**UNIT-3(EXP-3)**

**1. Entity: Product.java**

**package** com.example.demo;

**import** jakarta.persistence.Entity;

**import** jakarta.persistence.GeneratedValue;

**import** jakarta.persistence.GenerationType;

**import** jakarta.persistence.Id;

@Entity

**public** **class** Product {

@Id

@GeneratedValue(strategy = GenerationType.***IDENTITY***)

**private** **long** id;

**private** String name;

**private** **double** price;

**public** Product() {}

**public** Product(String name, **double** price) {

**this**.name = name;

**this**.price = price;

}

// getters & setters

**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** getPrice() { **return** price; }

**public** **void** setPrice(**double** price) { **this**.price = price; }

}

**2. Repository: ProductRepository.java**

**package** com.example.demo;

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

**public** **interface** ProductRepository **extends** JpaRepository<Product, Long> {

}

**3. Service: ProductService.java**

**package** com.example.demo;

**import** org.springframework.stereotype.Service;

**import** java.util.List;

@Service

**public** **class** ProductService {

**private** **final** ProductRepository repo;

**public** ProductService(ProductRepository repo) {

**this**.repo = repo;

}

**public** Product saveProduct(Product product) {

**return** repo.save(product);

}

**public** List<Product> getAllProducts() {

**return** repo.findAll();

}

}

**4. REST Controller: ProductController.java**

**package** com.example.demo;

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

**import** java.util.List;

@RestController

@RequestMapping("/products")

**public** **class** ProductController {

**private** **final** ProductService service;

**public** ProductController(ProductService service) {

**this**.service = service;

}

@PostMapping("/add")

**public** Product addProduct(@RequestBody Product product) {

**return** service.saveProduct(product);

}

@GetMapping("/all")

**public** List<Product> getAllProducts() {

**return** service.getAllProducts();

}

}

**5. Logging Aspect: LoggingAspect.java**

**package** com.example.demo;

**import** org.aspectj.lang.JoinPoint;

**import** org.aspectj.lang.annotation.Aspect;

**import** org.aspectj.lang.annotation.Before;

**import** org.springframework.stereotype.Component;

@Aspect

@Component

**public** **class** LoggingAspect {

// Logs before executing any ProductService method

@Before("execution(\* com.example.demo.ProductService.\*(..))")

**public** **void** logBefore(JoinPoint joinPoint) {

System.***out***.println(">>> Entering method: " + joinPoint.getSignature().getName());

}

}

**6. application.properties (using H2 in-memory DB for simplicity)**

spring.application.name=product

spring.datasource.url=jdbc:mysql://localhost:3306/mca

spring.datasource.username=root

spring.datasource.password=naga@1331

spring.jpa.hibernate.ddl-auto=create-drop

spring.jpa.show-sql=true

server.port=8888

**7.pom.xml**

<?xml version="1.0" encoding="UTF-8"?>

<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>

<parent>

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

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

<version>3.5.4</version>

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

</parent>

<groupId>com</groupId>

<artifactId>productApplication</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>product</name>

<description>Demo project for Spring Boot</description>

<url/>

<licenses>

<license/>

</licenses>

<developers>

<developer/>

</developers>

<scm>

<connection/>

<developerConnection/>

<tag/>

<url/>

</scm>

<properties>

<java.version>21</java.version>

</properties>

<dependencies>

<dependency>

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

<artifactId>spring-boot-starter-data-jdbc</artifactId>

</dependency>

<dependency>

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

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

</dependency>

<dependency>

<groupId>com.mysql</groupId>

<artifactId>mysql-connector-j</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

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

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

<scope>test</scope>

</dependency>

<!-- Spring AOP -->

<dependency>

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

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

</dependency>

<!-- Lombok (optional, just to reduce boilerplate) -->

<dependency>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

<optional>true</optional>

</dependency>

<!-- Spring Boot Starter Data JPA -->

<dependency>

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

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<!-- H2 Database (in-memory, no need for MySQL setup) -->

<dependency>

<groupId>com.h2database</groupId>

<artifactId>h2</artifactId>

<scope>runtime</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

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

</plugin>

</plugins>

</build>

</project>

**8.main.java**

**package** com.example.demo;

**import** org.springframework.boot.CommandLineRunner;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

**import** org.springframework.context.annotation.Bean;

@SpringBootApplication

**public** **class** ProductApplication {

**public** **static** **void** main(String[] args) {

SpringApplication.*run*(ProductApplication.**class**, args);

}

@Bean

CommandLineRunner runner(ProductRepository repo) {

**return** args -> {

repo.save(**new** Product("Laptop", 55000));

repo.save(**new** Product("Mobile", 20000));

repo.save(**new** Product("Tablet", 30000));

repo.save(**new** Product("Mouse", 35000));

};

}

}

**7. Example Run**

GET http://localhost:8888/products/all