Тестовое задание на должность python backend developer

Необходимо создать сервер авторизации и новостей с комментариями и лайками на Django с использованием RestFramework на python 3.

* У каждого пользователя может быть две роли – пользователь и админ, админ может зайти в админ-панель, пользователь – нет. Как именно решить эту задачу не принципиально. Плюсом будет создание кастомного класса для авторизации наследуемого от BaseAuthentication - нам важно видеть как вы решите эту задачу.
* Каждый пользователь может создать новость. Все пользователи могут получать списки всех новостей с пагинацией. Пользователи могут удалять и изменять свои новости. Админ может удалять и изменять любую новость.
* Также нужно добавить механизм лайков и комментариев новостей – лайкать и комментировать может любой пользователь, автор может удалять комментарии к своим новостям, админ может удалять любые комментарии.   
  При получении списка новостей и одной конкретной новости нужно показать количество лайков и комментариев. Плюсом будет добавление списка последних 10 комментариев при получении списка новостей и одной новост.  
  Плюсом будет реализация механизма через микросервис.

Исходный код приложения необходимо выложить на GitLab или GitHub и запустить сервер. Сервер должен быть доступен для запросов из внешней сети, доступ к серверу (IP или доменное имя) а также роут для доступа к АПИ должны быть в README.md репозитория.

Краткий мануал по созданию виртуального сервера на базе яндекс.облака для новых пользователей даётся бесплатный период использования:  
<https://cloud.yandex.ru/docs/compute/quickstart/quick-create-linux>  
  
Также бесплатный сервер t2 micro можно поднять на Amazone:  
<https://us-east-2.console.aws.amazon.com/ec2/v2/home>

* Желательно:
* контейнеризация в Docker
* использование python venv
* использование web сервера CherryPy, gunicorn и им подобных (какого-нибудь одного)
* демонстрация понимание механизма миграций моделей (можно сделать несколько изменений в какой-либо модели и показать миграции)
* использование .env файла для хранения информации о подключении к базе данных
* файл README.md с описанием скриптов для разворачивания окружения и запуска сервера
* Модели:
* Users (имя пользователя, пароль в зашифрованном виде)
* News (дата новости, заголовок новости, текст новости, автор)
* Comments (Дата комментария, текст комментария, автор)
* Опционально: Tokens (при авторизации можно создавать токены и хранить их в базе или использовать JWT токены)
* Роуты
* POST auth (передаем имя пользователя и пароль, получаем токен если пользователь с таким паролем есть и ошибку, если такого пользователя нет)
* GET news (получаем список новостей с пагинацией)
* POST news (создаем новость, проверка на авторизацию)  
  PUT news (обновляем новость, проерка на атворизацию, проверка на наличие прав)
* DELETE news
* GET comments (получение списка комментариев новости с пагинацией)
* POST comments (создание нового комментария, проверка на авторизацию)
* DELETE comments
* Админка
* Нужна стандартная админка Django (пользователи админки могут не пересекаться с пользователями в таблице Users)

Краткий туториал и документацию по RestFramework можно найти тут: <https://www.django-rest-framework.org/tutorial/quickstart/>