**1.CREATE A SPRING WEB PROJECT USING MAVEN**

CONTEXT OF THE EXAMPLE:

In this exercise, I created a basic Spring Boot web project using Maven. I added essential dependencies (Spring Web, DevTools), configured the main application class, and verified successful setup using logging. The output confirms that the application runs as expected.

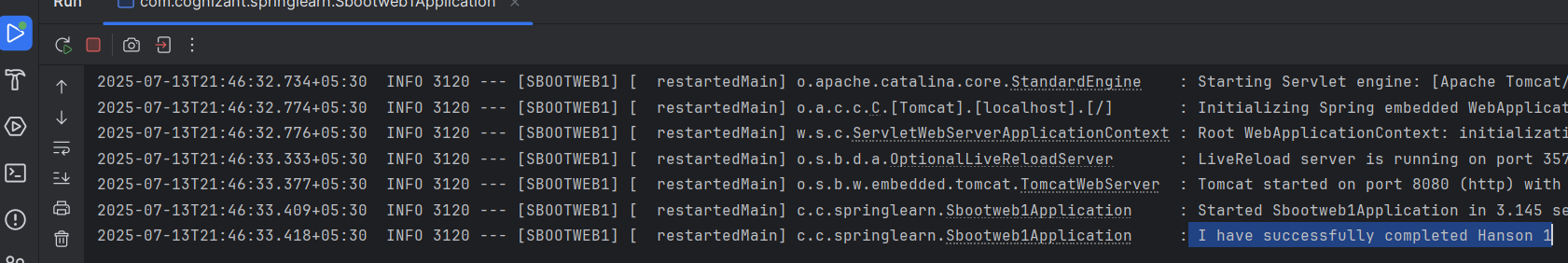
pom.xml

The pom.xml file includes dependencies for **Spring Boot Web** and **DevTools**. These enable auto-configuration for web applications and allow hot reloading during development. No additional manual configurations are required in this case.

Sbootweb1Application.java

package com.dinesh.sbootweb1; // Or your actual package name  
  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class Sbootweb1Application {  
  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(Sbootweb1Application.class);  
  
 public static void main(String[] args) {  
 SpringApplication.*run*(Sbootweb1Application.class, args);  
 *LOGGER*.info("I have successfully completed Hanson 1");  
 }  
}

OUTPUT:



**2. SPRING CORE – LOAD SIMPLEDATEFORMAT FROM SPRING CONFIGURATION XML**

CONTEXT OF THE EXAMPLE:

This example demonstrates loading a SimpleDateFormat bean from Spring's XML configuration and parsing a date string into a Date object using dependency injection.

pom.xml

OUTPUT: