**Exercise 1: Configuring a Basic Spring Application**

CODE:

package com.example.week3;

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

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

*@RestController*

public class BasicApplication {

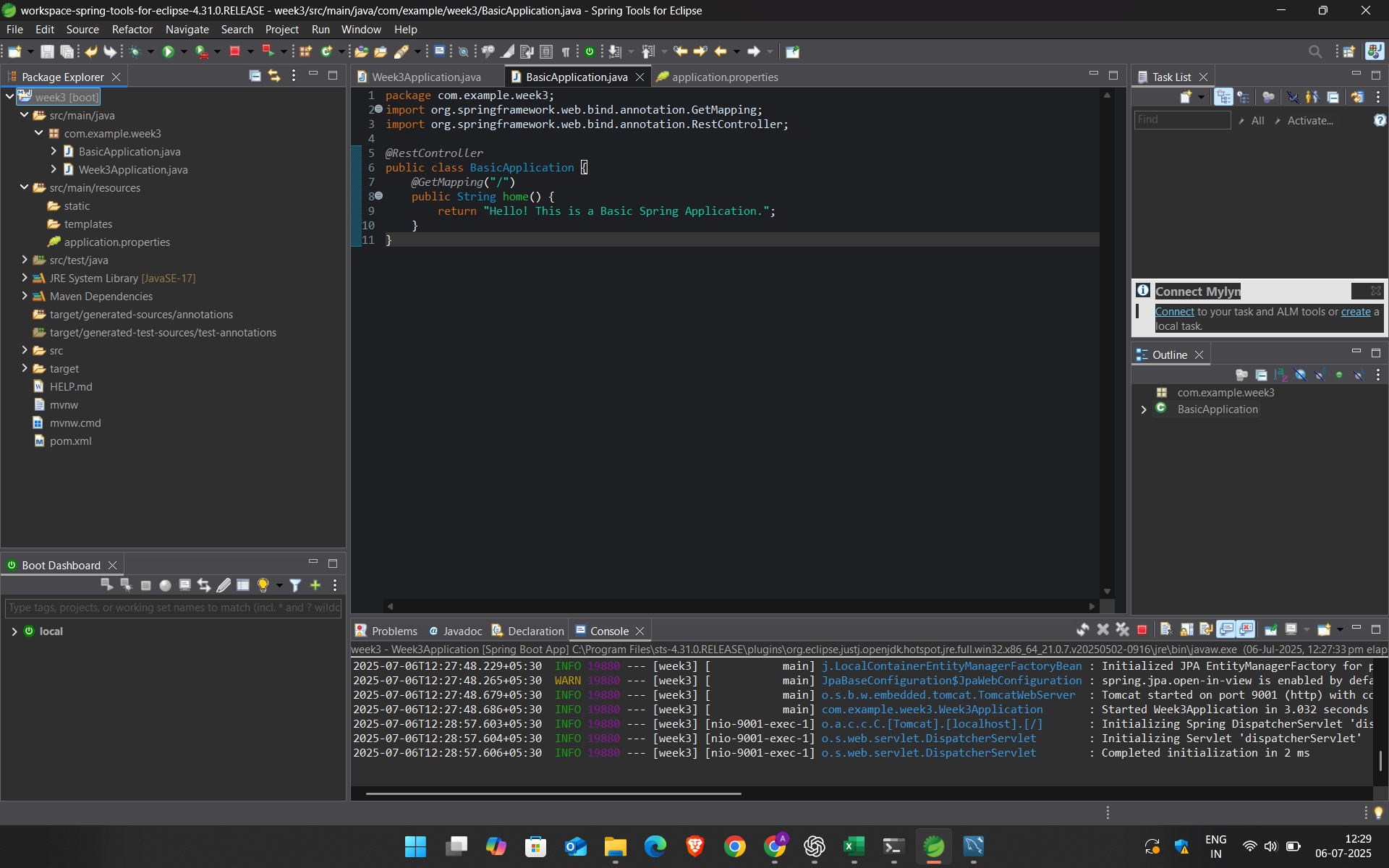
*@GetMapping*("/")

public String home() {

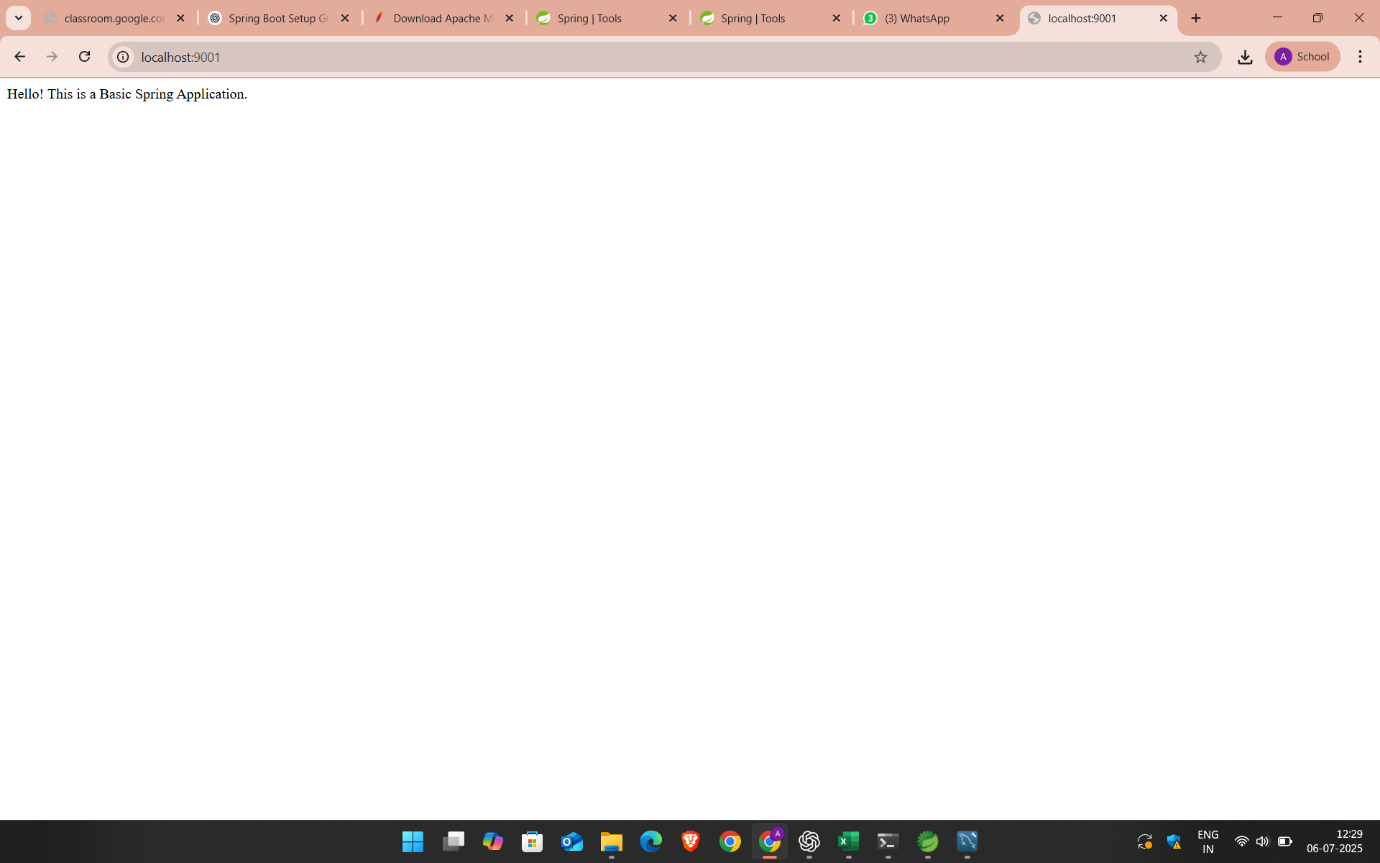
return "Hello! This is a Basic Spring Application.";

}

}



OUTPUT:



**EXERCISE 2:IMPLEMENTING DEPENDENCY INJECTION**

CODE:

package com.example.week3;

import org.springframework.stereotype.Service;

*@Service* // Marks this class as a Spring-managed component

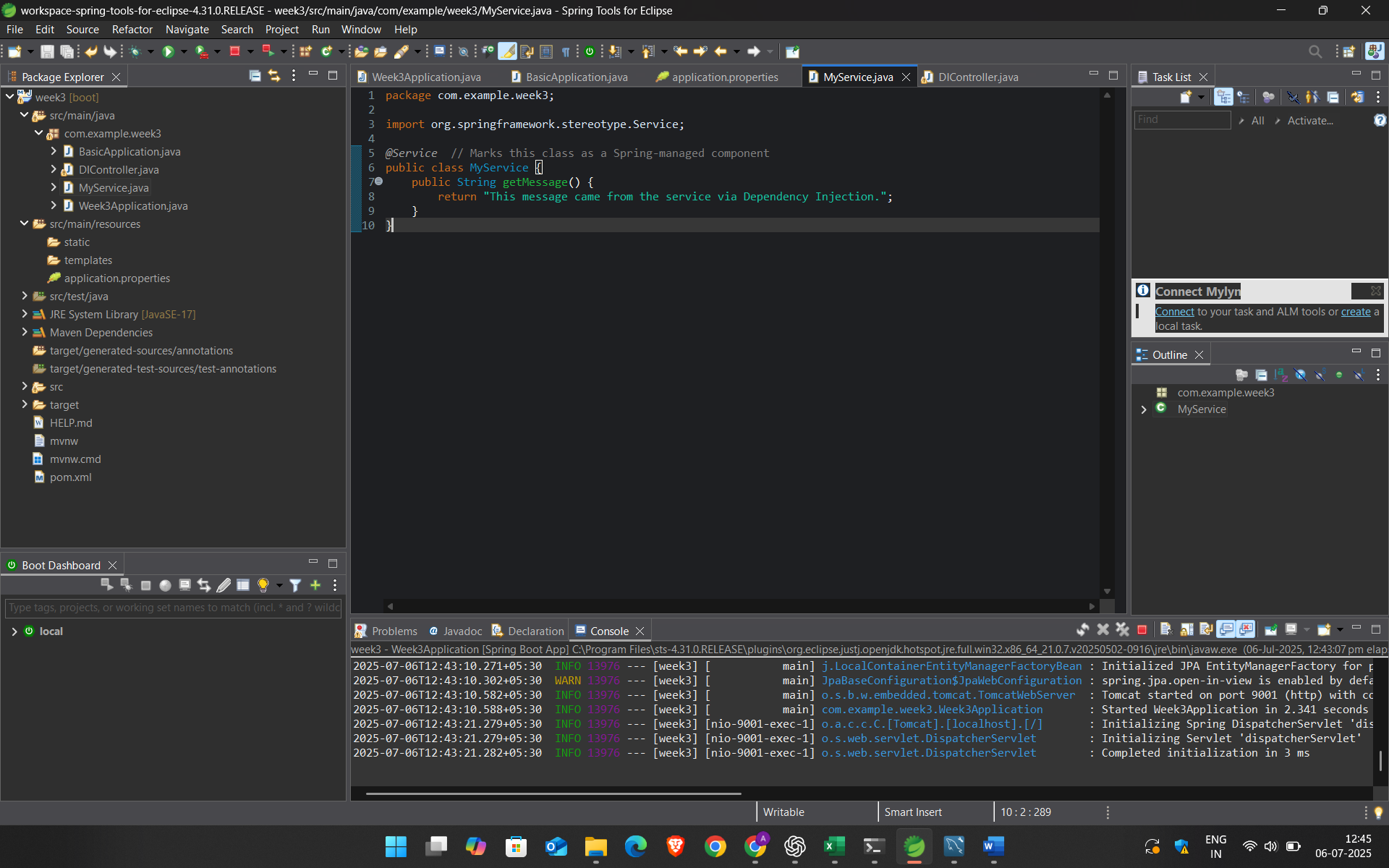
public class MyService {

public String getMessage() {

return "This message came from the service via Dependency Injection.";

}

}



CODE:

package com.example.week3;

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

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

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

*@RestController*

public class DIController {

private final MyService myService;

// Constructor-based Injection

*@Autowired*

public DIController(MyService myService) {

this.myService = myService;

}

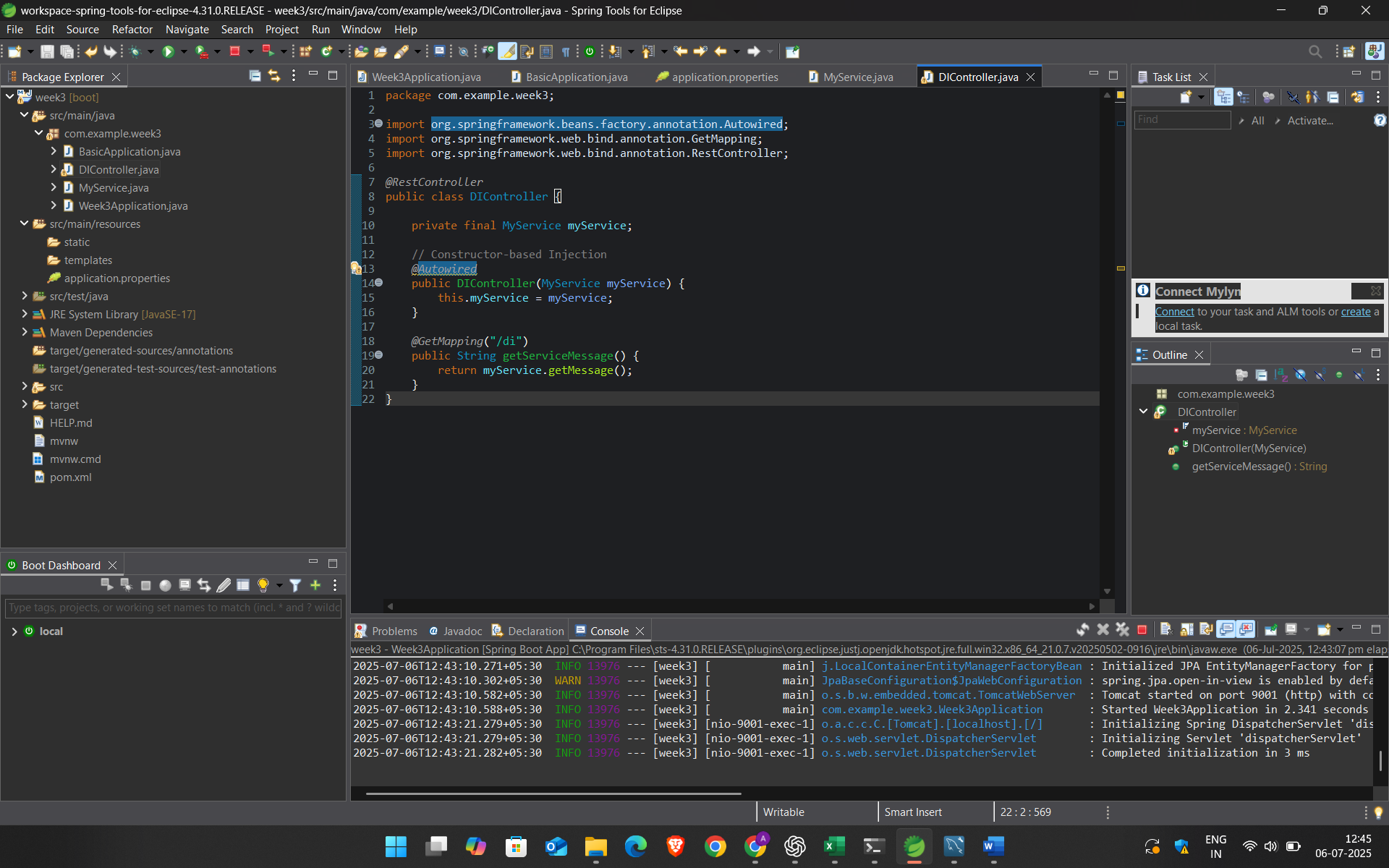
*@GetMapping*("/di")

public String getServiceMessage() {

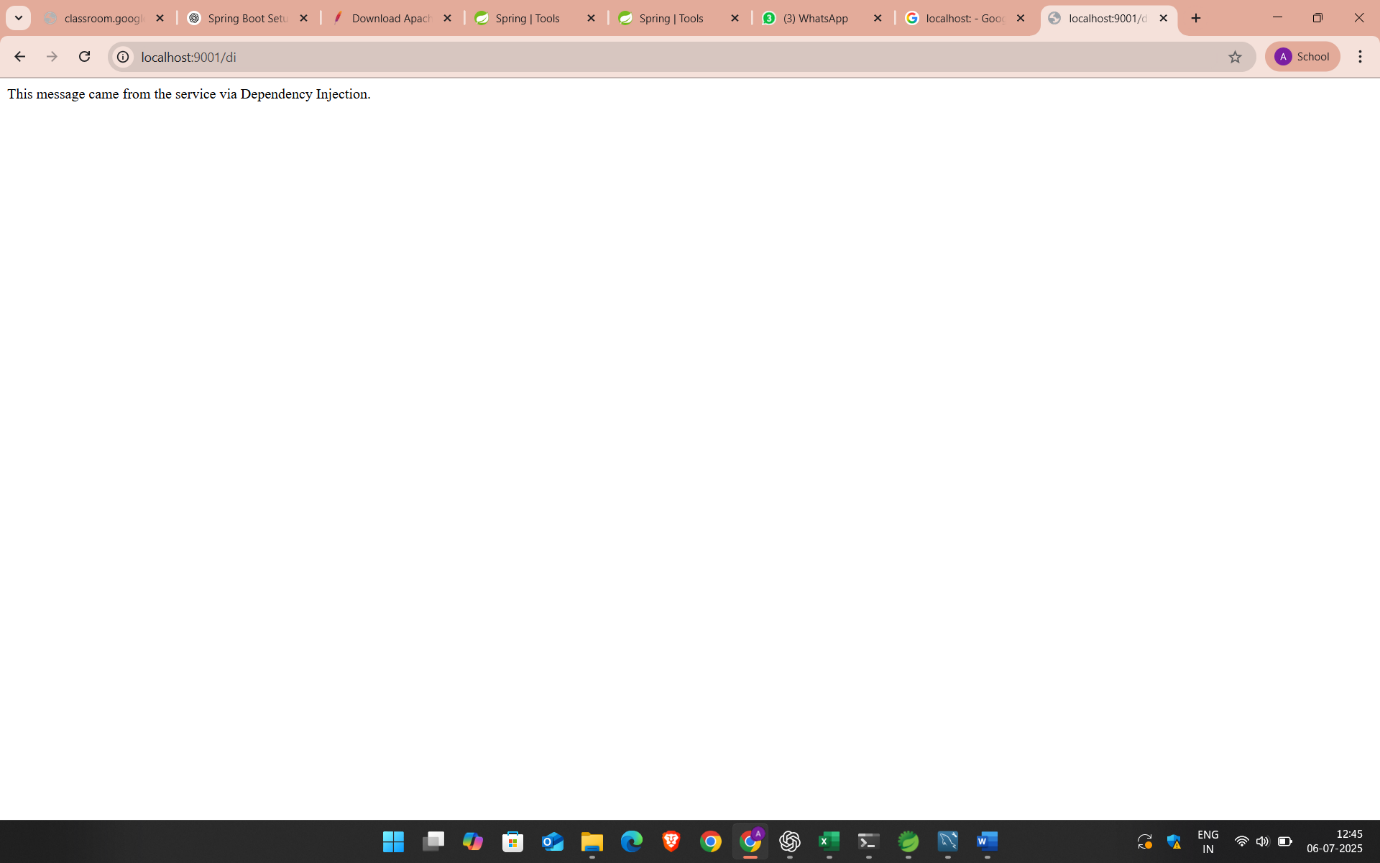
return myService.getMessage();

}

}



OUTPUT:



EXERCISE3:CRERATING AMND CONFIGURING THE MAVEN PROJECT

CODE:

<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 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>mavenexample</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>MavenExample</name>

<description>A basic Spring Boot Maven project</description>

<parent>

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

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

<version>3.1.5</version>

<relativePath/>

</parent>

<properties>

<java.version>17</java.version>

</properties>

<dependencies>

<!-- Spring Boot Web Starter -->

<dependency>

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

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

</dependency>

<!-- Spring Boot Test (for unit testing) -->

<dependency>

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

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

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<!-- Spring Boot Plugin -->

<plugin>

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

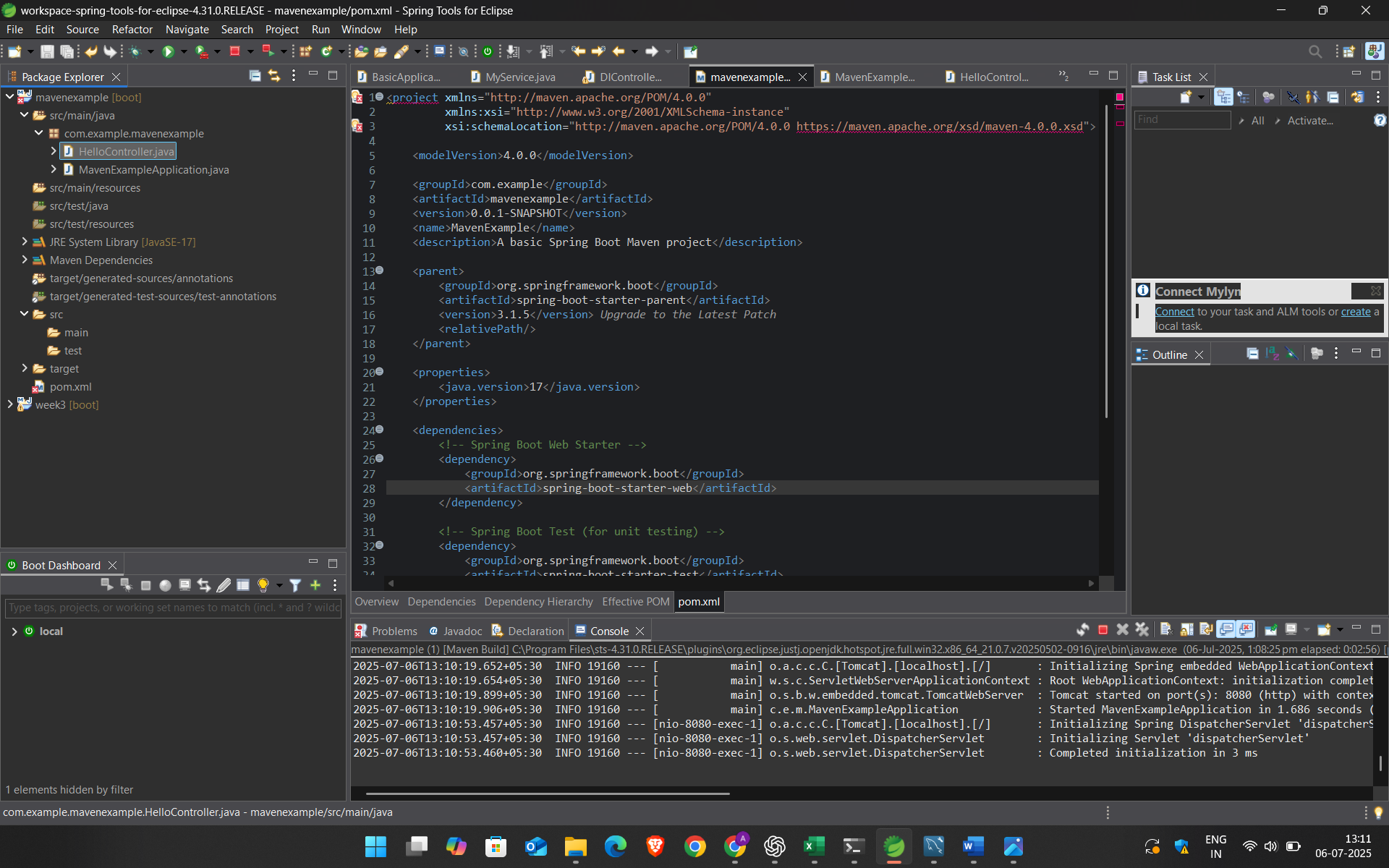
<artifactId>spring-boot-maven-plugin</artifactId>

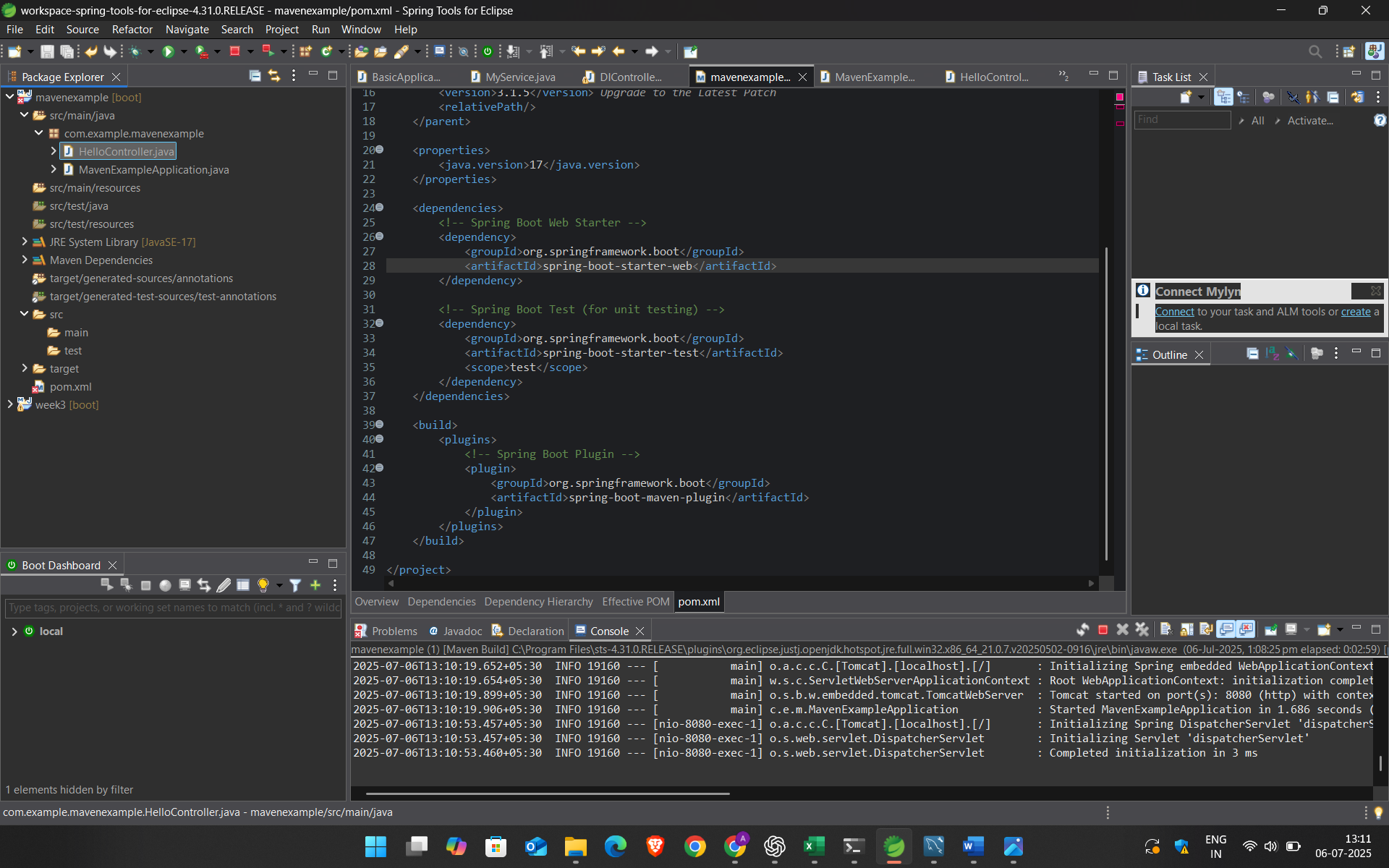
</plugin>

</plugins>

</build>

</project>





CODE:

package com.example.mavenexample;

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

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

*@RestController*

public class HelloController {

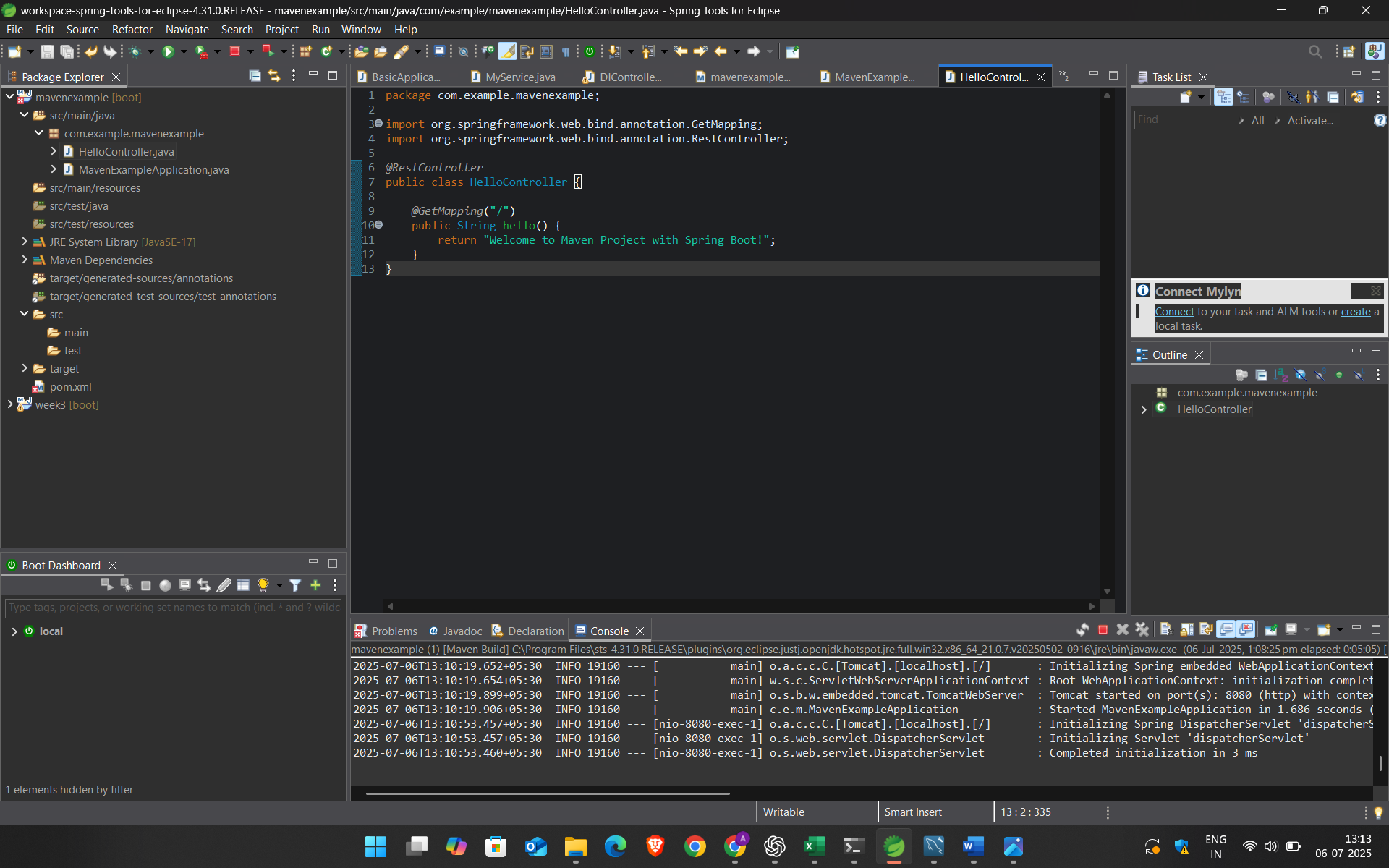
*@GetMapping*("/")

public String hello() {

return "Welcome to Maven Project with Spring Boot!";

}

}



CODE:

package com.example.mavenexample;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

*@*SpringBootApplication

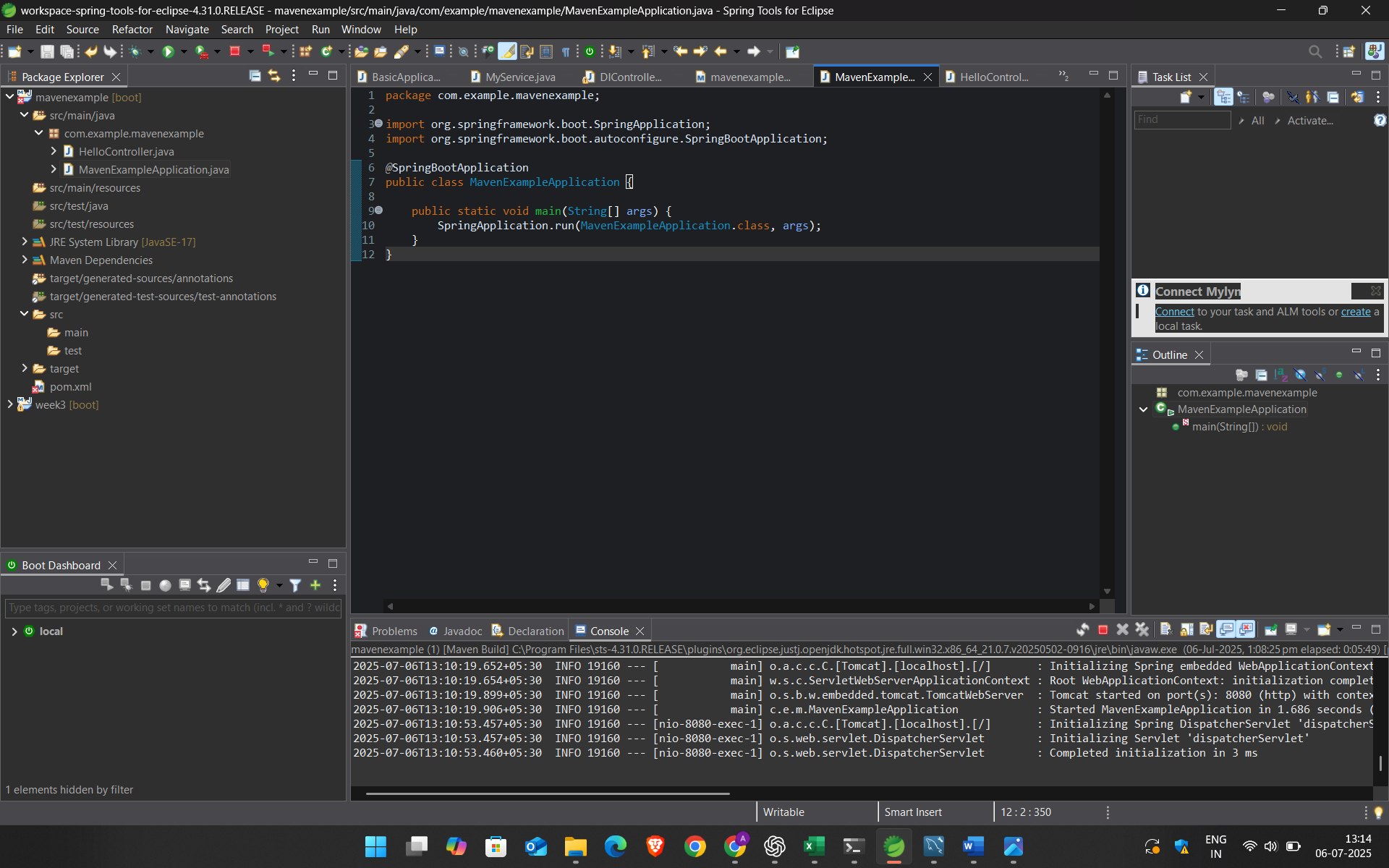
public class MavenExampleApplication {

public static void main(String[] args) {

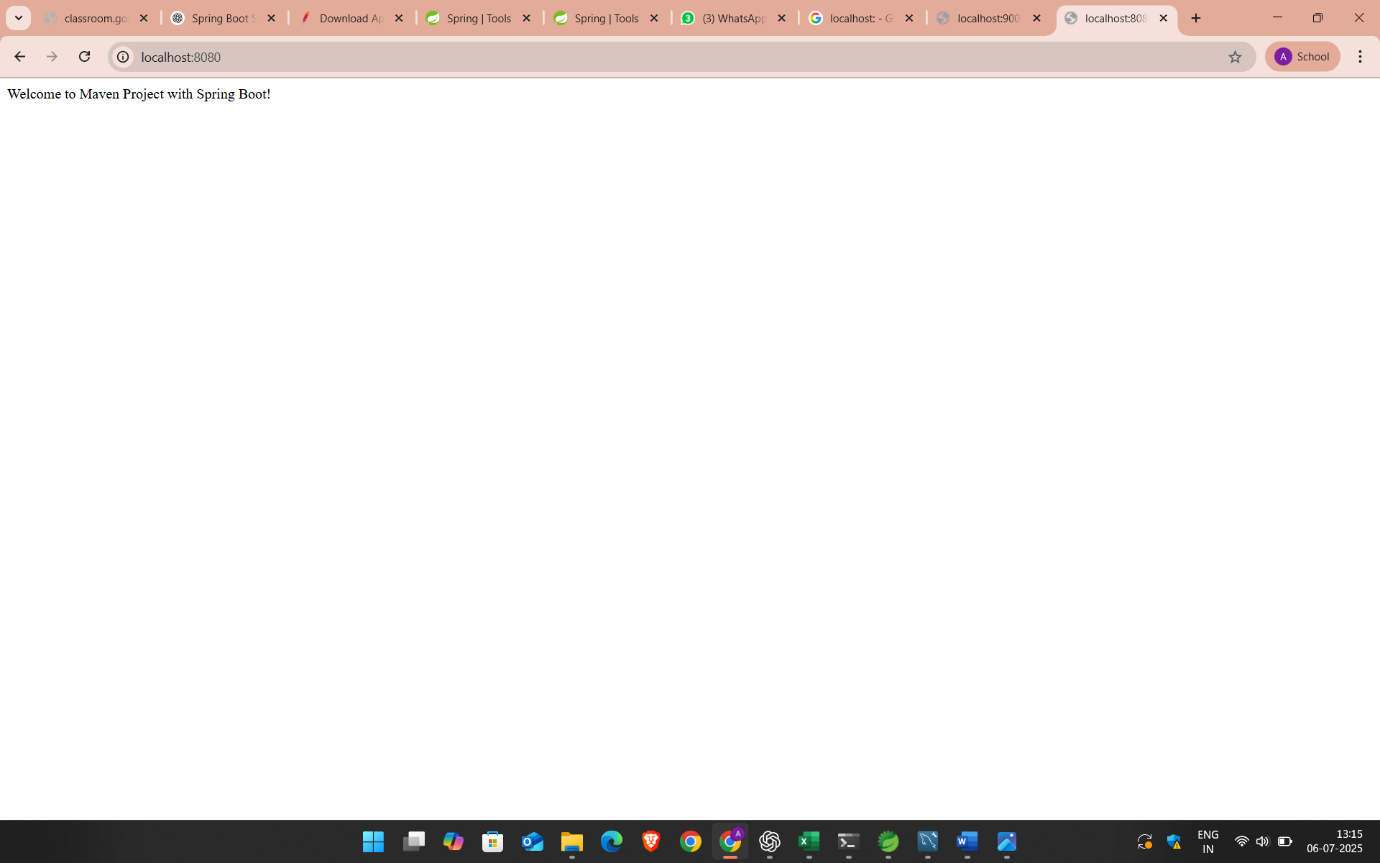
SpringApplication.run(MavenExampleApplication.class, args);

}

}



OUT[PUT:



**EXERCISE 4: Spring Data JPA - Quick Example**

**CODE:**

**CONTROLLER:**

**package com.example.demoproject.UserController;**

**import java.util.List;**

**import java.util.Optional;**

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

**import org.springframework.http.HttpStatus;**

**import org.springframework.http.ResponseEntity;**

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

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

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

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

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

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

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

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

**import com.example.demoproject.UserInterface.UserInterface;**

**import com.example.demoproject.UserModel.UserModel;**

**@RestController**

**@RequestMapping("/demo")**

**public class UserController {**

**@Autowired**

**private UserInterface userInterface;**

**@GetMapping("/getuser")**

**List<UserModel> getAllUsers(){**

**return userInterface.findAll();**

**}**

**@PostMapping("/adduser")**

**UserModel newuser(@RequestBody UserModel newuser) {**

**return userInterface.save(newuser);**

**}**

**@PutMapping("/updateuser/{id}")**

**public ResponseEntity<UserModel> updateUser(@PathVariable Long id,@RequestBody UserModel updateUser){**

**Optional<UserModel> userdata=userInterface.findById(id);**

**if(userdata.isPresent())**

**{**

**UserModel user=userdata.get();**

**user.setName(updateUser.getName());**

**user.setIncome(updateUser.getIncome());**

**userInterface.save(user);**

**return new ResponseEntity<>(HttpStatus.OK);**

**}**

**else**

**{**

**return new ResponseEntity<>(HttpStatus.NOT\_FOUND);**

**}**

**}**

**@DeleteMapping("/deleteuser/{id}")**

**public ResponseEntity<HttpStatus> deleteuser (@PathVariable Long id){**

**try {**

**userInterface.deleteById(id);**

**return new ResponseEntity<>(HttpStatus.NO\_CONTENT);**

**}**

**catch(Exception e) {**

**return new ResponseEntity<>(HttpStatus.INTERNAL\_SERVER\_ERROR);**

**}**

**}**

**@GetMapping("And/{name}/{income}")**

**public List<UserModel> getByNameAndIncome(@PathVariable String name,@PathVariable Double income){**

**return userInterface.findByNameAndIncome(name,income);**

**}**

**}**

**INTERFACE:**

**package com.example.demoproject.UserInterface;**

**import java.util.List;**

**import org.springframework.data.jpa.repository.JpaRepository;**

**import com.example.demoproject.UserModel.UserModel;**

**public interface UserInterface extends JpaRepository<UserModel, Long> {**

**List<UserModel>findAll();**

**List<UserModel> findByNameAndIncome(String name, Double income);**

**}**

**MODEL:**

**package com.example.demoproject.UserModel;**

**import jakarta.persistence.Entity;**

**import jakarta.persistence.GeneratedValue;**

**import jakarta.persistence.Id;**

**@Entity**

**public class UserModel {**

**@Id**

**@GeneratedValue**

**private Long id;**

**private String name;**

**private Double income;**

**public Long getId() {**

**return id;**

**}**

**public void setId(Long id) {**

**this.id = id;**

**}**

**public String getName() {**

**return name;**

**}**

**public void setName(String name) {**

**this.name = name;**

**}**

**public Double getIncome() {**

**return income;**

**}**

**public void setIncome(Double income) {**

**this.income = income;**

**}**

**}**

OUTPUT:

