Hi !
This test has been designed to prove your knowledge of Python, Django, and Docker.

Youtube Clone

In this project you can create, list and enter a series of videos, you have been asked to add the following features.

1. Simple login/register feature to keep track on user activity on the website (comments, likes)

2. Upload videos via youtube embed link. Videos should have a title and their youtube embed link. You don’t need to actually upload the video file, just use the youtube embed link

3. Create a like/dislike system for videos where we keep track of the amount of likes and dislikes each video has

4. Create a comment feature for videos. Comments are linked to users and videos

5. Video page should have the embed youtube video and buttons to like, dislike and comment

6. A Home page where all videos are showed randomly. When the user reloads the website a different set of videos should be displayed

7. Create the URL "/popular/" to show the five most popular videos, based on the following rules:

8. Likes add 10 points of popularity

9. Dislikes subtract 5 points of popularity

10. Comments add 1 of popularity

11. Today videos are 100 points of popularity worth than yesterday's videos and so on (e.g. tomorrow video will have 100 more than today's video)

12. In case that all videos have the same popularity points, choose five videos at random.

13. Include only the videos of the current month. If there aren't any new videos, pick five videos at random.
14. Example:

    i. Today video with 2 likes and 1 dislike, and 1 comment -> 116
    ii. Yesterday video with 2 likes and 1 dislike, and 1 comment -> 16

15. This website is on fast-growing, think on a scalable solution (important)

- Create the URL "/history/" to show the videos that the logged user has seen, ordered from new to old, paginated by 10.

- Build the unit tests you consider necessary to test /popular/ and /history/

- Containerize the project using docker and docker-compose that would look like this. (1 Nginx container for load balancing api traffic => 1 django api container =>1 postgres database container)

Clarifications

- You can create, modify and delete any Django app that you consider necessary.
- You can add or remove any library you want.
- Use django templates for the client-facing UI. Use any library that would allow you to have a somewhat decent UI without too much effort like Bootstrap. We don’t care much about the UI of this project.
- Host the project on a public GitHub repo