WEZACARE BACKEND INTERNSHIP CHALLENGE

Synopsis

Kevin, a social worker consulting with different children homes in the region reaches out with a problem he's been facing. He elaborates that he usually finds issues with getting truthful, accurate and readily available answers to questions about various social issues while in the field. He therefore asks the engineering department if they can roll out a service that can enable social workers to ask and answer questions from colleagues.

Bare-minimum features

The engineering team receives Kevin's request and after a careful analysis and design phase, come up with the following as features that the first iteration of Kevin's desired service must have:

1) Users can create an account as well as log in to the platform
2) Users can post questions on the platform
3) Users can answer questions posted only by others on the platform
4) Given an ID, users can retrieve a particular question with that ID, along with the answers to the question.
5) User can view all the questions that they have ever asked on the platform

Expectations

You are delegated the role of lead backend engineer for Kevin's service and are therefore responsible for developing API endpoints that will expose the expected functionality to frontend clients. It is expected that you implement the outlined features and expose them through the following endpoints:

EndPoint	**Functionality**	**Note**
POST /auth/register	Registers a user	
POST /auth/login	Log a user in	
GET /questions	Fetch all questions	
POST /questions	Post a question	Only authenticated users will be allowed to post questions to the platform.
GET /questions/<questionId>	Fetch a specific question	This should come with all answers provided so far for the question.
Delete /questions/<questionId>	Delete a question	A question can only be deleted by the

		authenticated author of the question.
POST /questions/<questionId>/answers	Post an answer to a question	Only authenticated users can provide answers to a question
PUT /questions/<questionId>/answers/<answerId>	Update an answer.	This endpoint should be available to only the authenticated answer author.. The answer author calls the route to update the answer.
DELETE /questions/<questionId>/answers/<answerId>	Update an answer.	This endpoint should be available to only the authenticated answer author. The answer author calls the route to update answer

Useful technical information

1. We insist that you do not bother coding any client side version of this service. You are required to hand over only a backed service that exposes the implemented features through a suite of RESTful APIs.
2. You are required to persist data between sessions and are thus expected to store your data in a database, preferably an SQL database. Any other database can be used as long as the candidate justifies why they adopted the particular database type.
3. When querying the database, both raw queries and ORMs are allowed.
4. As TDD is strongly embedded in Wezacare's engineering culture, you are expected to flex your TDD knowledge by including a comprehensive test suite.
5. Where necessary i.e where user identity is required , protect your API endpoints with a JWT or a token specific to a particular user.
6. You are expected to implement this challenge in a stack familiar to Wezacare's backend stack. Only solution written in **NodeJS(Express)/Python(Flask/Django)** will be accepted

NOTE: While a complete solution would be preferred, it is not necessary. Incomplete solutions will also be accepted, as long as they satisfy at least 70 % of the expectations. However, Projects submitted past the deadline will not be accepted.
You are given a maximum of up to 5 days to submit your final project. Submissions will be done by sending a link of the Github repository hosting your project to **bruno.obonyo@wezacare.org.**