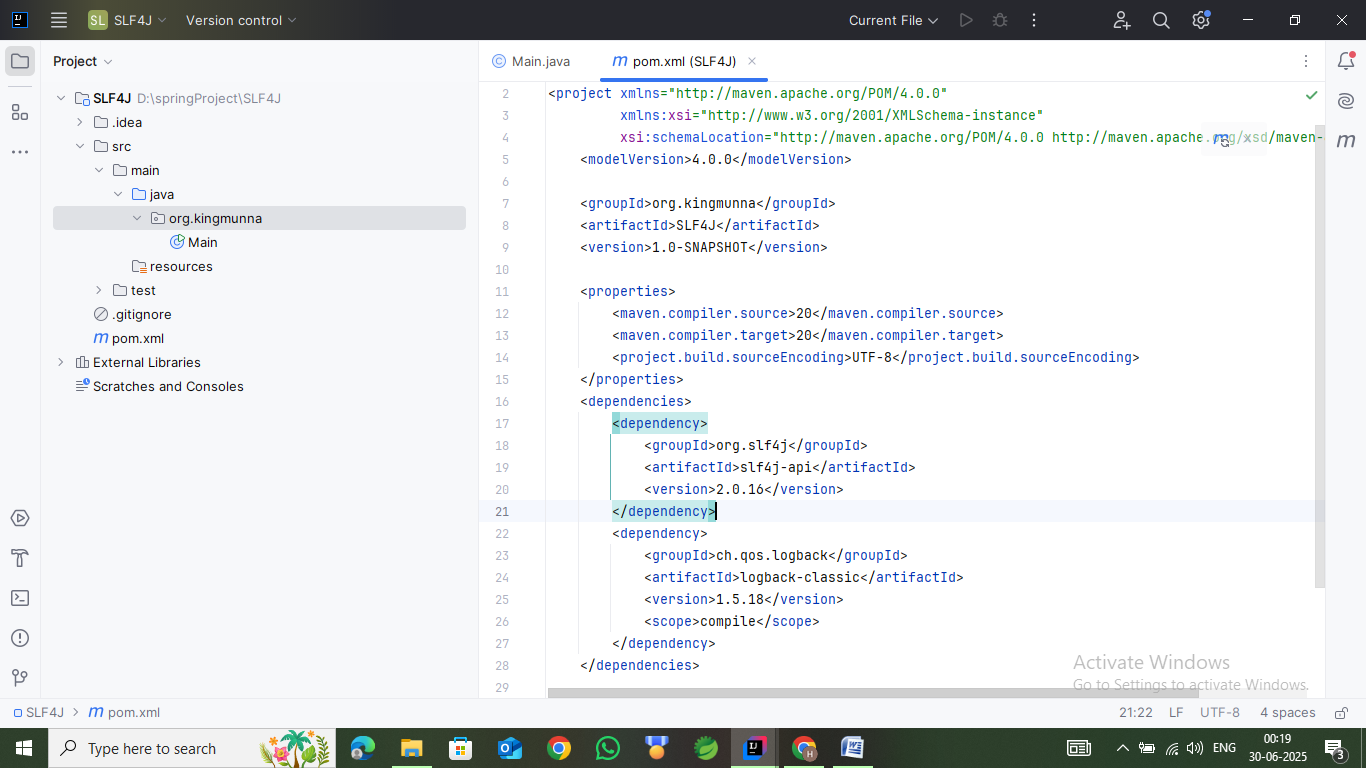
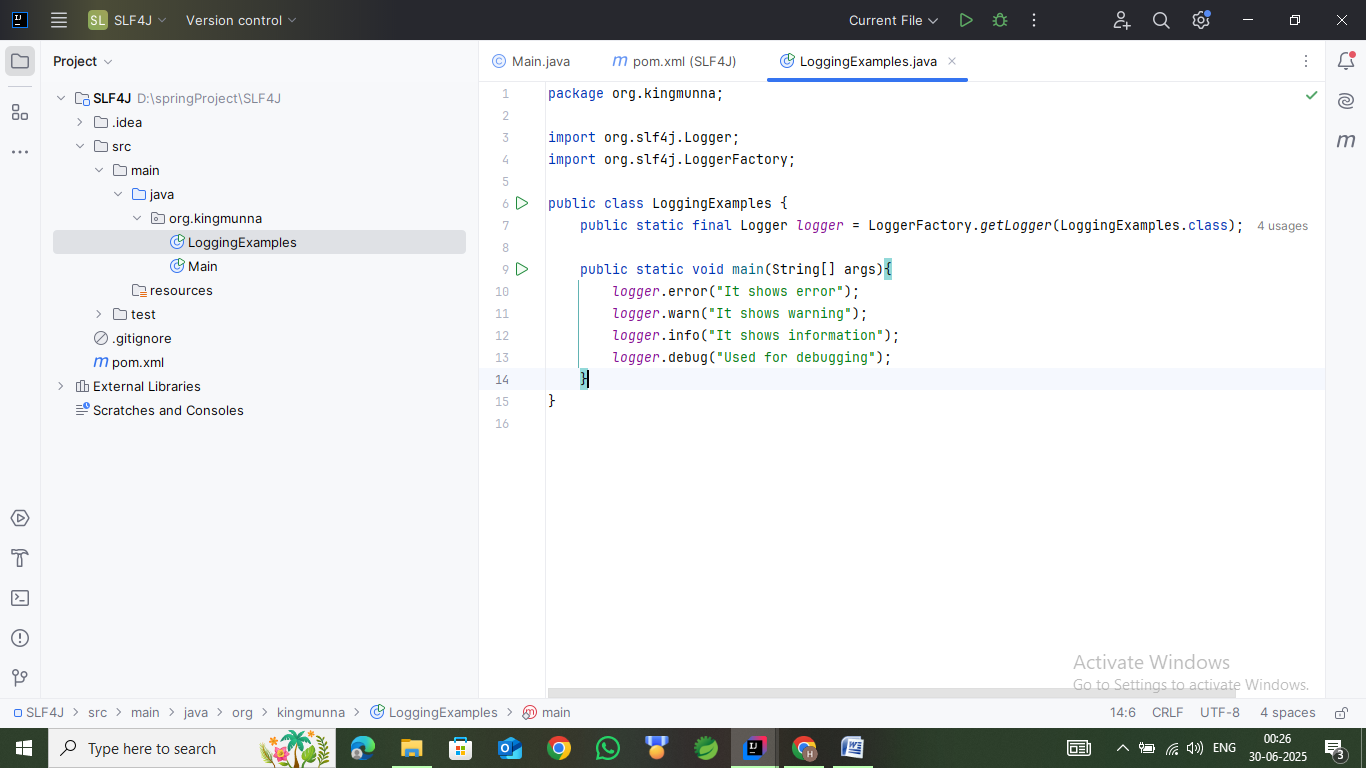
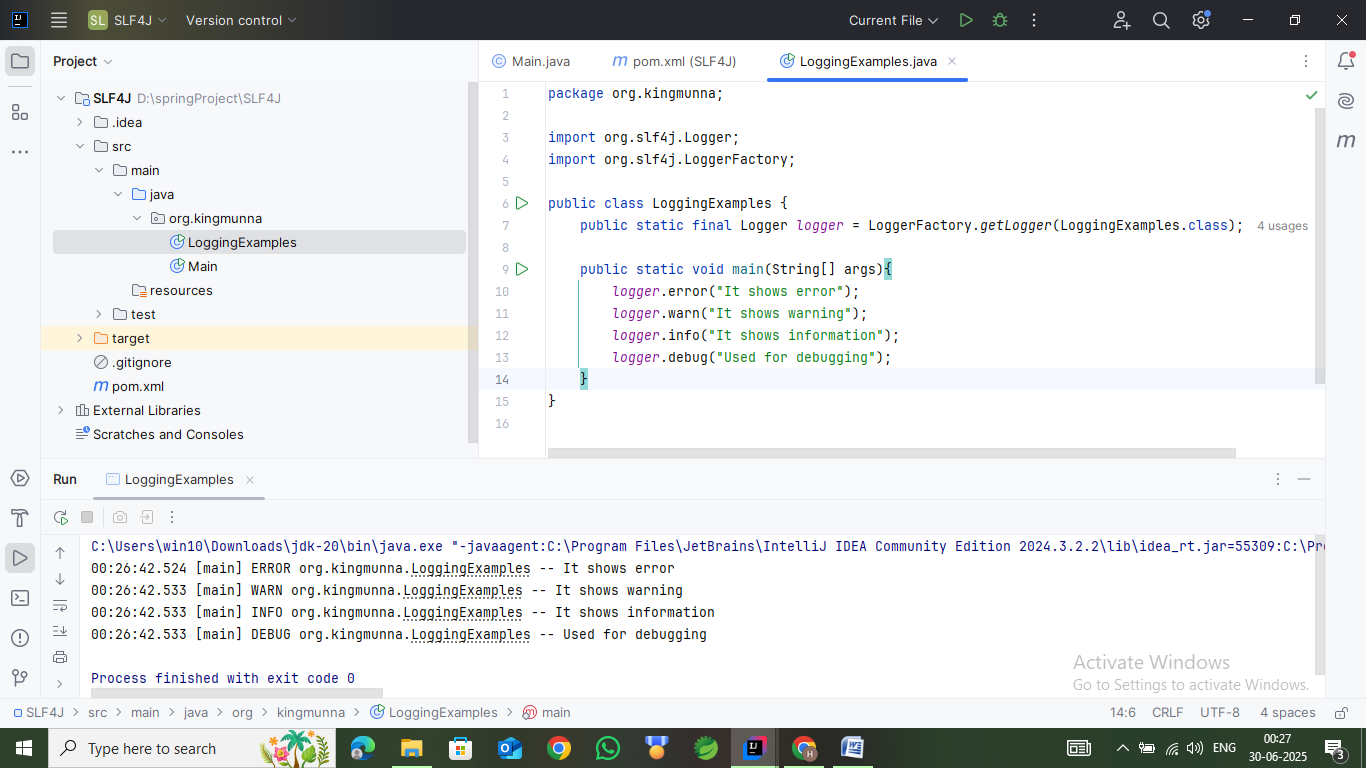
| ***Exercise 1: Logging Error Messages and Warning Levels*** |
| --- |

**Add Required Dependencies to pom.xml** To begin, we need to ensure that our project includes the necessary dependencies for SLF4J and Logback. SLF4J (Simple Logging Facade for Java) provides a generic API for logging, while Logback serves as the actual implementation. Since we are using Maven as our build tool, we should include the following dependencies inside the <dependencies> block of the pom.xml file:  
  
  


**Create the Logging Java Class** After configuring the dependencies, we can create a new Java class that uses the SLF4J Logger interface to demonstrate different log levels. In this example, we focus on logging messages at ERROR and WARN levels, while also including INFO and DEBUG to illustrate the full range of logging capabilities.  
  


In this class, we use LoggerFactory.getLogger() to create a logger instance associated with LoggingExample.class. We then use this logger to log messages at various levels. The actual visibility of these messages depends on the configuration of the underlying logging framework.

**Build and Run the Application** After setting up the logging configuration and writing the logging code, we can compile and run the application. With the log level set to warn, we should observe only the warning and error messages in the console output. If we wish to see informational and debug messages as well, we would need to adjust the logging level in the logback.xml file accordingly.



**Result** This exercise helps us understand how to integrate SLF4J with Logback, which is a common and powerful logging setup in modern Java applications. It demonstrates how we can log messages at various severity levels and manage log output through configuration. By following these steps, we can build a solid foundation for implementing effective logging practices, which are essential for debugging, performance monitoring, and application maintenance.