Java Task Assignment :-

OAuth2 Implementation with Multi-tenancy Support using Spring Security 6

1 - Resource Owner ->The service responsible for controlling resources' access

2 – Client -> Resource used by client.

3 - Authorization Server -> The service handling authorization process acting as a middleman between client and resource owner.

4 - Resource Server -> The service who actually supplies the resources

1. **Spring-security-client** (server.port: 8080)

   Dependencies –

   Spring web, Spring Data JPA, MySQL Driver, Lombok, OAuth2 Client

   Model –

   UserModel (user_data), PasswordModel (reset_pass)

   Entity –

   User, PasswordResetToken, VerificationToken

   Repository –

   PasswordResetTokenRepo, UserRepo , VerificationTokenRepo
   extends JpaRepository

   Event –

   RegistrationComp, RegistrationCompleteListener

   Service –

   Interface -> UserService
   Class - > UserServiceImpl (defines all the methods of interface)

   Configuration –

   - WebSecurityConfig (using @EnableWebSecurity)

    - Standard Bean class SecurityFilterChain injected with http methods.
    - PasswordEncoder bean injected.
  - ➢ WebClientConfig
    - WebClient and OAuth2AuthorizedClientManager bean injected

Controller –

RegistrationController, HelloController

2. **Authorization-Server** (server.port : 9000)

Dependencies –

Spring web, Spring Data JPA, MySQL Driver, Lombok, OAuth2 Authorization Server

Entity –

User

Repository –

UserRepo extends JpaRepository

Service –

- ➢ CustomUserDetailsService by implementing UserDetailsService,
- ➢ CustomAuthenticationProvider by implementing AuthenticationProvider

Configuration –

- ➢ AuthorizationServerConfig (using RegisteredClientRepository)
  - Selection of use an in-memory or JDBC based storage for client details with inMemory or jdbc methods
  - Client's basic authentication
    using clientId and clientSecret (encoded with the
    chosen PasswordEncoder bean) attributes
  - Grant types allowed using authorizedGrantTypes attribute

- Defines access scopes with scopes method
- Identify client's accessible resources

- ➢ DefaultSecurityConfig (using CustomAuthenticationProvider)
  - Standard Bean class SecurityFilterChain injected

3. **Resource- Server** (server.port : 8090)

   Dependencies –

   Spring web, OAuth2 Resource Server

   Config –

   ResourceServerConfig (using SecurityFilterChain and http methods).

   Controller –

   UserController