

Group Member:

1. Prave Vinuth
2. Rom Tola
3. Sokhom Panha
4. Sreng Panha

Project Progression

Project Name: Library tracking system

In the second week we have done some part such as:

- **update table schema**
 - change Id datatype from varchar to bigint
- **AuthController**
 - Created REST controller for authentication.
 - Exposed endpoints for **login** and **register**.
 - Integrated with **UserService**.
- **Login**
 - **LoginRequest**: username/email & password DTO implemented.
 - **LoginResponse**: returns auth result (message / token / user info).
 - Validation and error handling added.

The screenshot shows a Postman collection named "E-commmerce" with a "New Request" item. The request is a POST to `http://localhost:8080/api/login`. The body is set to raw JSON with the following content:

```

1  {
2    "username": "admin",
3    "password": "123456"
4  }

```

The response status is 200 OK, with a response time of 634 ms and a response size of 614 B. The response body is:

```

1  {
2    "username": "admin",
3    "message": "Login successful",
4    "token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJhZG1pbkiSImhdCI6MTc2NjQ3ODc1NCwiZXhwIjoxNzY2NTeONzU0fQ.sGoYhUl20KEE-kuBmmfCCzH1L_BViVaL5k2Mi-wgocU"
5  }

```

- **Register**

- **RegisterRequest**: user input DTO (username, password, email, etc.).
- **RegisterResponse**: registration status and user data.
- Password encoding applied before saving.

The screenshot shows the Postman interface. At the top, it says "POST" and "http://localhost:8080/api/register". Below that, there are tabs for "Docs", "Params", "Authorization", "Headers (9)", "Body" (which is selected), "Scripts", and "Settings". Under "Body", there are options for "none", "form-data", "x-www-form-urlencoded", "raw" (which is selected), "binary", "GraphQL", and "JSON" (which is selected). To the right of these are "Schema" and "Beautify" buttons. The "Body" section contains the following JSON:

```

1  {
2    "username": "nuth",
3    "password": "1233",
4    "full_name": "Prave Vinuth",
5    "email": "pravevinuth@gmail.com"
6  }
7

```

Below the body, there are tabs for "Body", "Cookies (1)", "Headers (14)", and "Test Results". The "Test Results" tab shows a green "201 Created" status with "322 ms" and "587 B" response times. To the right, there are icons for "Save Response" and other actions. The "Body" tab shows the JSON response received:

```

1  {
2    "username": "nuth",
3    "token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJudXRoIiwiaWF0IjoxNzY2NDc4NDI3LCJleHAiOjE3NjY1MTQ0Mjd9. oEMTdPIxGz5vx5k36jV_aD_2j4-9ZYTx_BiLRqvQ9zs"
4  }

```

At the bottom, there is a table showing user data:

	id	username	password	full_name	email
1	admin	\$2a\$12\$7CN.L2JUFvZPgTr2sC03vOJq8JLW3V...	Administrator	admin@example.com	
2	testuser	\$2a\$12\$7Ze4rIHn3LkZ0D7H5QNzMuGZXJzsT...	Test User	test@example.com	
6	testuser2	\$2a\$12\$3ZONyWrkmscQyHIPW5qtOnBJXPyn...	Test User 2	test2@example.com	
7	mynewuser123	\$2a\$12\$MMIsLWZccT9DHUqPrZkgNeGPalpx3...	My Name	my@email.com	
8	testuser7	\$2a\$12\$CZgVH5iUdzNxilhlie3SWuRLG9nwxF...	Test User 3	test22@example.com	
9	nuth	\$2a\$12\$4PoDeTUsoO5S.QqqQ7KsFucl4ucgh...	NULL	pravevinuth@gmail.com	
	NULL	NULL	NULL	NULL	

- **UserService**

- Business logic for login and registration.
- User lookup and creation implemented.
- Connected to repository and security components.

- **JwtFilter**

- Implemented JWT authentication filter.
- Extracts token from request header.
- Validates JWT and sets authentication in SecurityContext.
- JwtProperties
- Configured JWT settings (secret key, expiration time).
- Loaded values from application.yml / properties.

- **JwtUtil**

- Implemented JWT generation and validation logic.
- Extracts username and checks token expiration.
- Used by AuthController and JwtFilter.
- SecurityConfig
- Configured Spring Security with JWT.
- Disabled CSRF for REST API.
- Defined public and protected endpoints.
- Registered JwtFilter in security filter chain.