Pre-bootcamp Challenge Preparation

BUILD A PRODUCT: StackOverflow-lite

Project Overview

StackOverflow-lite is a platform where people can ask questions and provide answers.

Required Features

1. Users can create an account and log in.
2. Users can post questions.
3. Users can delete the questions they post.
4. Users can post answers.
5. Users can view the answers to questions.
6. Users can accept an answer out of all the answers to his/her question as the preferred answer.

Optional Features

1. Users can upvote or downvote an answer.
2. Users can comment on an answer.
3. Users can fetch all questions he/she has ever asked on the platform
4. Users can search for questions on the platform
5. Users can view questions with the most answers.

You are expected to build a RESTful API with all the endpoints defined below. Your API should meet all the requirements listed under the required features section and your data should be persisted with a database. You are to write SQL queries that will help you write to and read from your database. The endpoints are to be secured with JWT.

NB:

- *You are to create a pull request to elicit review and feedback when you are done working on this challenge.*
- *Important concepts to keep in mind while doing this challenge are **OOP** and **TDD**.*

Tools

- Server-Side Framework: <*Flask Python Framework*>
- Linting Library: <*Pylint, a Python Linting Library*>
- Style Guide: <*PEP8 Style Guide*>
- Testing Framework: <*PyTest, a Python Testing Framework*>

Guidelines

1. **On Pivotal Tracker, create user stories to setup and test API endpoints that do the following using data structures**
    - Get all questions.
    - Get a question.
    - Post a question.
    - Post an answer to a question.
2. **On Pivotal Tracker create stories to capture any other tasks not captured above. The tasks can be feature, bug or chore for this challenge.**
3. **Setup the server side of the application using the specified framework**
4. **Setup linting library and ensure that your work follows the specified style guide requirements**
5. **Setup test framework**
6. **Version your API using url versioning starting, with the letter "v". A simple ordinal number would be appropriate and avoid dot notation such as 2.5. An example of this will be : https://somewebapp.com/api/v1/users**
7. **Using separate branches for each feature, create version 1 (v1) of your RESTful API to power front-end pages**
8. **At minimum, you should have the following API endpoints working:**

EndPoint	Functionality	Note
`POST /auth/signup`	Register a user	
`POST /auth/login`	Login a user	
`GET /questions`	Fetch all questions	
`GET /questions/<questionId>`	Fetch a specific question	This should come with the all answers provided so far for the

		question.
`POST /questions`	Post a question	
`Delete /questions/<questionId>`	Delete a question	This endpoint should be available to the question author.
`POST /questions/<questionId>/answers`	Post an answer to a question	
`PUT /questions/<questionId>/answers/<answerId>`	Mark an answer as accepted or update an answer.	This endpoint should be available to only the answer author and question author. The answer author calls the route to update answer while the question author calls the route to accept answer.

9. **Write tests for the API endpoints**
10. **Ensure to test all endpoints and see that they work using Postman.**
11. **Integrate TravisCI for Continuous Integration in your repository (with *ReadMe* badge).**
12. **Integrate test coverage reporting (e.g. Coveralls) with badge in the *ReadMe*.**
13. **Obtain CI badges (e.g. from Code Climate and Coveralls) and add to *ReadMe*.**
14. **Ensure the app gets hosted on Heroku.**

Target skills

After completing this challenge, you should have learnt and be able to demonstrate the following skills.

Skill	Description	Helpful Links
Project management	Using project management tool(pivotal tracker) to manage your progress while working on tasks.	• To get started with Pivotal Tracker, use Pivotal Tracker quick start. • Here is an sample template for creating Pivotal Tracker user stories.
Version control with GIT	Using GIT to manage and track changes in your project.	• Use the recommended Git Workflow, Commit Message and Pull Request (PR) standards.
HTTP & Web services	Creating API endpoints that will be consumed using Postman	• Guide to Restful API design • Best Practices for a pragmatic RESTful API

Test-driven development	Writing tests for functions or features.	• Getting started with TDD in Python
Data structures	Implement non-persistent data storage using data structures.	• Data Structures in Python
Continuous Integration	Using tools that automate build and testing when code is committed to a version control system.	• Continuous Integration in Python
Holistic Thinking and big picture thinking	An understanding of the project goals and how it affects end users before starting on the project	

1andelA!!!

3Tia###3