Задание 1:
Спроектировать схему базы данных для выбранного проекта. Схема должна включать в себя модели с полями, а так же связи моделей. Должен быть указан тип поля и данные которые он будет в себе хранить.

Задание 2:
Сформировать список функциональности системы.

Задание 3:
Подготовить описание эндпоинтов. Описание должно включать URL, метод, краткое описание его назначения, список принимаемых аргументов с указанием обязательное ли это поле или нет, валидацию данных, информацию о том что будет происходить с этими данными.

Пример:
Проект: Управление задачами в офисе
Цель проекта: Разработать базу данных для системы управления задачами в офисе, которая позволит эффективно отслеживать задачи, назначать ответственных, контролировать выполнение и анализировать данные о процессах.
Часть 1: Проектирование схемы базы данных

1. **Модель "Пользователи"**
    - Поля:
        - ID (Уникальный идентификатор пользователя)
        - Имя (Текстовое поле для имени пользователя)
        - Email (Электронная почта пользователя)
        - Пароль (Хэшированный пароль пользователя)
    - Связи:
        - Один ко многим с моделью "Задачи" (один пользователь может быть ответственным за много задач)
2. **Модель "Задачи"**
    - Поля:
        - ID (Уникальный идентификатор задачи)
        - Название (Текстовое поле с описанием задачи)
        - Описание (Большое текстовое поле с подробным описанием)
        - Статус (Перечисляемый тип данных: Новая, В процессе, Завершена и т.д.)
        - Дата создания (Дата создания задачи)
        - Дедлайн (Дата и время, когда задачу нужно завершить)
    - Связи:
        - Многие ко многим с моделью "Пользователи" через промежуточную таблицу "Назначения задач"
3. **Модель "Назначения задач"** (Промежуточная таблица для связи многие ко многим)
    - Поля:
        - ID (Уникальный идентификатор записи)
        - ID_Пользователя (Внешний ключ, связанный с ID пользователя)
        - ID_Задачи (Внешний ключ, связанный с ID задачи)

Часть 2: Функциональность проекта
Необходимо разработать следующий список функциональности для системы управления задачами:

1. **Авторизация и аутентификация пользователя**:

- Пользователь должен иметь возможность зарегистрироваться или войти в систему с помощью своего email и пароля.

2. **Создание новых задач**:
   - Пользователь может создавать новые задачи, указывая их название, описание и дедлайн.
3. **Назначение задач**:
   - Пользователь должен иметь возможность назначить задачу другому пользователю из списка зарегистрированных пользователей.
4. **Отслеживание статуса задач**:
   - Пользователь должен видеть статус каждой задачи (новая, в процессе, завершена и т.д.).
5. **Фильтрация и поиск задач**:
   - Реализовать возможность фильтрации и поиска задач по различным параметрам, таким как статус, ответственный пользователь и дедлайн.
6. **Уведомления и напоминания**:
   - Система должна отправлять уведомления пользователям о назначенных задачах, приближающихся дедлайнах и изменениях статуса задач.
7. **Отчетность и аналитика**:
   - Реализовать возможность генерации отчетов о выполненных задачах, времени, затраченном на каждую задачу, и анализа производительности пользователей.
8. **Администрирование системы**:
   - Предусмотреть возможность администрирования системы, включая управление пользователями, ролей и доступом к данным.

Часть 3: Описание эндпоинтов

1. **Эндпоинт: /api/auth/login**
   - Метод: POST
   - Описание: Аутентификация пользователя.
   - Входные данные:
     - Email (строка, обязательное поле)
     - Пароль (строка, обязательное поле)
   - Преобразование данных: Проверка соответствия введенных данных с данными в базе данных.
   - Запись в таблицы: Нет.
   - Валидация данных: Проверка формата электронной почты и наличия пароля.
2. **Эндпоинт: /api/auth/register**
   - Метод: POST
   - Описание: Регистрация нового пользователя.
   - Входные данные:
     - Имя (строка, обязательное поле)
     - Email (строка, обязательное поле)
     - Пароль (строка, обязательное поле)
   - Преобразование данных: Создание новой записи в таблице "Пользователи" с указанными данными.
   - Запись в таблицы: Таблица "Пользователи".
   - Валидация данных: Проверка формата электронной почты, уникальности email и длины пароля.
3. **Эндпоинт: /api/tasks/create**
   - Метод: POST
   - Описание: Создание новой задачи.
   - Входные данные:
     - Название (строка, обязательное поле)

        - Описание (строка)
        - Дедлайн (дата и время, обязательное поле)
    - Преобразование данных: Создание новой записи в таблице "Задачи" с указанными данными.
    - Запись в таблицы: Таблица "Задачи".
    - Валидация данных: Проверка наличия названия и дедлайна.
4. **Эндпоинт: /api/tasks/{taskId}/assign**
    - Метод: POST
    - Описание: Назначение задачи определенному пользователю.
    - Входные данные:
        - ID пользователя (число, обязательное поле)
    - Преобразование данных: Создание новой записи в таблице "Назначения задач" с указанным ID задачи и ID пользователя.
    - Запись в таблицы: Таблица "Назначения задач".
    - Валидация данных: Проверка существования пользователя с указанным ID.
5. **Эндпоинт: /api/tasks/{taskId}/update**
    - Метод: PUT
    - Описание: Обновление информации о задаче.
    - Входные данные:
        - Название (строка, обязательное поле)
        - Описание (строка)
        - Статус (строка)
        - Дедлайн (дата и время, обязательное поле)
    - Преобразование данных: Обновление записи в таблице "Задачи" с указанным ID задачи.
    - Запись в таблицы: Таблица "Задачи".
    - Валидация данных: Проверка наличия названия и дедлайна.
6. **Эндпоинт: /api/tasks/{taskId}/delete**
    - Метод: DELETE
    - Описание: Удаление задачи.
    - Входные данные: Нет.
    - Преобразование данных: Удаление записи из таблицы "Задачи" с указанным ID задачи.
    - Запись в таблицы: Таблица "Задачи".
    - Валидация данных: Проверка существования задачи с указанным ID.
7. **Эндпоинт: /api/tasks**
    - Метод: GET
    - Описание: Получение списка всех задач с возможностью пагинации, фильтрации и поиска.
    - Входные данные:
        - Статус (опционально, строка)
        - ID ответственного пользователя (опционально, число)
        - Дедлайн (опционально, дата и время)
        - Страница (опционально, число)
        - Размер страницы (опционально, число)
    - Преобразование данных: Получение задач из таблицы "Задачи" с учетом переданных параметров.
    - Запись в таблицы: Нет.
    - Валидация данных: Проверка корректности формата даты и времени, а также числовых значений страницы и размера страницы.
8. **Эндпоинт: /api/tasks/{taskId}**
    - Метод: GET

- Описание: Получение информации о конкретной задаче по ее ID.
- Входные данные: ID задачи (число)
- Преобразование данных: Получение информации о задаче с указанным ID из таблицы "Задачи".
- Запись в таблицы: Нет.
- Валидация данных: Проверка существования задачи с указанным ID.

9. **Эндпоинт: /api/users**
    - Метод: GET
    - Описание: Получение списка всех пользователей.
    - Входные данные: Нет.
    - Преобразование данных: Получение всех пользователей из таблицы "Пользователи".
    - Запись в таблицы: Нет.
    - Валидация данных: Не требуется, так как запрос не требует входных данных.