Backend Task - AI Quizzer

Task

Create a microservice with the following API endpoints:

- Authentication endpoint
- REST API endpoints for an AI Quizzer app that manages student quizzes and scores (details below)

Core functionalities and API Details

Authentication endpoints for Login

- Treat the endpoint as a mock authentication service and accept any username/password.
- The request body should contain a username/password pair.
- Return a signed JSON Web Token (JWT, https://jwt.io/) which can be used to validate subsequent requests.

AI Quizzer REST API endpoints

- Generates new quiz with data provided in the request body (including grade level) **(AI)**
    - Sample payload
- Submits quiz answers and returns evaluated score **(AI)**
    - Sample payload
- Retrieves quiz history and scores based on filters provided in the request
    - Filter based on grade, subject, marks, completedDate, etc.
    - Filter Specific date range i.e from=01/09/2024, to=09/09/2024
- Allows retrying a quiz and re-evaluates scores
    - Old submission should also be accessible

Application hosting

- Create and deploy the docker image on a third-party hosting service (Heroku, DigitalOcean, AWS, etc)

Important:

- Attach API documentation with a request which can be executed directly. You can use Postman Collection, Swagger, etc for documentation.
- Attach README instructions that contain all required information
- SQL files to create required tables should be present in the project.
- Please refrain from adding node_modules or other auto generated packages in your submission.

Bonus functionalities: Extra points for attempting these :)

- Integrate an endpoint which provide hints on given question **(AI)**
- Implement a notification feature to send results over email
    a. The results should include two suggestions, to improve skills based on quiz responses **(AI)**
- Integrate a caching layer mechanism (e.g., Redis) for reduced API latency while fetching quiz data

Technical Requirements and Guidelines

- You are free to use any language (Java, Node.js, Python, etc.)
- You are free to use any RDBMS / NoSQL.
- There are no restrictions on the frameworks to use / database design schema. Choose whichever seems best suited for the task.
- You can use any AI tool to integrate and generate quiz data. (Suggested AI Tool: Groq)

Things we are interested in

- Completeness of the APIs - authentication & AI Quizzer with handling of all possible edge cases
- Knowledge of REST APIs, SQL/NoSQL, JWT, and integration with AI services
- Your database design
- Modularity & readability of code
- Attention to detail and quality

How to submit

Please email your submission to the following address.

To: assessments@playpowerlabs.com
Subject: <Your Name> - <College shortname> Backend Task

Along with:
1. Zip of your code
2. Hosted URL (Optional) [Netlify, Heroku, etc..]
3. Screen recording of your working application (Optional) [You can use Loom to create one and send us the link]

For any clarifications contact: hiring.support@playpowerlabs.com

Important

We condemn plagiarism. Please maintain the dignity and originality of your work. If we suspect any attempt towards copying, we will disqualify your submission.
Also, Don't put your code on GitHub.
Please DO NOT submit *node_modules* or other auto generated packages.

Remember: While bonus features are impressive, prioritize completing the core features with high-quality, well-structured code. Demonstrate your understanding of modern web development practices and your ability to create efficient, user-friendly applications. We thoroughly review each submission, examining both the overall functionality and the nuances of your code. Show us your best work!