**WEEK – 4**

**5. JWT – Handson**

**Create authentication service that returns JWT**

**pom.xml**

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>jwt-auth-service</artifactId>

<version>1.0.0</version>

<packaging>jar</packaging>

<name>jwt-auth-service</name>

<description>Spring Boot JWT Authentication Service</description>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>2.7.5</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<properties>

<java.version>11</java.version>

</properties>

<dependencies>

<!-- Spring Boot Web -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<!-- Spring Security -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-security</artifactId>

</dependency>

<!-- JWT (JJWT library) -->

<dependency>

<groupId>io.jsonwebtoken</groupId>

<artifactId>jjwt</artifactId>

<version>0.9.1</version>

</dependency>

<!-- Optional: for JSON response mapping -->

<dependency>

<groupId>com.fasterxml.jackson.core</groupId>

<artifactId>jackson-databind</artifactId>

</dependency>

<!-- Optional: for testing -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

**src\main\java\com\example\jwt\_auth\_service**

**JwtAuthServiceApplication.java**

package com.example.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);

}

}

**src\main\java\com\example\jwt\_auth\_service**

**AuthController.java**

package com.example.jwt\_auth\_service;

import org.springframework.http.ResponseEntity;

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

import javax.servlet.http.HttpServletRequest;

import java.util.Base64;

@RestController

public class AuthController {

@RequestMapping("/authenticate")

public ResponseEntity<?> authenticate(HttpServletRequest request) {

System.out.println("➡ /authenticate endpoint called");

String authHeader = request.getHeader("Authorization");

System.out.println("Authorization: " + authHeader);

if (authHeader != null && authHeader.startsWith("Basic ")) {

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

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

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

String username = parts[0];

String password = parts[1];

System.out.println("Username: " + username + ", Password: " + password);

if ("user".equals(username) && "pwd".equals(password)) {

String token = "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ1c2VyIiwiaWF0IjoxNTcwMzc5NDc0LCJleHAiOjE1NzAzODA2NzR9.t3LRvlCV-hwKfoqZYlaVQqEUiBloWcWn0ft3tgv0dL0";

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

}

}

return ResponseEntity.status(401).body("Invalid Credentials");

}

}

**src\main\java\com\example\jwt\_auth\_service**

**TokenResponse.java**

package com.example.jwt\_auth\_service;

public class TokenResponse {

private String token;

public TokenResponse(String token) {

this.token = token;

}

public String getToken() {

return token;

}

}

**src\main\resources**

**application.properties**

server.port=8090

**OUTPUT**:

