**Week 3:**

**Spring Core and Maven:**

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

**pom.xml:**

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.29</version>

</dependency>

</dependencies>

**Create a MessageService class**:

public class MessageService {

public void printMessage() {

System.out.println("Hello from Spring!");

}

}

**Create a Spring configuration XML (beans.xml):**

<beans xmlns="http://www.springframework.org/schema/beans"

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

xsi:schemaLocation="

http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="messageService" class="com.example.MessageService" />

</beans>

**Main class**:

public class MainApp {

public static void main(String[] args) {

ApplicationContext context = new ClassPathXmlApplicationContext("beans.xml");

MessageService service = context.getBean("messageService", MessageService.class);

service.printMessage();

}

}

**OUTPUT:**



**Exercise 2: Implementing Dependency Injection**

**Create a service class:**

public class GreetingService {

public String greet() {

return "Hello, Mahalakshmi!";

}

}

**Create a client class**:

public class User {

private GreetingService greetingService;

// setter injection

public void setGreetingService(GreetingService greetingService) {

this.greetingService = greetingService;

}

public void showMessage() {

System.out.println(greetingService.greet());

}

}

**beans.xml(update):**

<bean id="greetingService" class="com.example.GreetingService" />

<bean id="user" class="com.example.User">

<property name="greetingService" ref="greetingService"/>

</bean>

**Main class**:

User user = context.getBean("user", User.class);

user.showMessage();

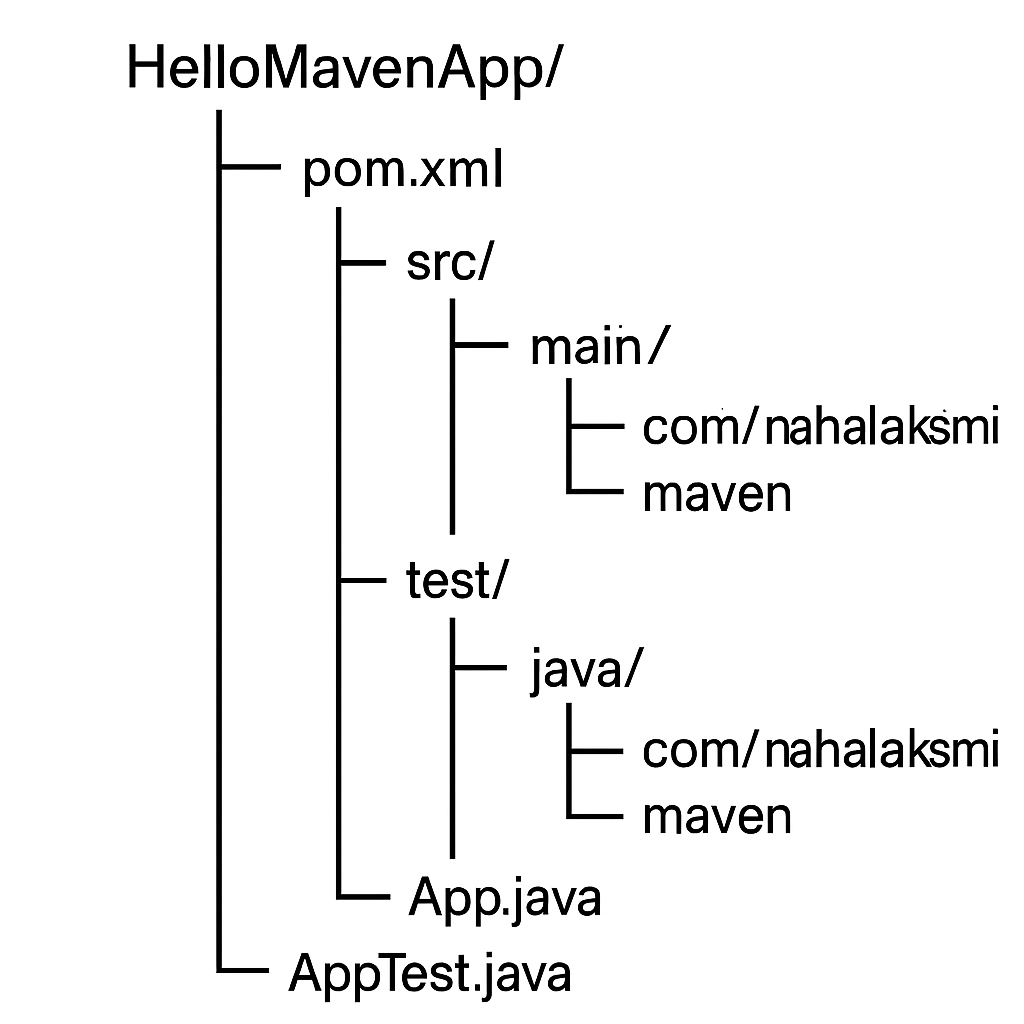
**OUTPUT:**

****

**Exercise 4: Creating and Configuring a Maven Project**

**Maven project Structure:**

**Explore and Modify pom.xml:**

****

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

</dependencies>

**App.java:**

package com.mahalakshmi.maven;

public class App {

public static void main(String[] args) {

System.out.println("Hello from Maven Project!");

}

}

**Build and Run the Project:**

mvn compile

mvn exec:java -Dexec.mainClass="com.mahalakshmi.maven.App"

**Spring Data JPA with Spring Boot, Hibernate:**

**Spring Data JPA - Quick Example**

**application.properties:**

spring.datasource.url=jdbc:h2:mem:testdb

spring.jpa.show-sql=true

spring.jpa.hibernate.ddl-auto=update

**Entity class**:

@Entity

public class Product {

@Id

@GeneratedValue

private Long id;

private String name;}

**Repository Interface**:

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

**Rest Controller**:

@RestController

@RequestMapping("/products")

public class ProductController {

@Autowired

private ProductRepository repo;

@PostMapping

public Product add(@RequestBody Product product) {

return repo.save(product);

}

@GetMapping

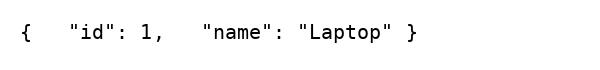
public List<Product> list() {

return repo.findAll();

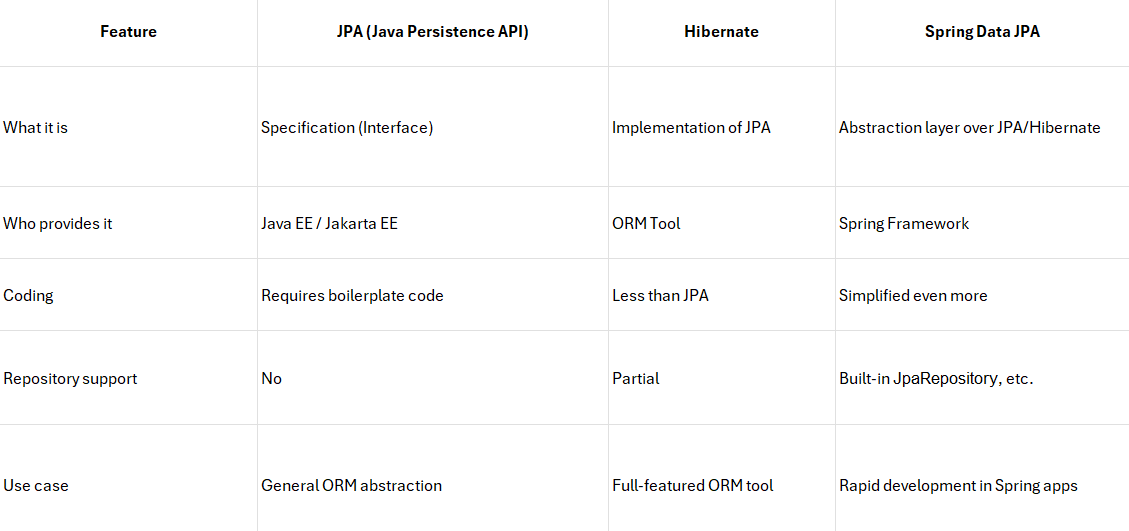
}

}

**OUTPUT:**

****

**Difference Between JPA, Hibernate and Spring Data JPA:**

****