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

**Code:**

**JwtAuthApplication.java:**

package com.cognizant.jwt\_auth;  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class JwtAuthApplication {  
 public static void main(String[] args) {  
 SpringApplication.*run*(JwtAuthApplication.class, args);  
 }  
}

**JwtUtil.java:**

package com.cognizant.jwt\_auth.util;  
import io.jsonwebtoken.Jwts;  
import io.jsonwebtoken.SignatureAlgorithm;  
import org.springframework.stereotype.Component;  
import java.util.Date;  
  
@Component  
public class JwtUtil {  
 private final String SECRET\_KEY = "secret";  
 public String generateToken(String username) {  
 return Jwts.*builder*()  
 .setSubject(username)  
 .setIssuedAt(new Date(System.*currentTimeMillis*()))  
 .setExpiration(new Date(System.*currentTimeMillis*() + 1000 \* 60 \* 60))  
 .signWith(SignatureAlgorithm.*HS256*, SECRET\_KEY)  
 .compact();  
 }  
}

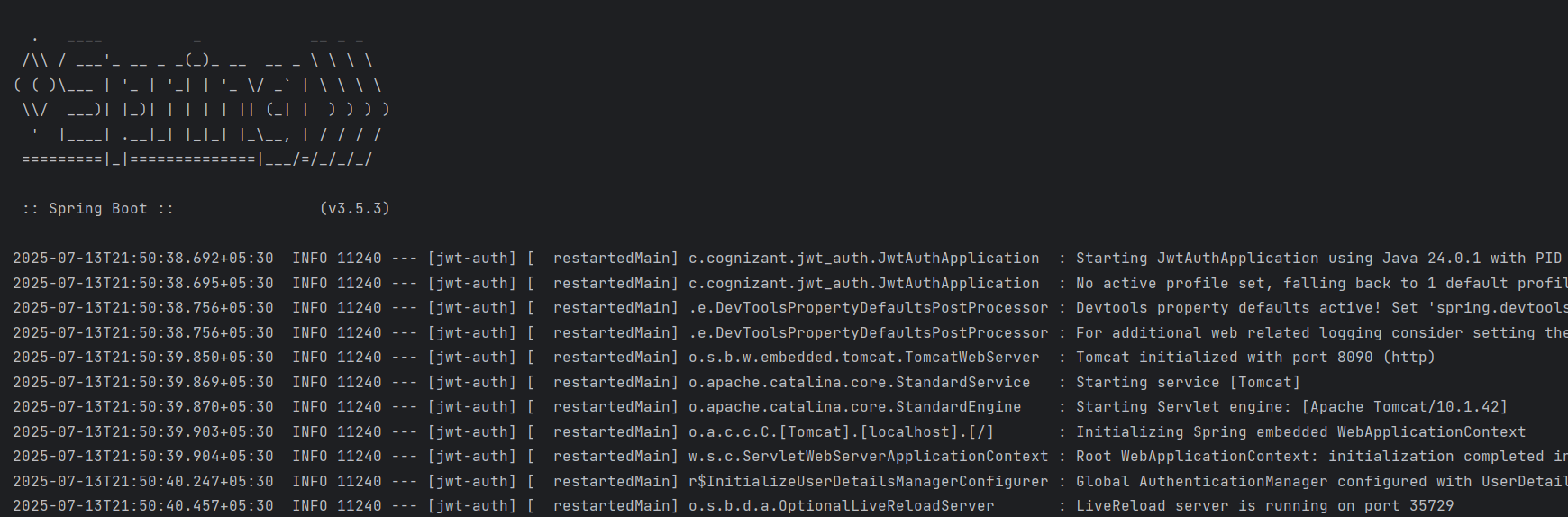
**SecurityConfig.java:**

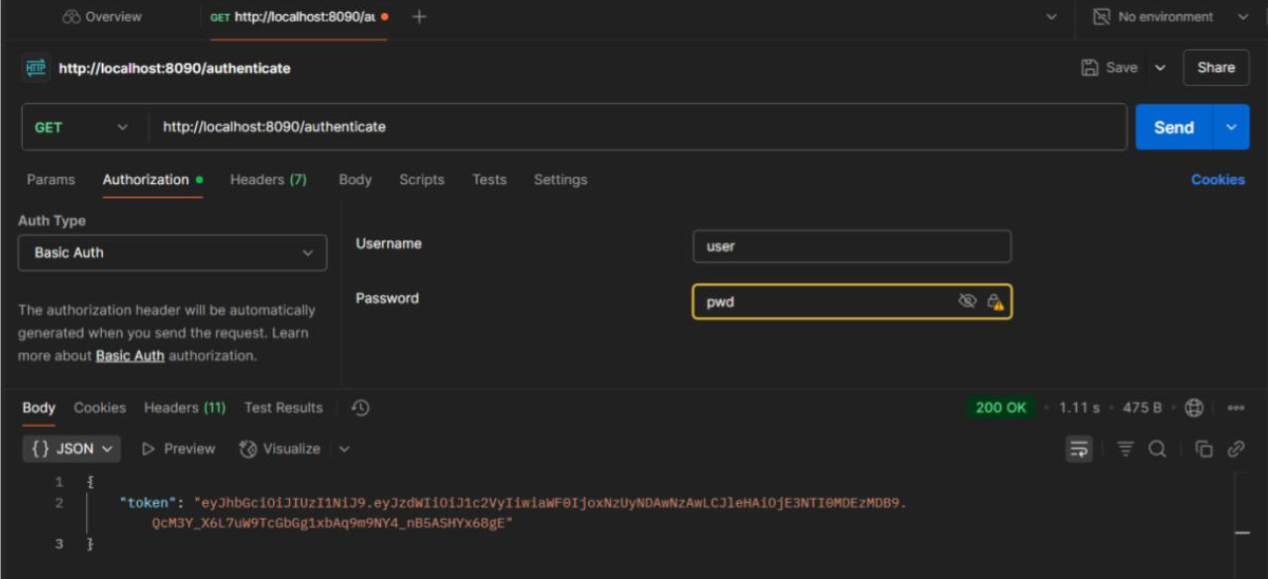
package com.cognizant.jwt\_auth.config;  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.Configuration;  
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;  
import org.springframework.security.core.userdetails.User;  
import org.springframework.security.core.userdetails.UserDetailsService;  
import org.springframework.security.provisioning.InMemoryUserDetailsManager;  
import org.springframework.security.web.SecurityFilterChain;  
import org.springframework.security.config.annotation.web.builders.HttpSecurity;  
import org.springframework.security.config.Customizer;  
  
@Configuration  
@EnableWebSecurity  
public class SecurityConfig {  
 @Bean  
 public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {  
 http  
 .csrf(csrf -> csrf.disable())  
 .authorizeHttpRequests(auth -> auth  
 .requestMatchers("/authenticate").authenticated()  
 .anyRequest().permitAll()  
 )  
 .httpBasic(Customizer.*withDefaults*()); // recommended in Spring Security 6  
 return http.build();  
 }  
 @Bean  
 public UserDetailsService userDetailsService() {  
 return new InMemoryUserDetailsManager(  
 User.*withUsername*("user").password("{noop}pwd").roles("USER").build()  
 );  
 }  
}

**AuthController.java:**

package com.cognizant.jwt\_auth.controller;  
import com.cognizant.jwt\_auth.util.JwtUtil;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.http.ResponseEntity;  
import org.springframework.security.core.Authentication;  
import org.springframework.web.bind.annotation.\*;  
import java.util.Map;  
  
@RestController  
public class AuthController {  
 @Autowired  
 private JwtUtil jwtUtil;  
 @GetMapping("/authenticate")  
 public ResponseEntity<Map<String, String>> authenticate(Authentication authentication) {  
 String username = authentication.getName();  
 String token = jwtUtil.generateToken(username);  
 return ResponseEntity.*ok*(Map.*of*("token", token));  
 }  
}

**Output:**

****

****