

Проект 1: "Система управления коворкингом (SmartSpace API)"

Этот проект сфокусирован на **CRUD-операциях**, работе с **датами** и сложной **бизнес-логике** (проверка доступности ресурса).

Суть: API для бронирования рабочих мест, переговорных комнат и оборудования в офисе.

Основные фичи:

- **Управление ресурсами:** Добавление комнат, столов, мониторов (с указанием характеристик).
- **Система бронирования:** Пользователь может забронировать объект на определенный слот времени.
 - **Сложность:** Реализовать проверку, чтобы нельзя было забронировать один и тот же стол дважды на одно время.
- **Профили пользователей:** Роли (Админ/Пользователь). Админ может подтверждать или отменять брони.
- **Фильтрация:** Поиск свободных мест на конкретную дату.

Что ребята закрепят:

- Связи One-to-Many и Many-to-Many в SQLAlchemy/SQLModel.
 - Валидацию дат через Pydantic.
 - Dependency Injection (для проверки прав доступа).
-

Проект 2: "Агрегатор курсов валют и криpto-портфель (CryptoTrack API)"

Проект с упором на **интеграцию с внешними API**, **фоновые задачи** и **безопасность**.

Суть: Сервис, который подтягивает актуальные курсы валют/криптовалют и позволяет пользователям вести учет своих активов.

Основные фичи:

- **Внешняя интеграция:** Написать клиент (через `httpx`), который раз в час запрашивает курсы с публичного API (например, CoinGecko или ExchangeRate-API).
- **Портфель:** Пользователь добавляет записи: "Купил 0.5 BTC по цене \$20,000". API должно считать текущую стоимость портфеля в реальном времени.

- **Watchlist:** Возможность подписаться на изменение цены монеты (уведомление в логах или отдельной таблице).
- **Аналитика:** Эндпоинт, который выдает процент прибыли/убытка по каждой позиции.

Что ребята закрепят:

- Асинхронные запросы ([httpx](#)).
- Работу с внешними данными и обработку ошибок (что если внешний API упал?).
- JWT-авторизацию (защита личного портфеля).
- Опционально: Background Tasks в FastAPI для обновления курсов.

Docker и Docker-Compose

Diagram Отношение Сущностей Use Case Sequence

Docker Hub и Github

У каждого должна быть своя ветка и минимум по 10 коммитов