1. **JUnit\_Basic Testing Exercises**

**Exercise 1: Setting Up Junit**

**pom.xml**

<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 http://maven.apache.org/maven-v4\_0\_0.xsd">

  <modelVersion>4.0.0</modelVersion>

  <groupId>com.example</groupId>

  <artifactId>Exercise1\_SettingUpJUnit</artifactId>

  <packaging>jar</packaging>

  <version>1.0-SNAPSHOT</version>

  <name>Exercise1\_SettingUpJUnit</name>

  <url>http://maven.apache.org</url>

<dependencies>

    <dependency>

      <groupId>junit</groupId>

      <artifactId>junit</artifactId>

      <version>4.13.2</version>

      <scope>test</scope>

      </dependency>

  </dependencies>

  <build>

  <plugins>

    <plugin>

      <groupId>org.apache.maven.plugins</groupId>

      <artifactId>maven-surefire-plugin</artifactId>

      <version>3.0.0-M5</version>

    </plugin>

  </plugins>

</build>

</project>

**Multiplication.java**

package com.example;

public class Multiplication {

    public int multiply(int a, int b) {

        return a \* b;

    }

}

**MultiplicationTest.java**

package com.example;

import org.junit.Test;

import static org.junit.Assert.assertEquals;

public class MultiplicationTest {

    @Test

    public void testMultiply() {

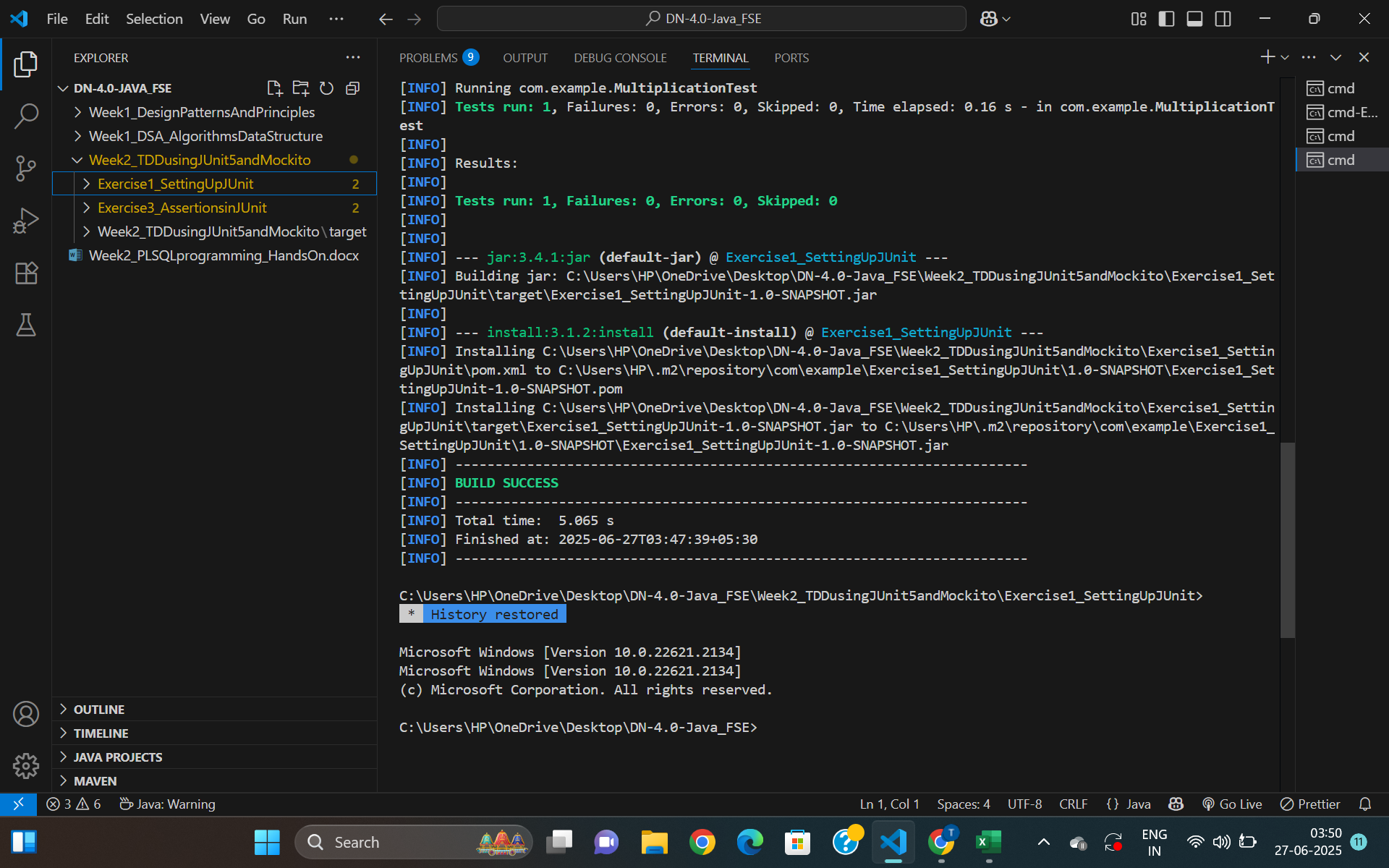
        Multiplication calc = new Multiplication();

        assertEquals(36, calc.multiply(9, 4));

    }

}

Output:



**Exercise 3: Assertions in JUnit**

**AssertionsTest.java**

package com.example;

import org.junit.Test;

import static org.junit.Assert.\*;

public class AssertionsTest {

    @Test

    public void testAssertions() {

        assertEquals(5, 2 + 3);

        assertTrue(5 > 3);

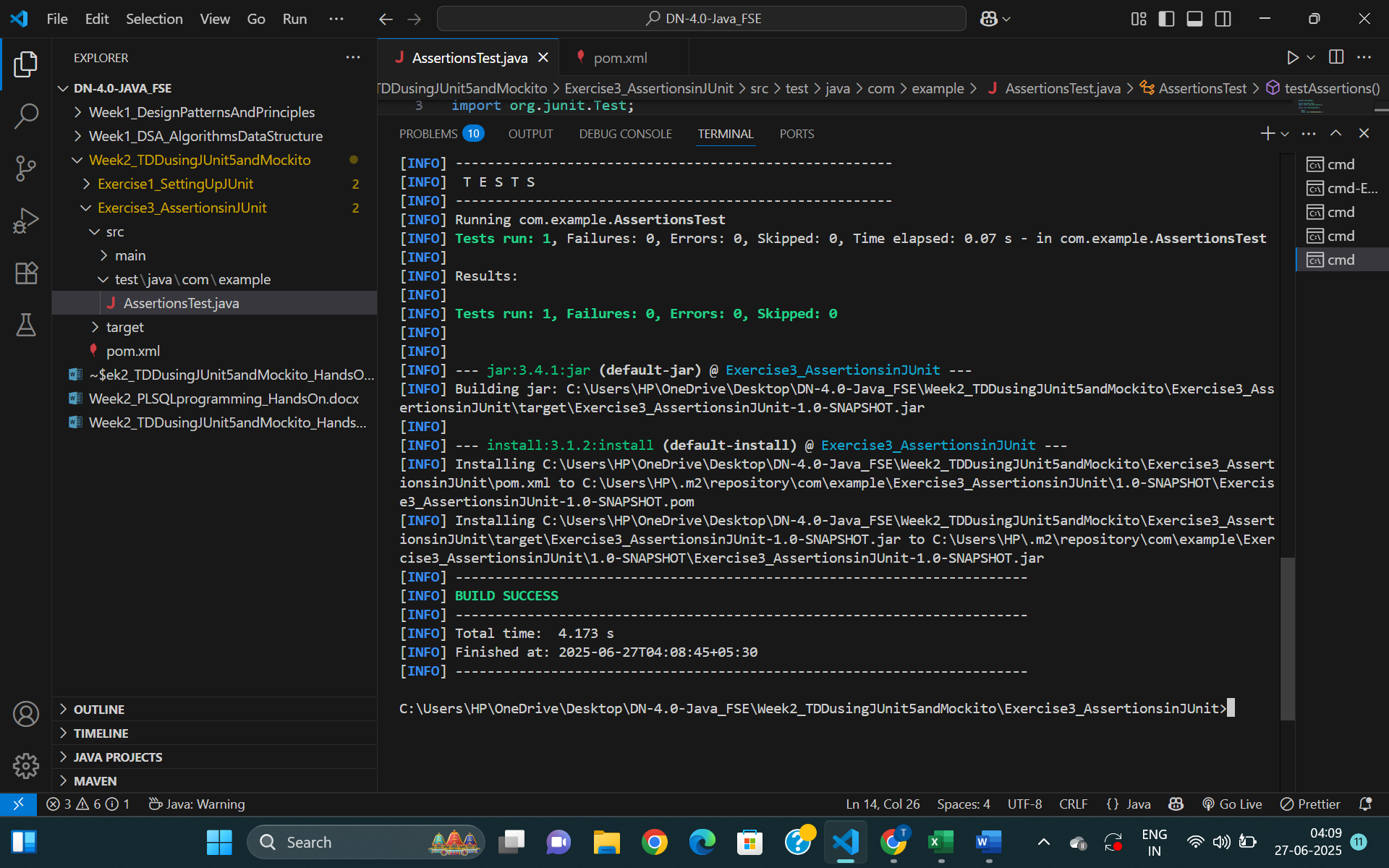
        assertFalse(5 < 3);

        assertNull(null);

        assertNotNull(new Object());

    }

}

**Output:**

**Exercise 4: Arrange-Act-Assert (AAA) Pattern, Test Fixtures, Setup and Teardown Methods in JUnit**

**Calculator.java**

package com.example;

public class Calculator {

    public int sum(int a, int b) {

        return a + b;

    }

}

**CalculatorTest.java**

package com.example;

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

import static org.junit.Assert.assertEquals;

public class CalculatorTest {

    private Calculator calculator;

    @Before

    public void setUp() {

        // Arrange

        calculator = new Calculator();

        System.out.println("Setup complete.");

    }

    @After

    public void tearDown() {

        // Cleanup

        calculator = null;

        System.out.println("Teardown complete.");

    }

    @Test

    public void testAddPositiveNumbers() {

        // Act

        int result = calculator.sum(10, 5);

// Assert

        assertEquals(15, result);

    }

    @Test

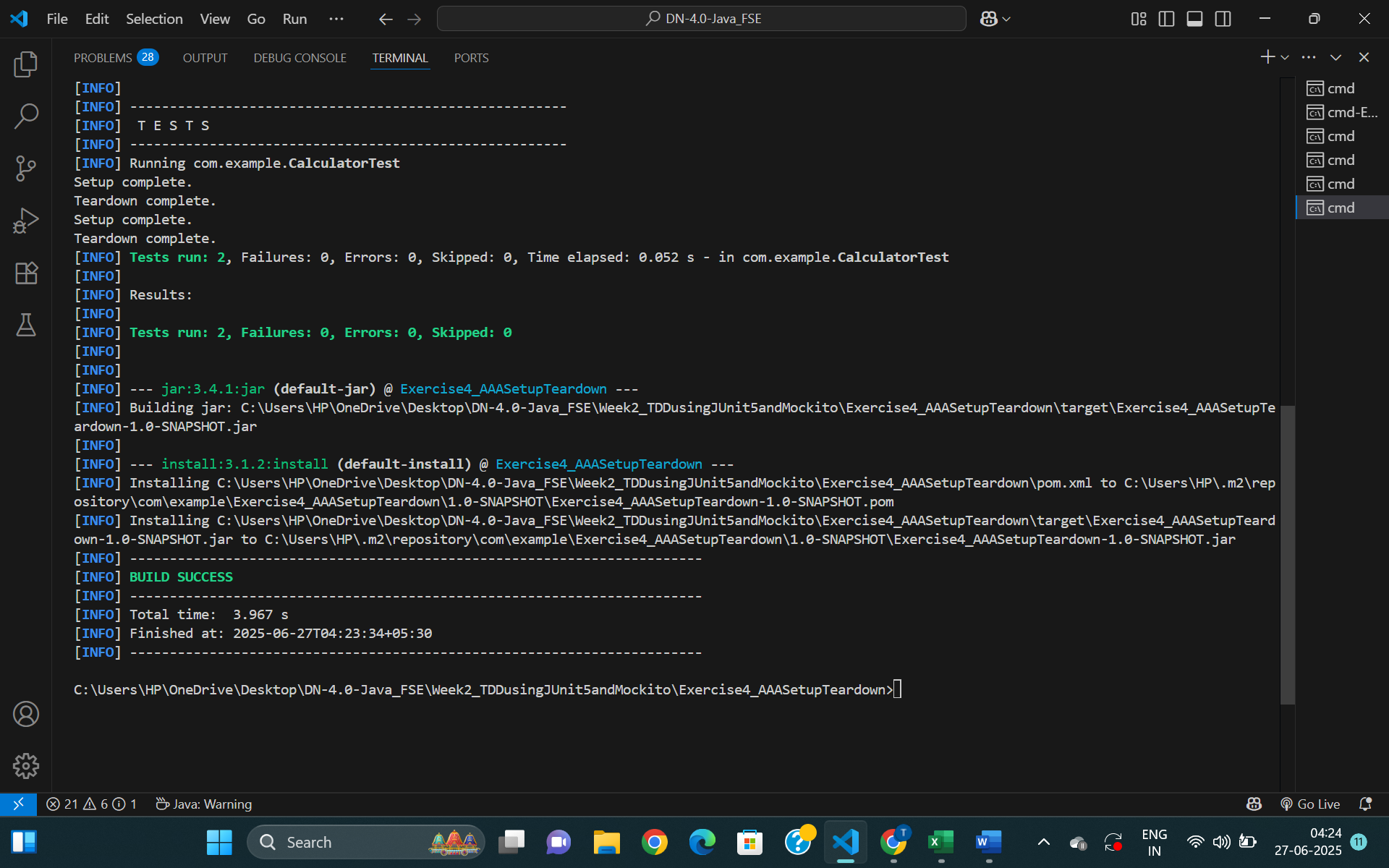
    public void testAddNegativeNumbers() {

        int result = calculator.sum(-2, -3);

        assertEquals(-5, result);

    }

}

Output:

1. **Mockito exercises**

**Exercise 1: Mocking and Stubbing**

**pom.xml**

<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 http://maven.apache.org/maven-v4\_0\_0.xsd">

  <modelVersion>4.0.0</modelVersion>

  <groupId>com.example</groupId>

  <artifactId>Exercise1\_MockingAndStubbing</artifactId>

  <version>1.0-SNAPSHOT</version>

  <properties>

    <maven.compiler.source>17</maven.compiler.source>

    <maven.compiler.target>17</maven.compiler.target>

  </properties>

  <dependencies>

    <!-- JUnit 5 -->

    <dependency>

      <groupId>org.junit.jupiter</groupId>

      <artifactId>junit-jupiter</artifactId>

      <version>5.10.2</version>

      <scope>test</scope>

    </dependency>

    <!-- Mockito -->

    <dependency>

      <groupId>org.mockito</groupId>

      <artifactId>mockito-core</artifactId>

      <version>5.11.0</version>

      <scope>test</scope>

    </dependency>

  </dependencies>

  <build>

    <plugins>

      <plugin>

        <groupId>org.apache.maven.plugins</groupId>

        <artifactId>maven-surefire-plugin</artifactId>

        <version>3.2.5</version>

      </plugin>

    </plugins>

  </build>

</project>

**ExternalApi.java**

package com.example;

public interface ExternalApi {

    String getData();

}

**MyService.java**

package com.example;

public class MyService {

    private final ExternalApi externalApi;

    public MyService(ExternalApi externalApi) {

        this.externalApi = externalApi;

    }

    public String fetchData() {

        return externalApi.getData();

    }

}

**MyServiceTest.java**

package com.example;

import org.junit.jupiter.api.Test;

import static org.mockito.Mockito.\*;

import static org.junit.jupiter.api.Assertions.\*;

public class MyServiceTest {

    @Test

    public void testExternalApi() {

        ExternalApi mockApi = mock(ExternalApi.class);

        when(mockApi.getData()).thenReturn("Mock Data");

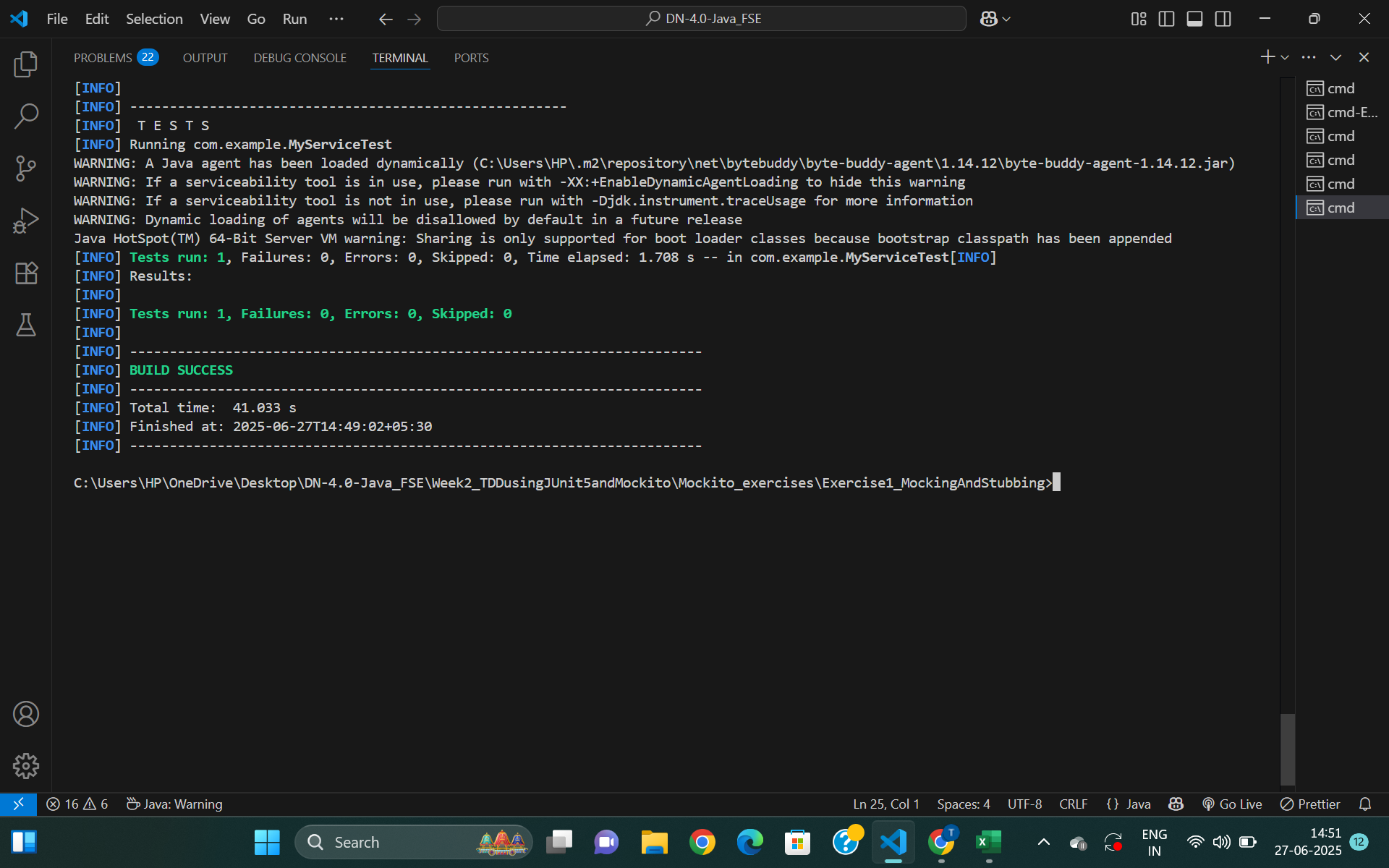
        MyService service = new MyService(mockApi);

        String result = service.fetchData();

        assertEquals("Mock Data", result);

    }

}

**Output**

**Exercise 2: Verifying Interactions**

**ExternalApi.java**

package com.example;

public interface ExternalApi {

    String getData();

}

**MyService.java**

package com.example;

public class MyService {

    private final ExternalApi externalApi;

    public MyService(ExternalApi externalApi) {

        this.externalApi = externalApi;

    }

    public String fetchData() {

        return externalApi.getData(); // Method to verify interaction

    }

}

**MyServiceTest.java**

package com.example;

import org.junit.jupiter.api.Test;

import static org.mockito.Mockito.\*;

public class MyServiceTest {

    @Test

    public void testVerifyInteraction() {

        ExternalApi mockApi = mock(ExternalApi.class);

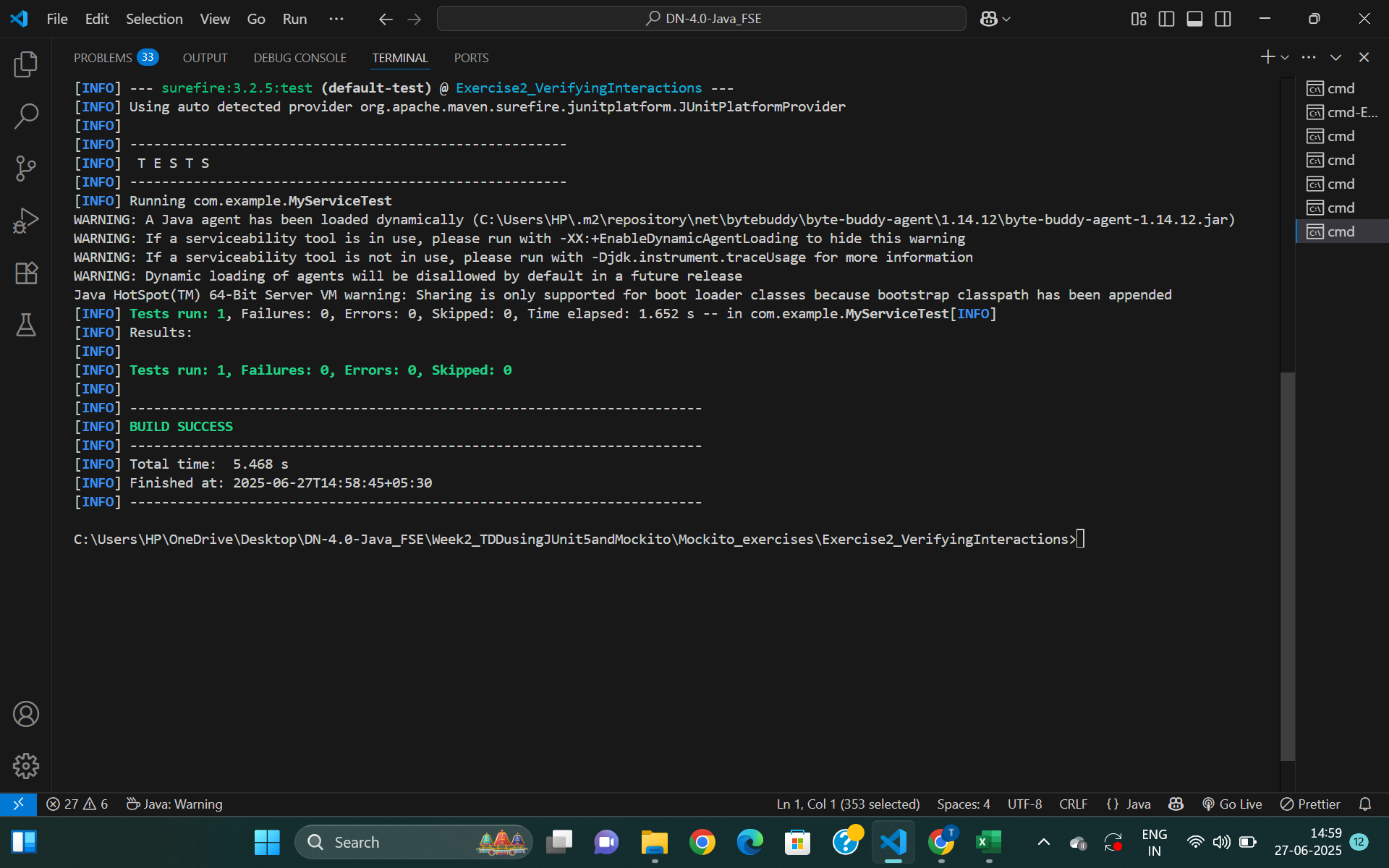
        MyService service = new MyService(mockApi);

        service.fetchData();

        verify(mockApi).getData();

    }

}

**Output**