**Hands on 1**

**Create a Spring Web Project using Maven**

1. src/main/java – Folder with application code

**SpringLearnApplication.java**

1. package com.cognizant.spring\_learn;
2. import org.springframework.boot.SpringApplication;
3. import org.springframework.boot.autoconfigure.SpringBootApplication;
4. @SpringBootApplication
5. public class SpringLearnApplication {
6. public static void main(String[] args) {
7. System.out.println("SpringLearnApplication main() started");
8. SpringApplication.run(SpringLearnApplication.class, args);
9. System.out.println("SpringLearnApplication main() completed");
10. }
11. }

2. src/main/resources - Folder for application configuration

**application.properties**

spring.application.name=spring-learn

3. src/test /java – Folder with code for testing the application

**SpringLearnApplicationTests.java**

package com.cognizant.spring\_learn;

import org.junit.jupiter.api.Test;

import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest

class SpringLearnApplicationTests {

@Test

void contextLoads() {

}

}

4. SpringLearnApplication.java – Walkthrough the main() method.

**SpringLearnApplication.java**

package com.cognizant.spring\_learn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearnApplication {

public static void main(String[] args) {

System.out.println("SpringLearnApplication main() started");

SpringApplication.run(SpringLearnApplication.class, args);

System.out.println("SpringLearnApplication main() completed");

}

}

5. Purpose of @SpringBootApplication annotation

- The **@SpringBootApplication** annotation that is used to mark the main class of a Spring Boot Application.

- It combines :

@Configuration

@EnableAutoConfiguration

@ComponentScan

- Sets up the Spring Context.

- Enables component scanning.

- Auto-configures beans.

6. pom.xml

(a) Walkthrough all the configuration defined in XML file.

<?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>3.5.3</version>

<relativePath/>

</parent>

<groupId>com.cognizant</groupId>

<artifactId>spring-learn</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>spring-learn</name>

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

<url/>

<licenses>

<license/>

</licenses>

<developers>

<developer/>

</developers>

<scm>

<connection/>

<developerConnection/>

<tag/>

<url/>

</scm>

<properties>

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

</properties>

<dependencies>

<dependency>

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

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

</dependency>

<dependency>

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

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<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>

(b) Open ‘Dependecy Hierarchy’ and show the dependency tree.

A screen shot of a computer screen

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

**Output**

* **./mvnw spring-boot:run**

A computer screen shot of a program

AI-generated content may be incorrect.