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 №1

Дисциплина: «Основы сетевого программирования»

Тема «Разработка базы данных»

Выполнили:

Воронин Павел  
Максимович,

Фомин Максим Андреевич.

Группа: 6304-010302D

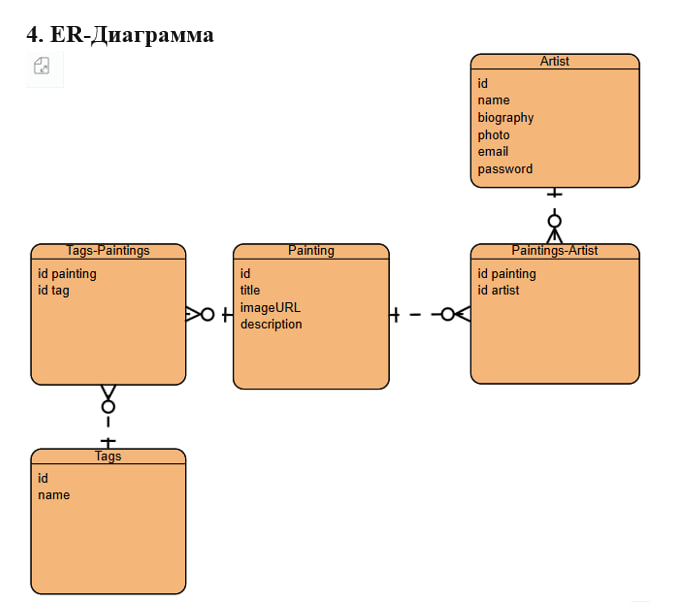
Самара 2025

Основные задачи

1. **Настройка базы данных:**
   * Развернуть PostgreSQL в Docker;
   * Подключиться к БД через DBeaver;
2. **Разработка ORM-моделей**:
   * Определить сущности и связи между ними;
   * Реализовать модели с использованием SQLAlchemy/Django ORM.
   * Настроить миграции.
3. **Разработка модели пользователя**:
   * Определить структуру модели (email, пароль, имя и т. д.);
   * Настроить хеширование паролей.
4. **Функционал для работы с моделями**:
   * Написать скрипты для заполнения базы данных тестовыми данными.
   * Реализация функционала для работы с данными в соответствии с тематикой выбранного приложения.
5. **Оформление отчета**
   * Описать структуру базы данных
   * Подготовить скрипты создания таблиц и миграции
   * Создать презентацию с результатами работы

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

1. **Разработка ORM-моделей**

Согласно нашей ER-Диаграмме разработаем модели 

Из ER-диаграммы видны следующие сущности:

- Artist (Художник)

- Painting (Картина)

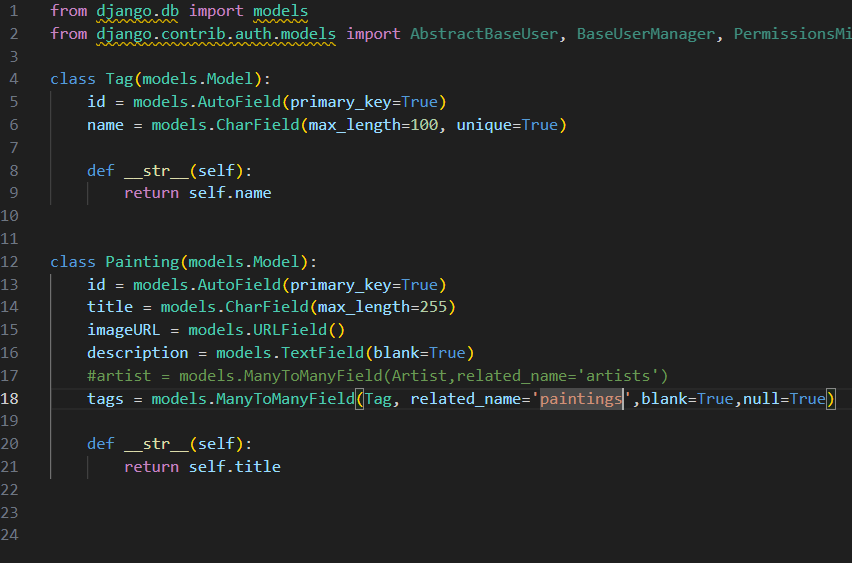
- Tags (Теги)

- Tags-Paintings (связь между тегами и картинами)

- Paintings-Artist (связь между картинами и художниками)

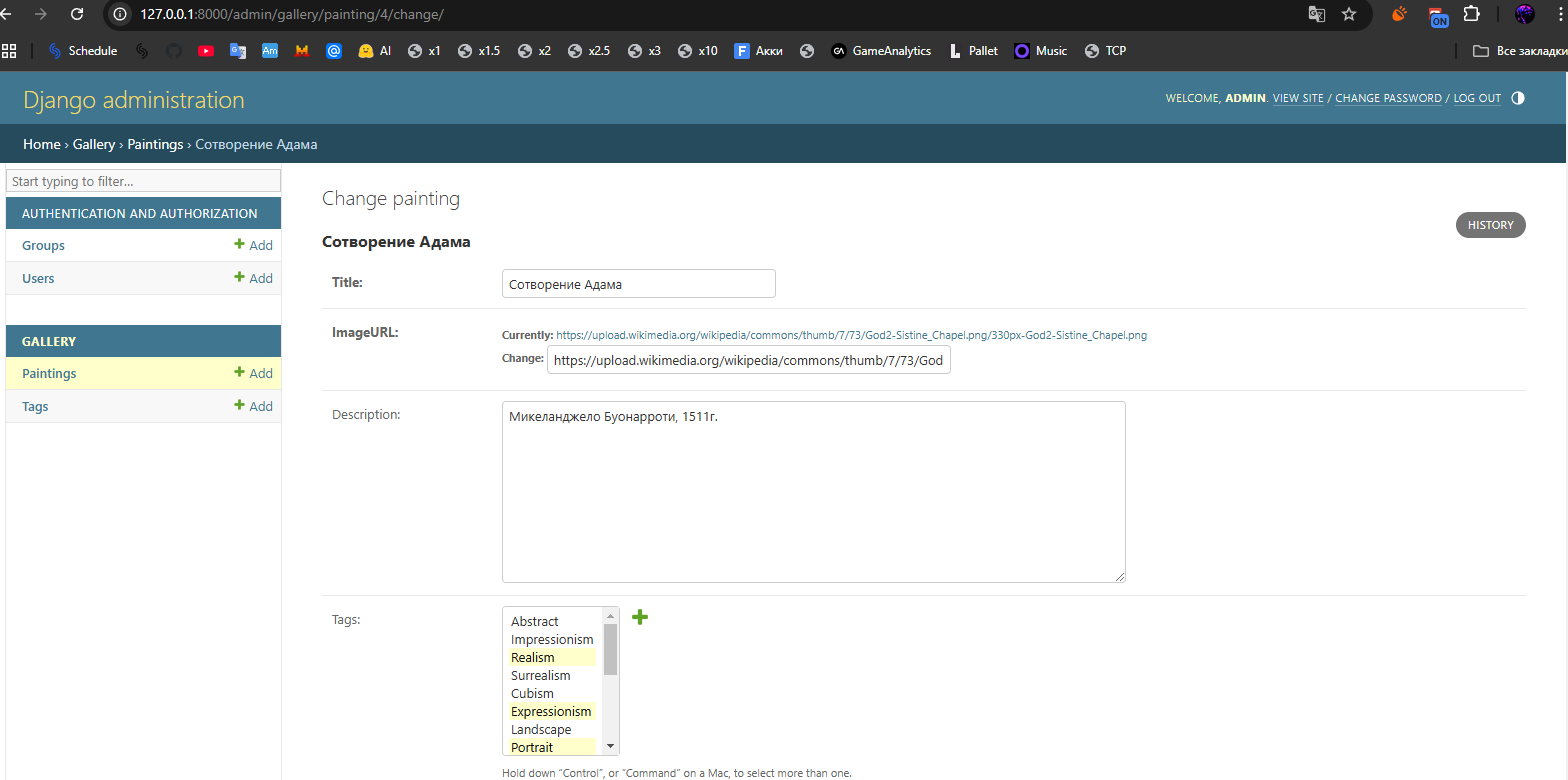
Для выполнения этого пункта в VSC терминале создаем новое приложение командой «python manage.py startapp gallery».

Создаем в папке проекта файл models.py и определяем внутри него наши сущности [1]. Создали Paintings и Tags.

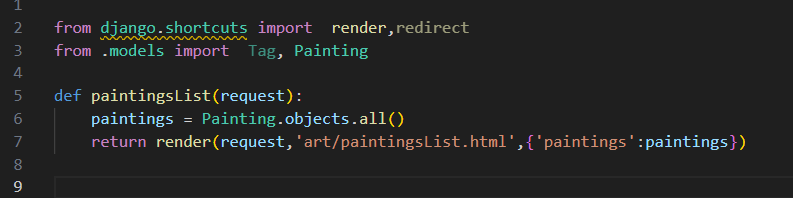


Так как мы используем Django, миграции были настроены при помощи терминальных команд python manage.py makemigrations и python manage.py migrate

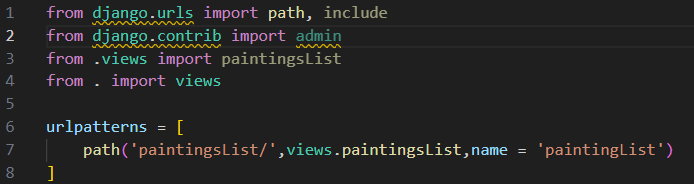
Далее попробовали заполнить базу данных.



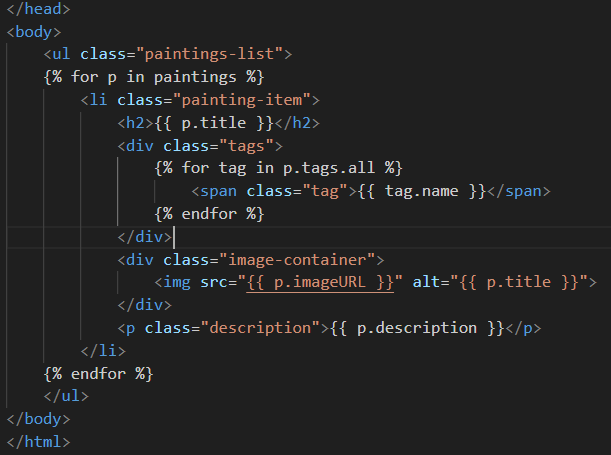
Мы можем получить данные из базы данных, а затем отобразить на странице.



Ссылка по которой мы сможем увидеть наш request:

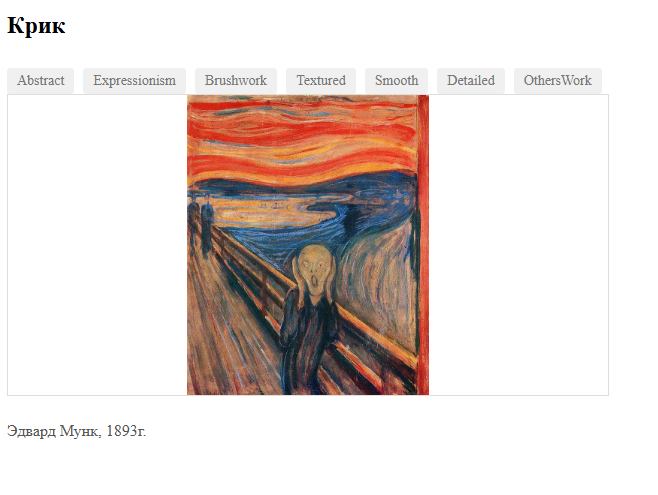


Вывод из request данных на страницу html:



* **Backend**: Python, Django, sqlite3.
* **GitHub**: <https://github.com/PeVANoD/VoroninFominTCP>

**Пример оформления картины:**



Итог.

В ходе этой лабораторной, мы настроили базу данных, заполнить ее и выполнить миграции, составили примерную архитектуру и настроили стартовые инструменты и окружения.

В процессе выполнения программ мы смогли настроить создание картин и тегов, а также их отображение. К картинам можно добавлять теги.