**Title:  
JWT Authentication Service Using Spring Boot**

**Problem Statement:**  
Implement a Spring Boot-based authentication service that generates and returns a JWT when valid user credentials are sent using Basic Authentication. The service should expose an endpoint /authenticate which, upon successful authentication, returns a JWT token in JSON format.

**Steps to Execute via Command Line:**

1. Ensure Java 17 and Maven are installed
2. Extract the project folder jwt-auth-service-fixed
3. Open Command Prompt and navigate to the project directory:
4. cd jwt-auth-service-fixed
5. Build the project:
6. mvn clean install
7. Run the application:
8. mvn spring-boot:run
9. In a new terminal, execute the following curl command:
10. curl -s -u user:pwd http://localhost:8080/authenticate

**Code:**

SecurityConfig.java

package com.cognizant.jwt\_auth\_service.config;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuilder;

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

import org.springframework.security.crypto.password.NoOpPasswordEncoder;

import org.springframework.security.web.SecurityFilterChain;

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

@Configuration

public class SecurityConfig {

@Bean

public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {

http.csrf().disable()

.authorizeHttpRequests()

.requestMatchers("/authenticate").permitAll()

.anyRequest().authenticated()

.and()

.httpBasic();

return http.build();

}

@Bean

public static NoOpPasswordEncoder passwordEncoder() {

return (NoOpPasswordEncoder) NoOpPasswordEncoder.getInstance();

}

@Autowired

public void configureGlobal(AuthenticationManagerBuilder auth) throws Exception {

auth.inMemoryAuthentication()

.withUser("user")

.password("{noop}pwd")

.roles("USER");

}

}

AuthController.java

package com.cognizant.jwt\_auth\_service.controller;

import com.cognizant.jwt\_auth\_service.util.JwtUtil;

import com.cognizant.jwt\_auth\_service.model.JwtResponse;

import org.springframework.http.ResponseEntity;

import org.springframework.security.core.Authentication;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RestController;

@RestController

public class AuthController {

@GetMapping("/authenticate")

public ResponseEntity<JwtResponse> authenticate(Authentication authentication) {

String username = authentication.getName();

String token = JwtUtil.generateToken(username);

return ResponseEntity.ok(new JwtResponse(token));

}

}

JwtUtil.java

package com.cognizant.jwt\_auth\_service.util;

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.SignatureAlgorithm;

import java.util.Date;

public class JwtUtil {

private static final String SECRET\_KEY = "secret";

public static String generateToken(String username) {

return Jwts.builder()

.setSubject(username)

.setIssuedAt(new Date())

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

.signWith(SignatureAlgorithm.HS256, SECRET\_KEY)

.compact();

}

}

JwtResponse.java

package com.cognizant.jwt\_auth\_service.model;

public class JwtResponse {

private String token;

public JwtResponse(String token) {

this.token = token;

}

public String getToken() {

return token;

}

}

JwtAuthServiceApplication.java

package com.cognizant.jwt\_auth\_service;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class JwtAuthServiceApplication {

public static void main(String[] args) {

SpringApplication.run(JwtAuthServiceApplication.class, args);

}

}

**Output:**

