**Create authentication service that returns JWT**

**Code:**  
**AutheticationController.java:**

package com.cognizant.controller;

import com.cognizant.security.JwtUtil;

import jakarta.servlet.http.HttpServletRequest;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.\*;

import java.util.Base64;

*@RestController*

public class AuthenticationController {

*@Autowired*

private JwtUtil jwtUtil;

*@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");

}

// Decode credentials

String base64Credentials = authHeader.substring("Basic ".length()).trim();

String credentials = new String(Base64.*getDecoder*().decode(base64Credentials));

String[] values = credentials.split(":", 2);

String username = values[0];

String password = values[1];

// In real app, verify username/password. Here we just accept any input.

if (!username.equals("user") || !password.equals("pwd")) {

return ResponseEntity.*status*(403).body("Invalid credentials");

}

String token = jwtUtil.generateToken(username);

return ResponseEntity.*ok*().body("{\"token\":\"" + token + "\"}");

}

}

**SecurityConfig.java:**

package com.cognizant.security;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.security.web.SecurityFilterChain;

import org.springframework.security.config.annotation.web.builders.HttpSecurity;

*@Configuration*

public class SecurityConfig {

*@Bean*

public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {

http

.csrf(csrf -> csrf.disable())

.authorizeHttpRequests(auth -> auth

.anyRequest().permitAll() //

)

.httpBasic(httpBasic -> httpBasic.disable()); //

return http.build();

}

}

**JwtUtil.java:**

package com.cognizant.security;

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.SignatureAlgorithm;

import io.jsonwebtoken.security.Keys;

import org.springframework.stereotype.Component;

import java.security.Key;

import java.util.Date;

*@Component*

public class JwtUtil {

private final Key key = Keys.*secretKeyFor*(*SignatureAlgorithm*.***HS256***); // auto-generates safe key

public String generateToken(String username) {

return Jwts.*builder*()

.setSubject(username)

.setIssuedAt(new Date(System.*currentTimeMillis*()))

.setExpiration(new Date(System.*currentTimeMillis*() + 1000 \* 60 \* 10))

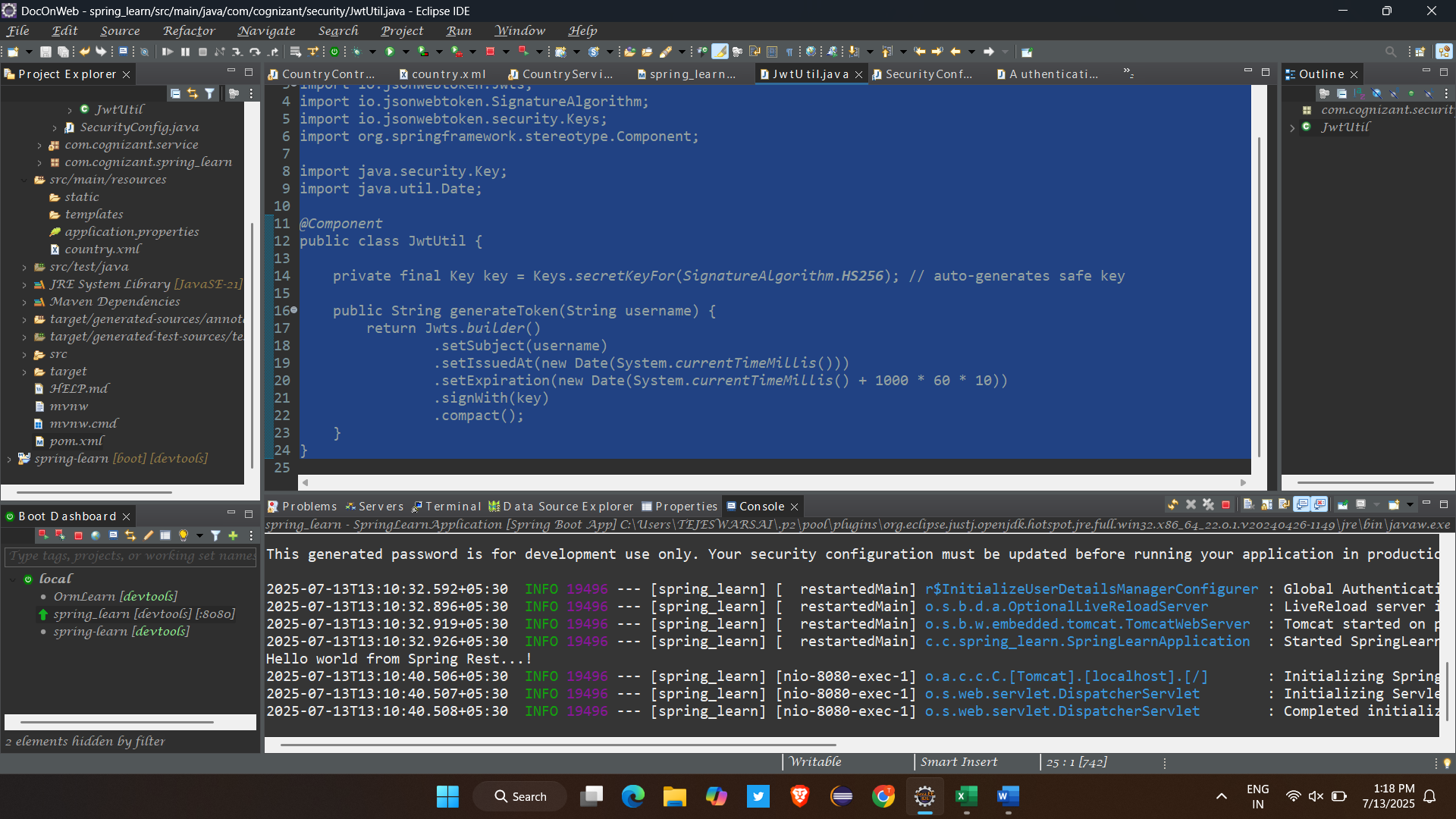
.signWith(key)

.compact();

}

}A screenshot of a computer

AI-generated content may be incorrect.

 A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect. A computer screen shot of a computer program

AI-generated content may be incorrect.