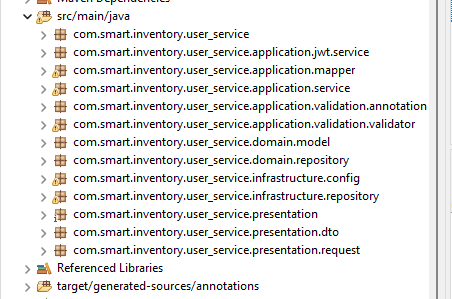
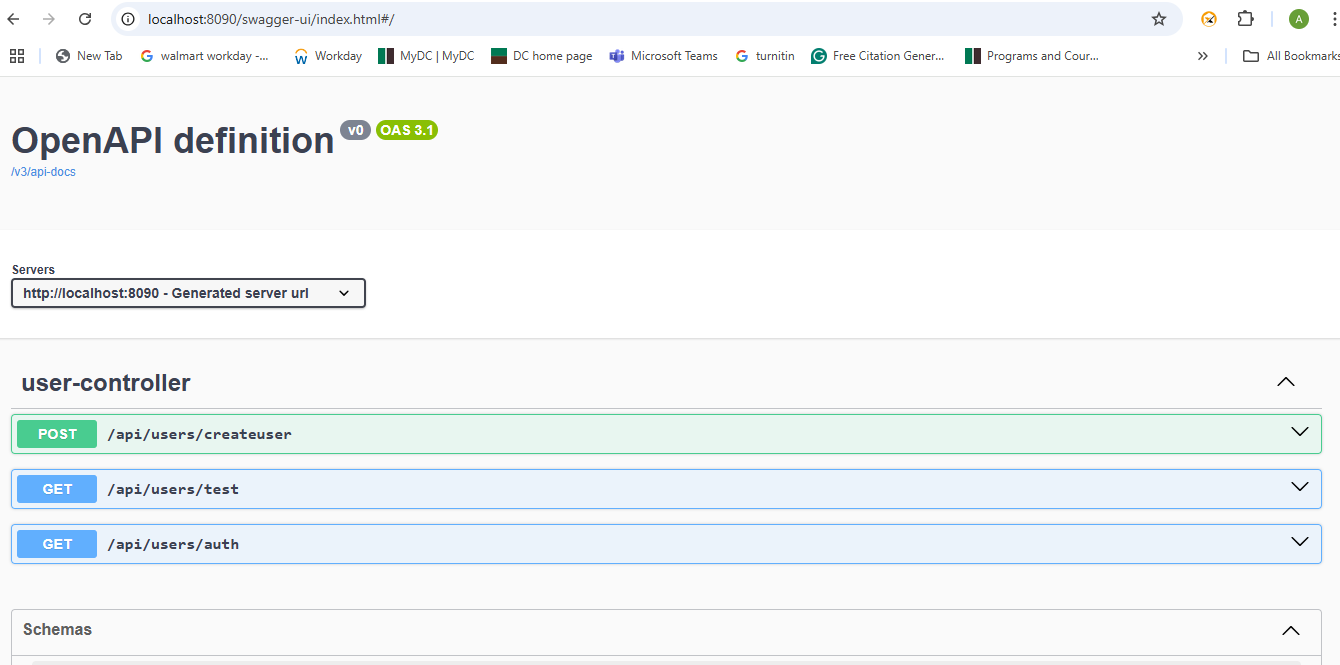
**Spring Boot Microservices Clean Architecture Pattern:**



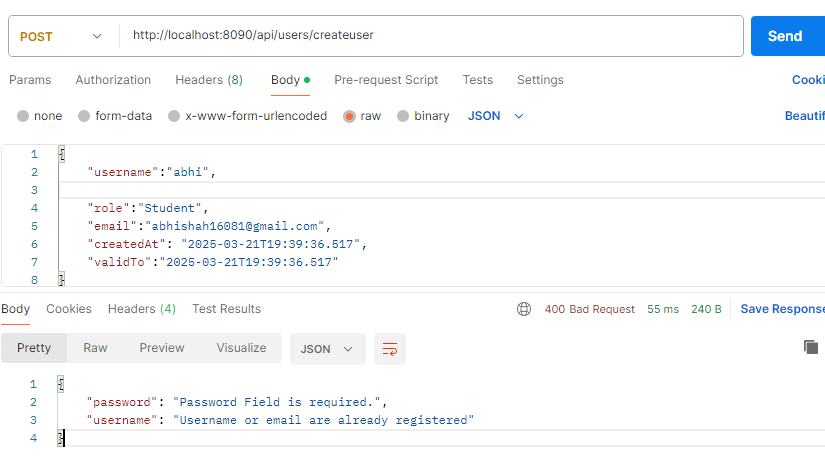
**Learned about Spring boot, different bean concepts, its scope and the other important concepts along with spring security and spring data (JDBC). Added user defined annotations.**

|  |  |  |
| --- | --- | --- |
| Sr No. | Functionalities: | Description |
| 1 | Configure spring project from this link | Used <https://start.spring.io/>  Developed spring application from this by selecting spring latest stable version, maven or gradle – selected maven and language – Java/ kotlin/ Grovy , selected Java as language. |
| 2 | POM.xml | Used to add third party dependencies such as jjwt, map struct for automapping of the models |
| 3 | Application.properties | Added  spring.profiles.active=dev  here based on environment, it will set values for different properties and configurations:  Application-dev.properties – development environment.  Application-staging.properties – staging environment.  Application-prod.properties – production environment. |
| 4 | org.mapstruct | Used as Automapper (to convert request object to object and from object to DTO) automatically based on same property names.  userRequest -> user -> userDto (userDto returned from API as response).  userRequest – request object from frontend.  User- user object to perform DB operation (from service (Applicationn layer)-> infrastructure layer (Repository) interact with db.  Used MS-SQL as the database. |
| 5 | JWT token generation based on username and role fields of the user. | Once user is validated against username and password, used jjwt as third party in maven to generate token using username and role after authentication based on username and password.  Here, token is returned in the response header (token header). |
| 6 | Used Hashing with salt and pepper to generate password. | Password stored in hash format: (SHA-256). |
| 7 | Aspect Oriented Programming (AOP) | Used AOP programming for JWT authentication for Controller endpoint.  AOP can also be used for the logging, Caching and some filtering before executing some API endpoints.  (@before). |
| 8 | **Will work on Redis cache using Docker.** |  |
| 9 | **Will work on Kong API gateway.** |  |
| 10 | **Integrate with Docker and kubernates.** |  |

Integrated swagger OpenAPI documentation for API:



**Added API request validation: (Hibernate 2.0 with @RestControllerAdvice (class level) and @ExceptionHandler at method level.**



**JWT token from get request (Auth):**

