**Spring REST using Spring Boot 3**

**Exercise 1:** **Create authentication service that returns JWT**

1. Setup Spring Boot Project

2. Create JwtUtil.java

|  |
| --- |
| JwtUtil.java  package com.cognizant.spring\_learn.util;  import java.util.Date;  import io.jsonwebtoken.Jwts;  import io.jsonwebtoken.SignatureAlgorithm;  public class JwtUtil {      private static final String SECRET\_KEY = "secret";      public static String generateToken(String username) {          return Jwts.builder()                  .setSubject(username)                  .setIssuedAt(new Date(System.currentTimeMillis()))                  .setExpiration(new Date(System.currentTimeMillis() + 10 \* 60 \* 1000))                  .signWith(SignatureAlgorithm.HS256, SECRET\_KEY)                  .compact();      }  } |

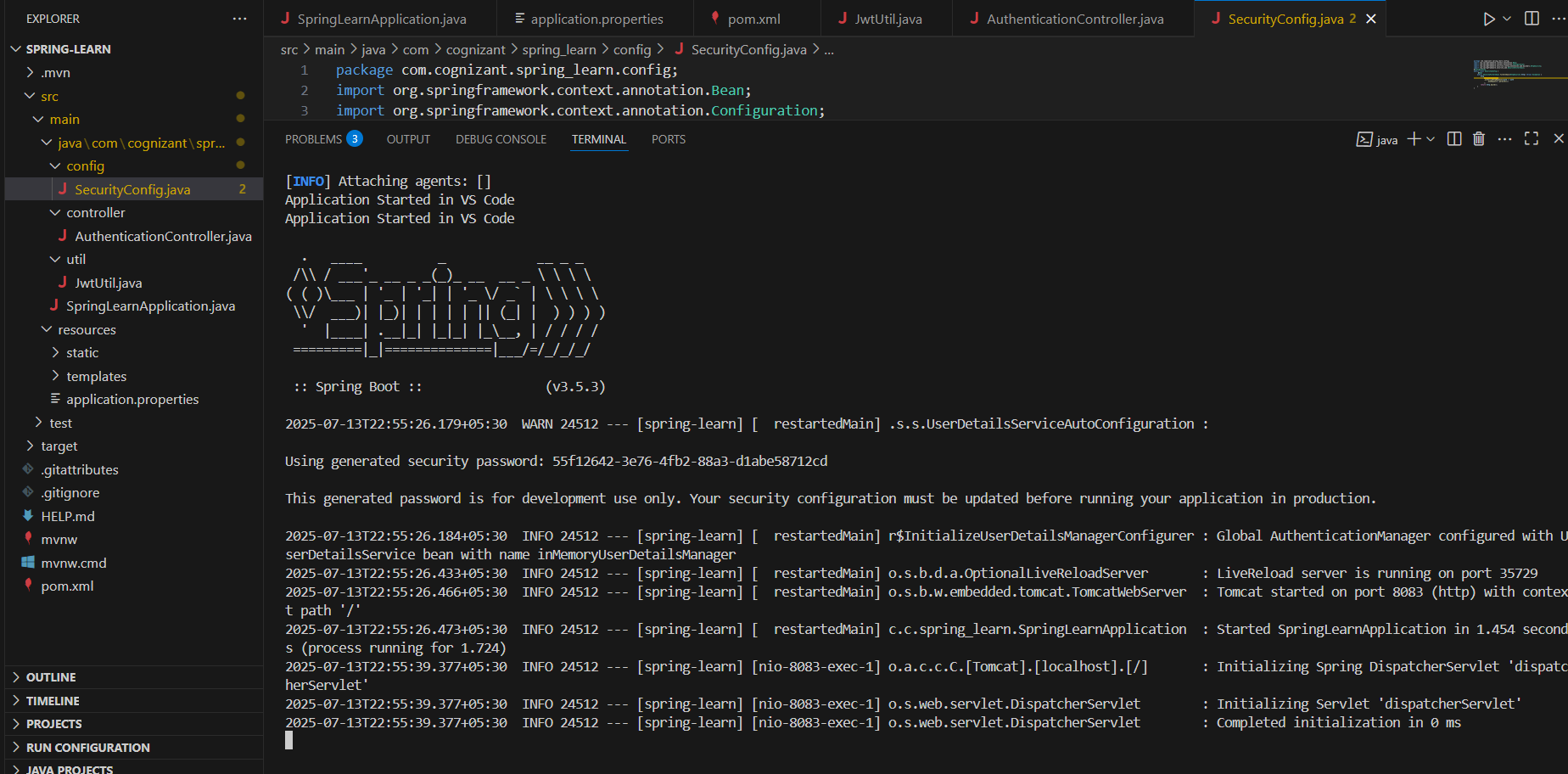
3. Create AuthenticationController.java

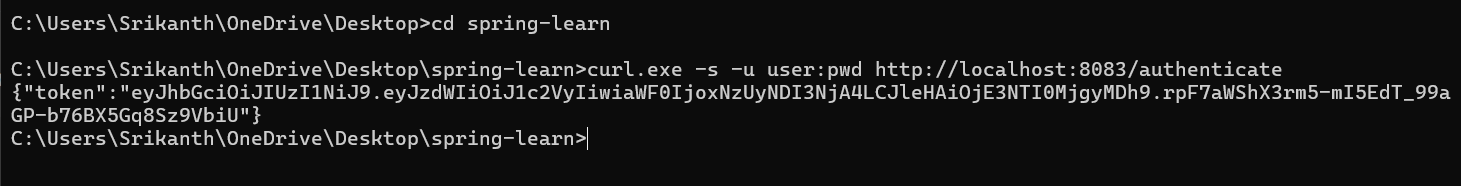
|  |
| --- |
| AuthenticationController.java  package com.cognizant.spring\_learn.controller;  import java.util.Base64;  import org.springframework.http.ResponseEntity;  import org.springframework.web.bind.annotation.GetMapping;  import org.springframework.web.bind.annotation.RestController;  import com.cognizant.spring\_learn.util.JwtUtil;  import jakarta.servlet.http.HttpServletRequest;  @RestController  public class AuthenticationController {       @GetMapping("/authenticate")      public ResponseEntity<?> authenticate(HttpServletRequest request) {          String authHeader = request.getHeader("Authorization");          if (authHeader == null || !authHeader.startsWith("Basic ")) {              return ResponseEntity.status(401).body("Missing or invalid Authorization header");          }            String base64Credentials = authHeader.substring("Basic ".length()).trim();          byte[] decodedBytes = Base64.getDecoder().decode(base64Credentials);          String credentials = new String(decodedBytes);          String[] values = credentials.split(":", 2);          String username = values[0];          String password = values[1];            if ("user".equals(username) && "pwd".equals(password)) {              String token = JwtUtil.generateToken(username);              return ResponseEntity.ok("{\"token\":\"" + token + "\"}");          } else {              return ResponseEntity.status(403).body("Invalid credentials");          }      }  } |

4. Configure Security – Disable Default Login

|  |
| --- |
| SecurityConfig.java  package com.cognizant.spring\_learn.config;  import org.springframework.context.annotation.Bean;  import org.springframework.context.annotation.Configuration;  import org.springframework.security.config.annotation.web.builders.HttpSecurity;  import org.springframework.security.web.SecurityFilterChain;  @Configuration  public class SecurityConfig {       @Bean      public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {          http              .csrf().disable()              .authorizeHttpRequests(auth -> auth                  .anyRequest().permitAll()              );          return http.build();      }  } |

**Output:**

****

****