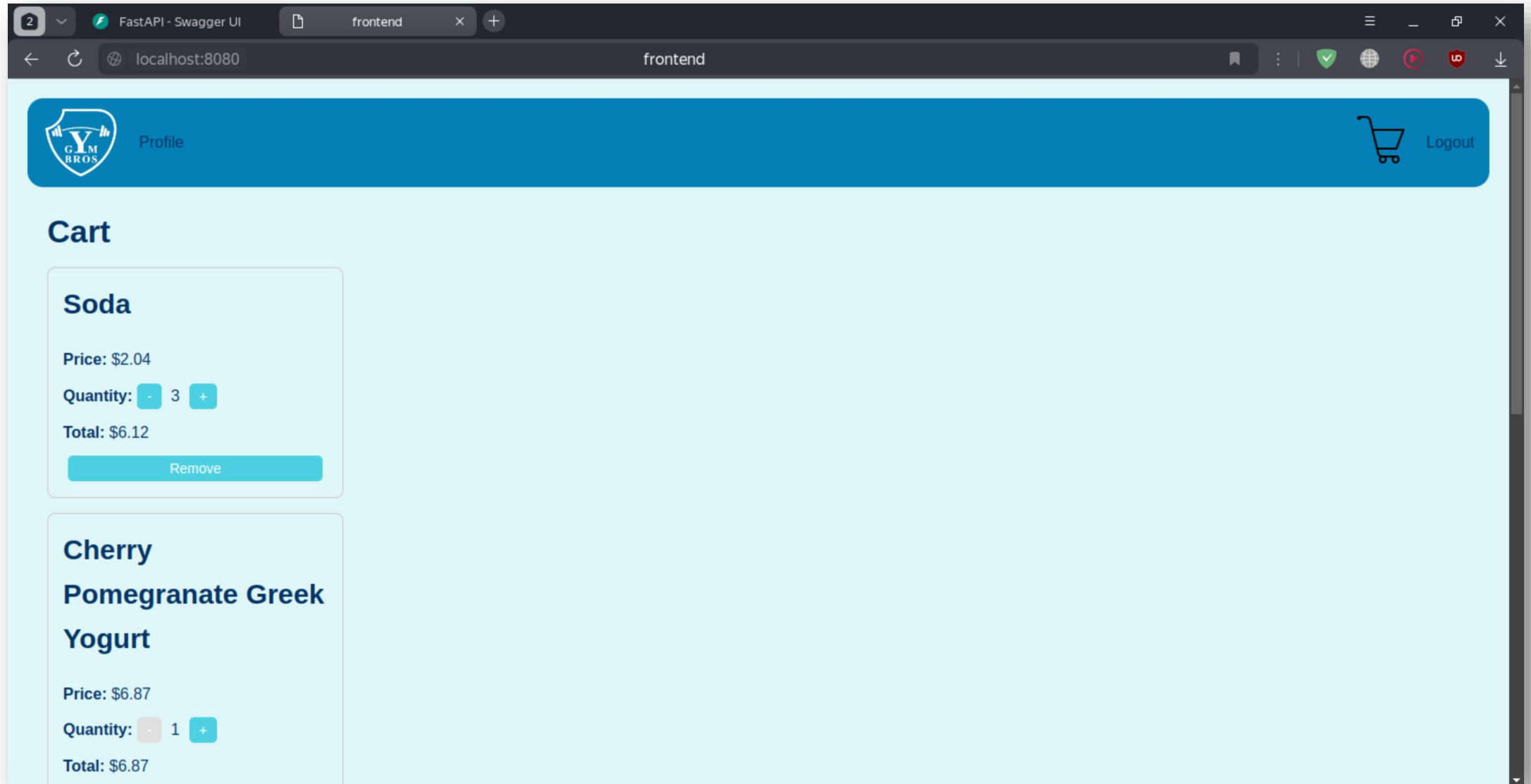
Name	Price	Department	Aisle	Actions
Soda	2.04	7	77	<a href="#">Edit</a> <a href="#">Delete</a>
Dried Sweetened Cranberries	7.24	19	117	<a href="#">Edit</a> <a href="#">Delete</a>
Vanilla Bean Natural Lowfat Yogurt	7.71	16	120	<a href="#">Edit</a> <a href="#">Delete</a>
Extra Soft Double Roll Bathroom Tissue	4.94	17	54	<a href="#">Edit</a> <a href="#">Delete</a>
Cherry Pomegranate Greek Yogurt	6.87	16	120	<a href="#">Edit</a> <a href="#">Delete</a>
Sustainably Soft Bath Tissue	2.77	17	54	<a href="#">Edit</a> <a href="#">Delete</a>
Organic Green Onions	7.8	4	83	<a href="#">Edit</a> <a href="#">Delete</a>
Pure Coconut Water	5.51	7	98	<a href="#">Edit</a> <a href="#">Delete</a>
Organic Lightly Salted Sea Salt Thin & Crispy Restaurant Style Tortilla Chips	1.34	19	107	<a href="#">Edit</a> <a href="#">Delete</a>
Olive Oil Cooking Spray	1.07	13	19	<a href="#">Edit</a> <a href="#">Delete</a>
Natural Sharp Cheddar Sliced Cheese	5.44	16	21	<a href="#">Edit</a> <a href="#">Delete</a>

## Products page

The screenshot shows a web browser window with the URL 'localhost:8080' and the page title 'frontend'. The browser tabs include 'FastAPI - Swagger UI' and 'frontend'. The page features a blue header with the 'GYM BROS' logo, a 'Profile' link, a search bar with the placeholder 'Search products...', a shopping cart icon, and a 'Logout' link. The main content area displays a grid of nine product cards, each with a product name, price, and an action button.

Product Name	Price	Action
Soda	\$2.04	Go to Cart
Dried Sweetened Cranberries	\$7.24	Add to Cart
Vanilla Bean Natural Lowfat Yogurt	\$7.71	Add to Cart
Extra Soft Double Roll Bathroom Tissue	\$4.94	Add to Cart
Cherry Pomegranate Greek Yogurt	\$6.87	Go to Cart
Sustainably Soft Bath Tissue	\$2.77	Add to Cart
Organic Green Onions	\$7.80	Go to Cart
Pure Coconut Water	\$5.51	Go to Cart
Organic Lightly Salted Sea Salt Thin & Crispy Restaurant Style Tortilla		

# Cart page



# Receipt

3

FastAPI - Swagger UI

frontend

91263388-cf9d-43fb-94 X

+

←

↺

blob:http://localhost:8080/91263388-cf9d-43fb-94be-9296799edc9b

🔖

↓ save

🖨 print

⋮

🛡

🌐

🔴

🔴

↓

☰

91263388-cf9d-43fb-94be-9296799edc9b

1 / 1

−

100%

+


🔍

🔄

📄

🖨

⋮



1

### Receipt for qwe

Date: 2025-01-09 08:14:03

Product Name	Quantity	Price	Total
Soda	3	\$2.04	\$6.12
Cherry Pomegranate Greek Yogurt	1	\$6.87	\$6.87
Organic Green Onions	2	\$7.80	\$15.60
Pure Coconut Water	3	\$5.51	\$16.53
Total Price			\$45.12

## Recommendations

http://localhost:8000/recommendations/user\_12

Server response

Code	Details
200	<p>Response body</p> <pre>{   "user_id": 12,   "recommended_products": [     {       "product_id": 13176,       "name": "Bag of Organic Bananas"     },     {       "product_id": 21616,       "name": "Organic Baby Arugula"     },     {       "product_id": 6182,       "name": "Natural Artesian Bottled Water"     },     {       "product_id": 33787,       "name": "Total 2% Lowfat Greek Strained Yogurt with Peach"     },     {       "product_id": 18534,       "name": "Grade A Extra Large Eggs"     },     {       "product_id": 45446,       "name": "Yogurt, Organic, Lowfat, Strawberry"     },     { </pre>



Над проектом работали студенты учебной группы М8О-109СВ-24:

## **ML**

- [Сорокин Никита](#)
  - Разработка модели машинного обучения;
  - Интеграция ML в backend.

## **Backend**

- [Ефанов Кирилл](#)
  - Аутентификация и управление пользователями;
  - Работа с базой данных.
- [Панкин Андрей](#)
  - Управление корзиной и продуктами;
  - Развертывание базы данных;
  - Интеграция ML в backend.

## **Frontend**

- [Бутаков Георгий](#)
  - Разработка UI-компонентов;
  - Настройка маршрутизации.
- [Дрёмов Александр](#)
  - Разработка UI-компонентов;
  - Управление состоянием приложения;
  - Интеграция с backend API.

## Список источников

<https://testdriven.io/blog/developing-a-single-page-app-with-fastapi-and-vuejs/>

Этот источник является полезным руководством для разработки приложений с использованием FastAPI и Vue.js. В статье пошагово описывается создание серверной части на FastAPI и клиентской на Vue.js, включая интеграцию между ними.

<https://dev.to/glennvirox/how-to-connect-a-fastapi-server-to-postgresql-and-deploy-on-gcp-cloud-run-4l5h>

Статья на dev.to посвящена соединению FastAPI с базой данных PostgreSQL и разворачиванию приложения на Google Cloud Run.

<https://fastapi.tiangolo.com/>

Официальная документация FastAPI.

<https://vuejs.org/guide/introduction.html>

Официальное руководство по Vue.js.

<https://zenodo.org/records/2654444>

Этот источник представляет собой научную публикацию, в которой проводится EDA для датасета Instacart Market Basket Analysis