Task: Backend developer (NestJS)

This section focuses on assessing the practical skills of a backend developer using Node.js and NestJS. The tasks evaluate the candidate's ability to build and optimize APIs, handle databases, and implement key backend functionalities.

Timebox and submission guidelines

- ***Framework:*** *NestJS*
- ***Format:*** *Please submit your work as a Git repository (e.g., via GitHub, GitLab) or as a ZIP file containing the project files.*
- ***Instructions:*** *Ensure that your submission's README file includes all required setup instructions (e.g., how to run the project and install dependencies).*

Task 1: Basic API development

Objective: *Build a simple CRUD API for managing a list of users.*

Requirements:

- *Create endpoints for creating, reading, updating, and deleting users.*
- *Implement proper validation for the input data.*
- *Ensure error handling is in place for invalid operations (e.g., trying to update a non-existent user).*

Task 2: Middleware implementation

Objective: *Implement a middleware to log the time taken for each API request.*

Requirements:

- *The middleware should log the start and end times of each request.*
- *Ensure the log includes the endpoint accessed and the HTTP method used.*

Task 3: Error handling

Objective: *Create a global error-handling mechanism.*

Requirements:

- *Implement a global exception filter to handle exceptions (e.g.,* `NotFoundException`*,* `BadRequestException`*).*
- *Ensure that the error responses are consistent and informative.*

Task 4: Database integration

Objective: *Set up a PostgreSQL database connection and integrate it with the API.*

Requirements:

- *Use Prisma or TypeORM to connect to the PostgreSQL database.*
- *Create a service that fetches data from a specific table and returns it as a REST endpoint response.*

Task 5 (optional): Unit testing

Objective: *Write unit tests for one of the services implemented in the API.*

Requirements:

- *Use Jest for writing the tests.*
- *Ensure the tests cover both positive and negative cases.*

Task 6 (optional): JWT authentication

Objective: *Implement JWT-based authentication for the user CRUD API.*

Requirements:

- *Secure the CRUD endpoints so that only authenticated users can access them.*
- *Ensure the JWT tokens are properly validated and handled.*