Использовано Backend: Flask, SQLAlchemy, Marshmallow Docker Frontend: React, Material-UI Docker

Запуск docker-compose up --build

Backend API: http://localhost:5000 Frontend: http://localhost:3000

backend/requirements.txt - зависимости для backend

Во фронте использован fetch для получения списка фильмов в frontend/src/list/components/BuildingsGrid.tsx

https://drive.google.com/file/d/1aYzovagIY4-9-X7BjhykVV1Rz9Au9lpP/view?usp=sharing DEMO

Сама fetch функция:

useEffect(() => {

    fetch("http://127.0.0.1:5000/api/v1/movies")

      .then((res) => res.json())

      .then((data) => {

        const movies = data.movies;

        const transformed = movies.map((b: Movie) => ({

          ...b,

          id: b.film\_id,

          genre\_name: b.genre.genre\_name,

          language\_name: b.language.language\_name,

        }));

        setFilms(transformed);

      })

      .catch((err) => {

        console.error("Ошибка при получении фильмов:", err);

      });

  }, []);

По <http://127.0.0.1:5000/api/v1/movies> выдется json с информацией по фильмам.

Файл Dockerfile.backend

FROM python:3.11-slim

WORKDIR /app

COPY backend/requirements.txt .

RUN pip install --no-cache-dir -r requirements.txt

COPY backend/ .

CMD ["python", "app.py"]

Файл Dockerfile.frontend

FROM node:20-alpine

WORKDIR /app

COPY frontend/package\*.json ./

RUN npm install

COPY frontend/ .

CMD ["npm","start"]

Docker-compose.yml

version: '3.8'

services:

  backend:

    build:

      context: .

      dockerfile: Dockerfile.backend

    ports:

      - "5000:5000"

    volumes:

      - ./backend:/app

    environment:

      - FLASK\_ENV=development

  frontend:

    build:

      context: .

      dockerfile: Dockerfile.frontend

    ports:

      - "3000:3000"

    volumes:

      - ./frontend:/app

    depends\_on:

      - backend

В package.json добавил "proxy": "http://localhost:5000",

Также файл зависимостей backend (requirements.txt)

blinker==1.9.0

click==8.1.8

colorama==0.4.6

Flask==3.1.0

flask-marshmallow==1.3.0

Flask-SQLAlchemy==3.1.1

greenlet==3.2.1

itsdangerous==2.2.0

Jinja2==3.1.6

MarkupSafe==3.0.2

marshmallow==4.0.0

marshmallow-sqlalchemy==1.4.2

numpy==2.2.5

pandas==2.2.3

psycopg2-binary==2.9.10

python-dateutil==2.9.0.post0

python-dotenv==1.1.0

pytz==2025.2

six==1.17.0

SQLAlchemy==2.0.40

typing\_extensions==4.13.2

tzdata==2025.2

Werkzeug==3.1.3

flask-cors~=5.0.1

flask-restful==0.3.10