# Задача: проектирование и разработка API для создания портала типа ЯндексДзен

## Функционал:

1. **Регистрация автора**
   * Для регистрации автора необходимо предоставить:
     + username
     + telegram\_chat\_id
     + email
     + password
2. **Авторизация в системе**
   * Авторизация на основе токена.
3. **Работа с постами**
   * Создание, изменение и удаление постов.
   * Посты могут просматривать все пользователи (даже не авторизованные).
   * Удалять или изменять пост могут только создатели постов или пользователи со статусом is\_staff.
   * Атрибуты поста:
     + текст
     + дата публикации
     + пользователь (автор)
   * Получение списка постов:
     + В списке постов должны быть указаны все поля постов.
     + Дополнительно должна быть рассчитана средняя оценка данного поста, которая берется из всех оценок за этот пост.
   * Публикация поста:
     + После успешной публикации поста автору должно приходить сообщение в Телеграмм об успешной публикации.
4. **Работа с комментариями**
   * Создание, изменение и удаление комментариев под постом.
   * Комментарии могут просматривать все пользователи.
   * Создавать комментарии могут все (даже не авторизованные). В качестве автора комментария пользователь может указать временный username.
   * Удалять или изменять комментарий могут только пользователи со статусом is\_staff.
   * Атрибуты комментария:
     + пост
     + автор (имя автора комментария)
     + текст комментария
     + дата публикации
5. **Оценка поста**
   * За каждый пост можно ставить оценку от 1 до 5.
   * Ставить оценку могут только авторизованные пользователи.
   * Каждый пользователь может изменить свою оценку.

## Пути, которые должны быть реализованы:

* POST /api/account\_register/ — создать аккаунт
* GET /api/account/ — список всех аккаунтов
* GET /api/post/ — список всех постов
* POST /api/post\_add/ — добавить пост
* GET /api/post/<int:post\_id>/ — детальный просмотр поста
* GET /api/post/<int:post\_id>/comment/ — список всех комментариев к посту
* POST /api/post/<int:post\_id>/comment\_add/ — добавить новый комментарий к посту
* POST /api/post/<int:post\_id>/mark\_add/ — добавить оценку для поста

## Используемые технологии:

* Django версии 4.1 и выше
* Django REST Framework
* SQLite (по умолчанию)
* PostgreSQL (бонус)

## Дополнительные задачи:

* Завернуть приложение в Docker контейнер.
* Использовать PostgreSQL в качестве базы данных.

## Результат:

Проект должен быть размещен на GitHub.