Recruit CRM : Technical Assignment (Freshers)

Overview:

In this assignment, your aim is to create -

- **a JWT (JSON Web Token) authentication system** using **Node.js.**
- The project will involve creating a **microservice** for a **public API** and **connecting it to the main service.**

Project Components:

- **Main Service:** This will be the core service responsible for handling user authentication and authorization using JWT.

- **Public API Microservice:** This microservice will provide a public API key that can be used to access main service routes without needing to login with credentials. [With the help of api key]

- **What is a Public API?**
  A public API, or Application Programming Interface, is a set of rules and protocols that allows one piece of software or application to interact with another. In the context of web development, a public API is made available by a service or platform to enable third-party developers to access certain features or data.

Main Service:

- **Implement user authentication using JWT.**
  - Generate JWT tokens upon successful login.
  - Validate JWT tokens for secure endpoints.

- **Implement endpoints to add candidates to the Database and retrieve them.**

- **Public API Microservice:**
  - Create a public API that does not require authentication with email and password but whose endpoints are rather authorised with api key.
  - Include at least two endpoints in the public API.

Connection between Main Service and Public API Microservice:

The main service should be able to communicate with the public API microservice to access its functionalities.

Resources:

1. User Database should have minimum these fields:

id	first_name	last_Name	email	password_hash

2. Following are the must have JWT authentication endpoints

Endpoint 1: POST /api/register
Endpoint 2: POST /api/login
Endpoint 3: POST /api/protected [Should not be accessible without logging in]

3. Other must have endpoints

Endpoint 1: POST /api/candidate
[To add candidate to the database]

Candidate DB:

id	first_name	last_Name	email	user_id

Where user_id is the id of the user [owner] who added Candidate.

Endpoint 2: GET /api/candidate
[It should retrieve candidates for whom the current user is the owner]

4. Must have public api endpoints

Endpoint 1: POST /api/public/profile
[This should retrieve the profile information in json format of the user corresponding to whose API key is being used.

Endpoint 2: GET /api/public/candidate
[This should retrieve all candidates respective to the user whose api key is being used].

<u>Documentation:</u>

- Provide a comprehensive documentation guide on how to set up and run both services.
- Include all necessary commands to initialise the projects.
- Clearly list the commands to install any required dependencies.
- If the project doesn't execute because of any missing dependencies which are not mentioned in the documentation, it will be a disqualification.

<u>Submission:</u>

- **Code Submission:**
  Submit the code for both the main service and the public API microservice. Include clear comments and documentation within the code.
- **Documentation Submission:**
  Submit a well-structured documentation guide in a separate document format (e.g., PDF). Include step-by-step instructions for setting up, running, and connecting both services. List all necessary commands and configurations.

IMPORTANT FOR SUBMISSION:

The **zip file** must contain the following:
1. Code
2. Documentation
3. Database dump

Compress the main folder and send a zip file on the form:
https://forms.gle/AmfZEo8HApHhEdPz9

****Note** - Failing to provide any of the above means disqualification.**

<u>Evaluation Criteria:</u>

Your assignment will be evaluated based on the following criteria:

- **Functionality:** Does the authentication system work as intended? Are the public API endpoints accessible with public api?

- **Code Quality:** Is the code well-structured, modular, and follows best practices? Are there comments where necessary?

- **Documentation:** Is the documentation clear and comprehensive? Can someone new to the project easily set up and understand the system?

- **Microservices Architecture:** How well are the main service and public API microservice designed to work together as separate entities?

- **Security:** Is the authentication system secure, and are best practices followed for communication between microservices?

End of the document*************************************