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 и кибернетики

Кафедра технической кибернетики

Отчет по лабораторной работе №0

Дисциплина: «Основы сетевого программирования»

Тема «Проектирование приложения»

Выполнили:

Воронин Павел  
Максимович,

Фомин Максим Андреевич.

Группа: 6304-010302D

Самара 2025

Основные задачи

1. **Проектирование архитектуры**:
   * Разработать схему взаимодействия компонентов (сервер, БД, клиент);
   * Разработать логическую схему базы данных (использование, как минимум, 3-х сущностей и 1-ой связи many-to-many);
   * Определить структуру API (REST, методы, URL, параметры, форматы запросов и ответов).
2. **Создание Git-репозитория**:
   * Инициализировать репозиторий (рекомендуется использовать GitHub);
   * Настроить базовую структуру проекта.
   * Добавить .gitignore для исключения ненужных файлов (виртуальное окружение, логи, артефакты сборки и т. д.).
3. **Подготовка окружения**:
   * Установить Java/Python;
   * Настроить виртуальное окружение и зависимости.
4. **Оформление документации**:
   * Подготовить отчет с результатами работы.

Приложение “Art Gallery” / “Галерея художников и картин”

**Концепт:**Приложение управляет базой картин и художников. Один художник может участвовать в создании разных картин. Авторы могут создавать учётные записи, грузить фотографии своих работ и описывать истории создания.

**Архитектура**

**1. Бэкенд**

• Язык: Python (VSC).

• База данных: PostgreSQL.

**2. Фронтенд**

• Компоненты: список картин, профиль художника, форма добавления новых картин.

**3. API**

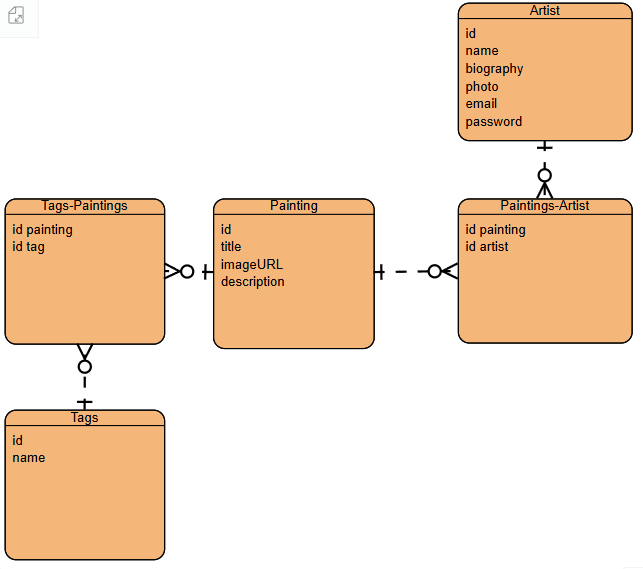
**Описание эндпоинтов**:

**Аутентификация**:

* + - **POST /auth/register**: Регистрация нового пользователя.
    - **POST /auth/login**: Вход пользователя в систему с использованием JWT для аутентификации.

**Управление картинами**:

* + - **GET /paintings**: Получение списка всех картин.
    - **GET /paintings?artist\_id=:artist\_id**: Получение списка картин одного автора.
    - **GET /paintings?tag\_id=:tag\_id**: Получение списка картин по тегу.
    - **POST /paintings**: Добавление новой картины.
    - **POST /paintings/:id/tags/:id**: Добавление тега картине
    - **DELETE /paintings**: Удаление картины.
    - **DELETE /paintings/:id/tags/:id**: Удаление тега картины.
    - **Управление художниками**:
    - **GET /artists**: Получение списка всех художников.
    - **POST /artists**: Добавление нового художника.
    - **DELETE / artists**: Удаление художника.

**4. ER-Диаграмма   
**

**Стек технологий**

* **Backend**: Python, Flask/Django?, PostgreSQL.
* **Frontend**: (если применимо) React/Angular/Vue.js.
* **Аутентификация**: JWT?
* **Дополнительные инструменты**: Docker для контейнеризации?
* **GitHub**: <https://github.com/PeVANoD/VoroninFominTCP>

Итог.

В ходе этой лабораторной, мы провели небольшое исследование доступных инструментов для разработки веб приложения, использующего базы данных, составили примерную архитектуру и настроили некоторые стартовые инструменты и окружения.

Возможно, в ходе работы будут пересмотрены некоторые элементы проекта.