

Back-End Developer Intern Assignment

Design & Develop Twitter API

Introduction:

In this assignment, the goal is to create a web application using Django, along with a set of APIs for managing tweets. Your work will be evaluated based on factors such as code quality, design choices, creativity, and the uniqueness of your approach.

Instructions:

- Each student must submit their own design and code implementation, which should be distinct from others.
- You're allowed to use any external libraries or frameworks.
- Your code should be well-documented with appropriate comments. Add the readme.md file in your github repo.
- The assignment completion time will be tracked, and you have 24 hours to finish the assignment, starting from when you begin working on it.

Submission:

Here are the steps to submit your code:

1. Put your code on GitHub. This means creating a repository (a place to store and manage your code) on GitHub's website.
2. Make sure your repository is public. This means that anyone can see the code you've uploaded.
3. Share the link of your repository with us in the following google form.

Assignment 1: **(Easy Level)**

Task: Create a Django Rest Framework API for user registration.

Requirements:

1. Create a model for the user with fields: `username` and `password` and other user details..
2. Implement a view or viewset that allows users to register with a POST request containing their username and password and all basic details of the user that you have to decide.
3. Return a response indicating success or failure after registration.

Evaluation Criteria:

1. Correctly creates the user model.
2. Implements a view or viewset for registration.
3. Handles validation and error responses appropriately.

Assignment 2: (Medium Level)

Task: Extend the API to allow users to post tweets and view tweets from the last 24 hours.

Requirements:

1. Create a model for tweets with fields: `user` (foreign key to user model), `content`, and `timestamp`.
2. Implement a view or viewset that allows users to post tweets.
3. Implement a view or viewset that allows users to view tweets posted in the last 24 hours.

Additional Constraints:

1. Users can post a maximum of 3 tweets every 24 hours.

Evaluation Criteria:

1. Correctly creates the tweet model and establishes the relationship with the user.
2. Implements views or viewsets for posting and viewing tweets.
3. Enforces the constraint of maximum 3 tweets per user every 24 hours.
4. Properly handles timestamp filtering for viewing tweets.

Assignment 3: (Hard Level)

Task: Implement Tweet Search Functionality

Requirements:

1. Enhance the API to allow users to search for tweets using specific keywords or a username.
2. Implement a view or viewset that accepts search queries and returns relevant tweets.
3. Implement search functionality for both:
 - Finding tweets that contain the given keyword(s) as a sub-string in their content.
 - Finding tweets posted by a specific user using their username.
4. There's a no-cost platform that can assist you in putting your Django project online and making your APIs accessible to the public. You can publish your project and place the API links in the documentation on your GitHub page.

Platform: <https://render.com/> , Tutorial: <https://www.youtube.com/watch?v=wcZWm8j4v9w>

Evaluation Criteria:

1. Successfully implements search functionality for tweets based on keywords and usernames.
2. The search returns the expected results based on the provided query.
3. Properly handles edge cases and potential errors in the search process.

Modern product-based startups often assess job candidates by reviewing their open-source contributions and GitHub profiles. This hiring approach emphasizes code quality, documentation, and contributions as key factors, enhancing your chances of making a standout impression.

Good luck