Сервис пользовательских рецептов
Тестовое задание

Стек технологий
1. Язык программирования: **Python**
2. Один из фреймворков на выбор: **aiohttp**, **FastAPI** (асинхронный)
3. На выбор один вариантов базы данных: **PostgreSQL**, **MongoDB.** Обязательно использование ORM/ODM (SQLAlchemy>=1.4.0, Tortoise, ODMantic, Beanie и другие)
4. Система контроля версий: git

Описание системы
Сервис для приложения пользовательских рецептов. Пользователи могут просматривать, добавлять, и совершать разные операции с рецептами.

Основные составляющие системы:

- **Пользователь:**
  - идентификатор
  - никнейм
  - статус (активен или заблокирован)
  - избранное
  - дата создания
  - дата изменения

- **Рецепт:**
  - автор (пользователь)
  - дата создания
  - дата изменения
  - название
  - тип блюда (салат, первое, второе, десерт, напиток, выпечка)
  - описание
  - шаги приготовления
  - фотография блюда (ссылка)
  - лайки
  - набор хэштегов
  - статус (активный или заблокирован)

Требования к системе и проекту
- Все внешние взаимодействия с сервисом реализовать через **HTTP** протокол. Интерфейс (API) должен соответствовать принципам **REST**: использовать соответствующие методы, применять, как минимум, базовые коды ответов, соблюдать принципы построения адресов.
- Задание следует выполнять поэтапно, отражая их в системе контроля версий **коммитами**. Выполненное задание залить на хостинг проектов - **github.com**.
- К проекту приложить файл с описанием схемы взаимодействия с сервисом. Он может быть как в каком-то из известных форматов (например, **OpenAPI**), так и в свободном формате (в таком случае, все однотипные вещи — передаваемые аргументы, формат ответа и т. п. — должны быть описаны одинаково)
- Обязательно наличие файла **README.md** с описанием выбранного стека технологий, версий платформы, языка и базы данных. Также в этом файле должно быть описано как развернуть проект с нуля, как сконфигурировать, как запустить и возможные параметры, необходимые для его запуска и использования.

- К проекту приложить файл в котором указаны зависимости с их версиями. Для **pip** это **requirements.txt**. В случае использования **Poetry**, необходимо приложить **pyproject.toml** и **poetry.lock**

Необходимый функционал

Примечание: Если возникают трудности с реализацией чего-либо, можно это пропустить, но это будет минусом.

1. Модели с полным набором полей и связей
2. Пользовательский API
    1. Регистрация и вход пользователя
    2. Получения профиля пользователя: идентификатор, никнейм, статус и количество рецептов пользователя.
    3. Получение первых 10 пользователей (кроме заблокированных), отсортированных по количеству добавленных рецептов (отдаются те же поля, что и при получении профиля)
    4. Добавление пользователем рецепта
    5. Получение списка рецептов (без поля «шаги приготовления»). Исключить заблокированные рецепты. Фильтрация по хэштегу, части названия, типу блюда, автору, наличию фотографии. Сортировка до дате создания, количеству лайков, названию. Добавить пагинацию
    6. Получение конкретного рецепта (все поля рецепта)
    7. Любые действия и любые обращения для заблокированных пользователей, кроме регистрации, входа и получения своего профиля должны быть заблокированы
3. Админский API
    1. Любая простая авторизация, например статичный токен, задаваемый в конфиге.
    2. Блокировка/разблокировка пользователя
    3. Блокировка/разблокировка рецепта

Дополнительный функционал

Примечание: Наличие дополнительного функционала не обязательно, но будет плюсом

1. Пользовательский API
    1. Простановка лайка рецепту
    2. Получение первых 10 пользователей (кроме заблокированных), отсортированных по количеству лайков в их рецептах (те же поля, что и при получении пользователя, плюс общее количество лайков по всем рецептам пользователя)
    3. Добавление в список рецептов никнейма и статуса пользователя, для каждого рецепта
    4. Добавление рецепта в избранное
    5. Получение списка избранных рецептов
    6. Изменение своего никнейма
    7. Изменение своих рецептов
    8. Удаление своего рецепта
    9. Удаление пользователем себя (со всей информацией и рецептами)