iiDENTIFii

Job Title: Backend Software Engineer
Candidate Technical Assessment: Building a web text forum

Task Description:

You are tasked with building the API and datastore backend of a web forum for a small number of users (< 100). The forum is a basic text system which has the capabilities to add posts, retrieve posts, and like posts. Management does not believe in users editing or deleting existing posts, for ethical reasons.

Task Specifications:

Functionality:

1. The forum is predominantly made up of posts created by users. Posts can be commented on. Posts can be assigned likes, each user can only like a post once, and cannot like their own post.
2. Users can view posts anonymously but must log in to post, comment, or like.
3. There are two categories of users, regular users and moderators. Regular users post, comment, and like. Moderators can tag posts as “misleading or false information”, for regulatory reasons.
4. The system should allow you to retrieve and page posts, as well as their associated comments.
5. The system should allow filters like a date range, author as well as tags and also have sort options for date and like count.
6. The system should allow you to add comments
7. The system should allow you to increment and decrement likes
8. The system should allow moderators to tag posts.

Backend:

1. The system should be built in C# using ASP.NET
2. The datastore backend may be whatever makes sense to you, your choice should be justified, your datastore should have some kind of out-of-the-box integration into ASP.NET, and efficient usage of the datastore is important.
3. User authentication needs to be handled in the application, do not use an external auth provider. Users need to log in with a password, you are welcome to extend this with 2FA or social auth but this is not a requirement.
4. We want to enable regular users to use external third-party applications to interact with our system in an automated way, make sure that there is a simple API for this, documented in Postman, so that users can interact with it easily.

Testing:

1. You can either provide a rudimentary UI if you opt for an MVC pattern or you can provide a postman collection.

Submission Guidelines:

1. Code must be submitted using a public git repository in GitHub, with a full commit history. Do not do the whole project in one go and commit it all at once, your use of Git is an important part of this assessment
2. Instructions to run the project need to be in the readme, the assessor needs to be able to run it on a local machine using only the readme as instructions to do so (don’t forget external dependencies!).
3. Your submission should also contain a datastore of test/dummy data for the assessor to use
4. Submit a public postman collection which allows easy testing of your API endpoints