Yêu cầu thêm của bài test Intern

**1. Thiết kế cơ sở dữ liệu (DB Design):**

• Bảng user: Chứa thông tin cơ bản của người dùng.

o id (Primary Key)

o username

o password

o email

o roleId

o profileId

• Bảng role: Chứa các vai trò mà người dùng có thể có.

o id (Primary Key)

o name

• Bảng profile: Chứa thông tin chi tiết của người dùng.

o id (Primary Key)

o username

o email

**2. Phân tích các chức năng:**

• Đăng ký:

o Người dùng mới có thể đăng ký tài khoản bằng cách cung cấp username, password, email.

o Khi đăng ký thành công, một bản ghi mới sẽ được tạo trong bảng user.

o Mỗi user mới sẽ được gán vai trò mặc định là ROLE\_USER.

o Profile của user sẽ được tạo và liên kết với user mới.

• Đăng nhập:

o Người dùng có thể đăng nhập bằng username và password.

o Sau khi xác thực thành công, hệ thống sẽ tạo một JWT token và trả về cho client.

• JWT để xác thực API:

o Các API sẽ yêu cầu JWT để xác thực.

o Hệ thống sẽ kiểm tra tính hợp lệ của JWT và xác định quyền truy cập của người dùng dựa vào vai trò.

• API /api/v1/me:

o API này sẽ trả về thông tin người dùng hiện tại dựa trên JWT.

o Thông tin bao gồm chi tiết của user, role và profile.

o Đảm bảo thông tin trả về đầy đủ nhưng không bao gồm dữ liệu nhạy cảm như password.

• Biến context:

o Tạo một biến context chung để lưu trữ các thông tin như requestId, userId, timestamp.

o Context này sẽ được sử dụng trong các API để theo dõi thông tin của các request

**API Documentation (API Docs):**

**• Đăng ký:**

o Method: POST

o Endpoint: /api/auth/register

o Request Body:

{

"username": "string",

"password": "string",

"email": "string"

}

Response:

• 200 Created: Khi đăng ký thành công.

• 400 Bad Request: Khi dữ liệu đầu vào không hợp lệ hoặc user đã tồn tại

**Đăng nhập:**

o Method: POST

o Endpoint: /api/ auth /login

o Request Body:

{

"username": "string",

"password": "string"

}

Response:

* 200 OK: Khi đăng nhập thành công, trả về JWT token.
* 401 Unauthorized: Khi thông tin đăng nhập không chính xác

**API /api/v1/me:**

• Method: GET

• Endpoint: /api/v1/me

• Headers:

o Authorization: Bearer <jwt\_token>

• Response:

“requestId”: “String”,

“userId”: “Long”,

“userid”:{

"id": "Long",

"userName": "string",

"userName": "Null",

"email": "string",

"roles": {

“id”:”Long”,

“roleName”:”String”

},

"profileid": {

"id": "Long",

"userName ": "string",

"email": "String"

},

"timestamp": "Instant"

}

• 200 OK: Khi lấy thông tin thành công.

• 401 Unauthorized: Khi JWT không hợp lệ hoặc đã hết hạn.

**Phân tích thêm:**

Bạn có thể triển khai các chức năng trên bằng cách tuân theo mô hình MVC, với Spring Security để xử lý việc xác thực và quản lý quyền truy cập, và JPA để tương tác với cơ sở dữ liệu. Hãy đảm bảo cấu hình các file application.properties và các service, repository, controller tương ứng để triển khai đầy đủ các chức năng yêu cầu.