**SLF4J logging framework**

**Exercise 1: Logging Error Messages and Warning Levels**

**Step 1: Update pom.xml with SLF4J and Logback**

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>1.7.30</version>

</dependency>

<dependency>

<groupId>ch.qos.logback</groupId>

<artifactId>logback-classic</artifactId>

<version>1.2.3</version>

</dependency>

**Step 2: Create Your Logging Class**

**LoggingExample.java**

package org.example;  
  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
  
public class LoggingExample {

private static final Logger *logger* = LoggerFactory.*getLogger*(LoggingExample.class);  
  
 public static void main(String[] args) {  
 *logger*.error("This is an error message");

*logger*.warn("This is a warning message");

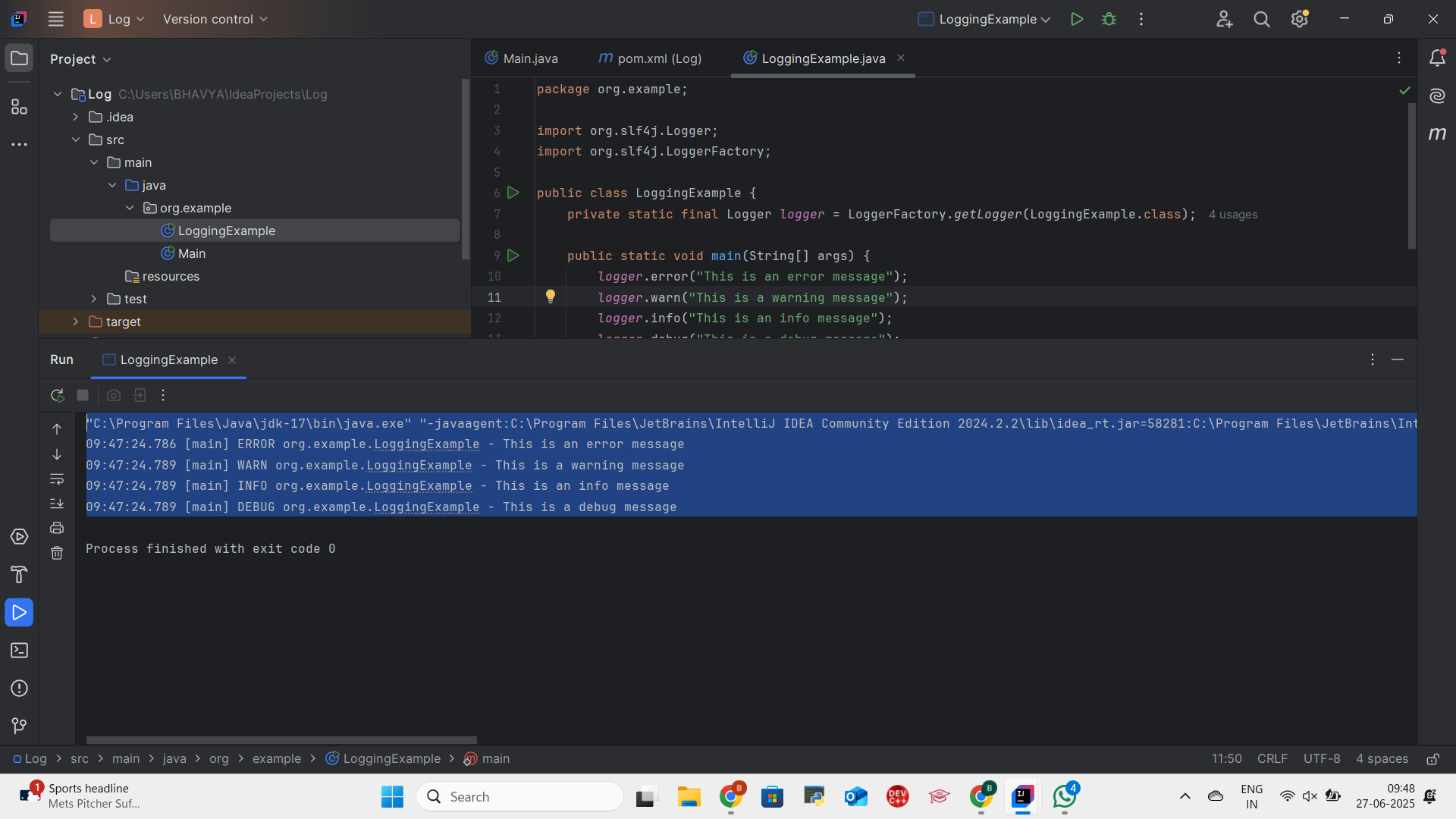
*logger*.info("This is an info message");

*logger*.debug("This is a debug message");  
 }  
]

**Step 3: Run the Program**

1. Right-click LoggingExample.java
2. Click Run 'LoggingExample'

**OUTPUT SCREENSHOT:**

****

**In this exercise, I demonstrated how to implement logging in a Java application using the SLF4J API with Logback as the logging backend. The program successfully logged messages at different severity levels such as ERROR, WARN, INFO, and DEBUG, helping to monitor and debug the application effectively. This practice highlighted the importance of structured logging in real-world Java applications for better traceability and maintenance.**