**JUnit Testing**

**JUnit Testing Exercises:**

**Exercise 1:**

**Calculator.java**

package com.example.junitdemo;

public class Calculator {

    public int add(int a, int b) {

        return a + b;

    }

}

**CalculatorTest.java**

package com.example.junitdemo;

import static org.junit.jupiter.api.Assertions.assertEquals;

import org.junit.jupiter.api.Test;

public class CalculatorTest {

    @Test

    public void testAdd() {

        Calculator calc = new Calculator();

        int result = calc.add(2, 3);

        assertEquals(5, result);

    }

}

**Pom.xml**

<dependency>

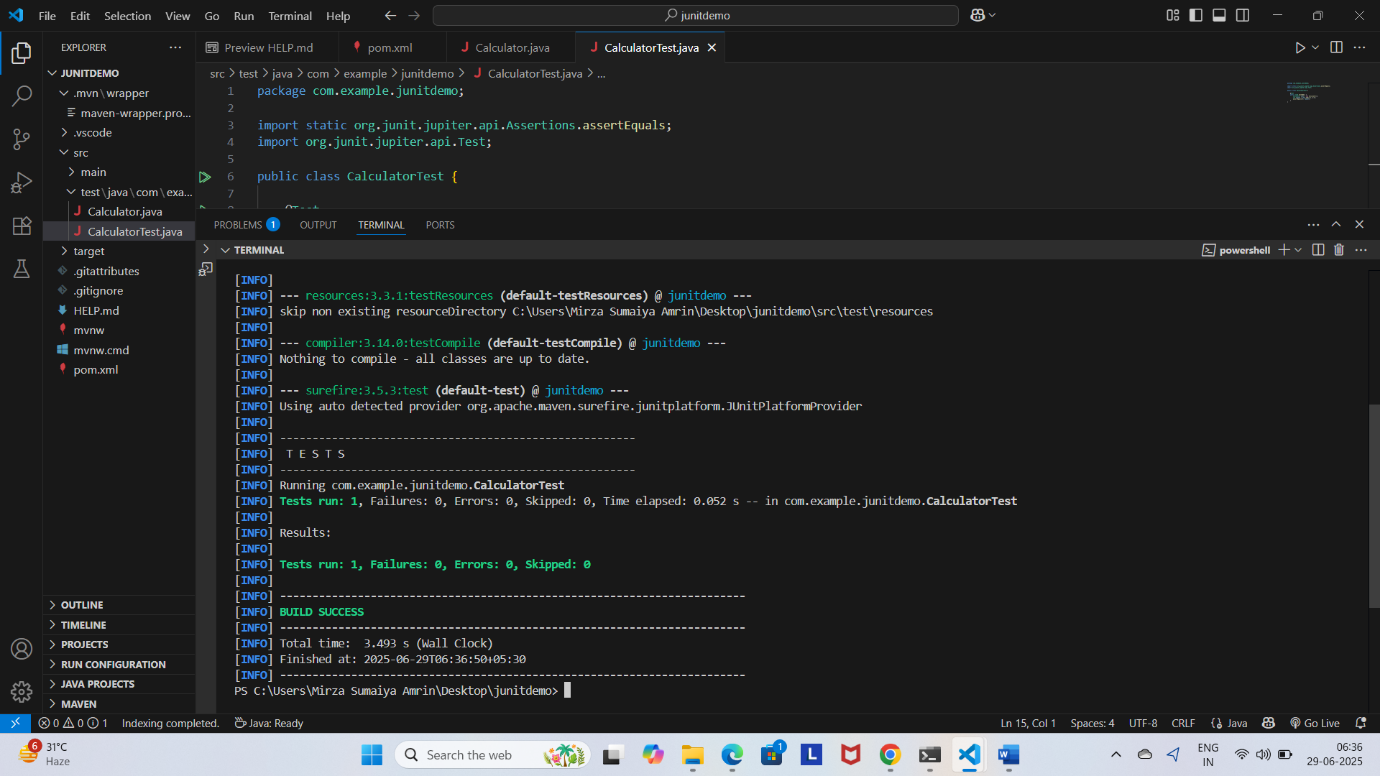
            <groupId>junit</groupId>

            <artifactId>junit</artifactId>

            <scope>test</scope>

        </dependency>

**Output**

****

**Exercise 2:**

**Calculator.java**

package com.example.junitdemo;

public class Calculator {

public int add(int a, int b) {

return a + b;

}

public int subtract(int a, int b) {

return a - b;

}

}

**CalculatorTest.java**

// File: src/test/java/com/example/junitdemo/CalculatorTest.java

package com.example.junitdemo;

import static org.junit.jupiter.api.Assertions.assertEquals;

import org.junit.jupiter.api.Test;

public class CalculatorTest {

    @Test

    void testAdd() {

        Calculator calc = new Calculator();

        assertEquals(5, calc.add(2, 3));

    }

    @Test

    void testSubtract() {

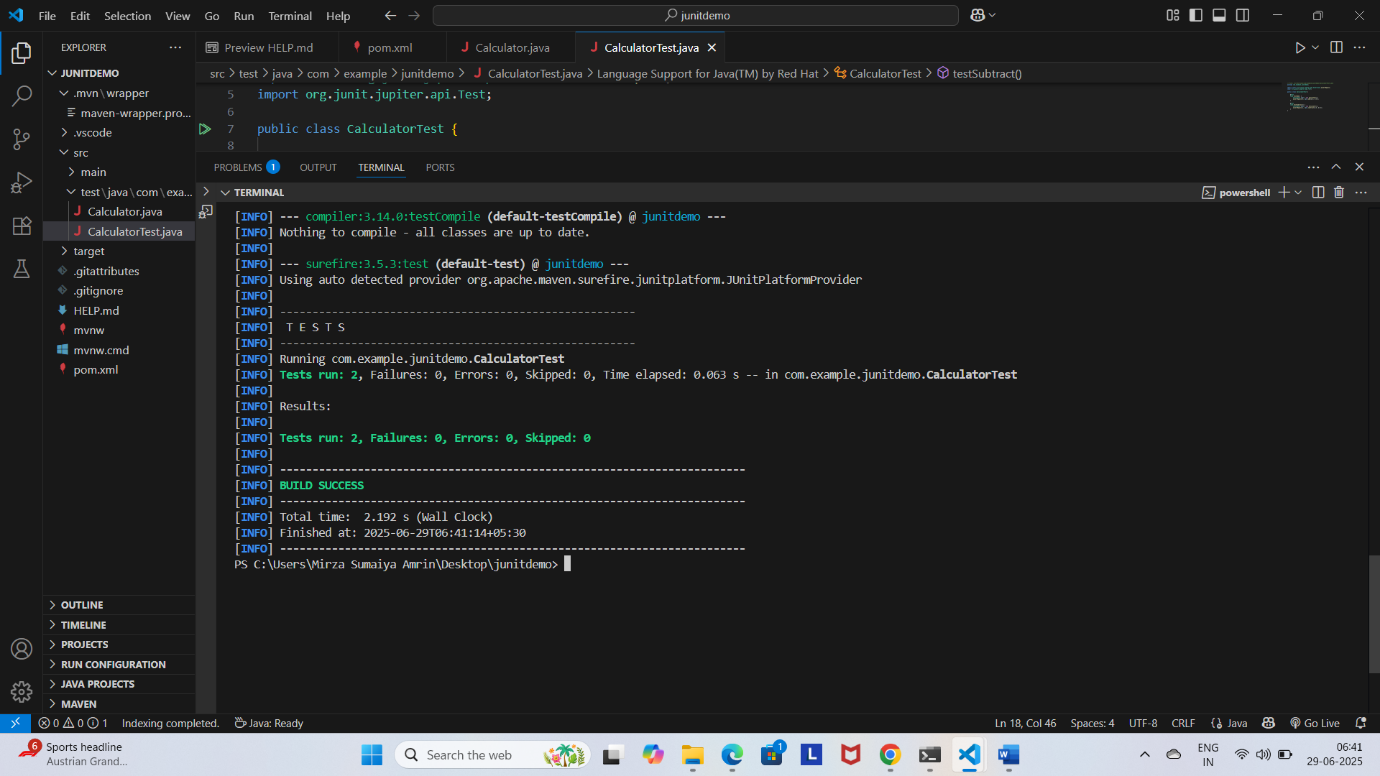
        Calculator calc = new Calculator();

        assertEquals(1, calc.subtract(4, 3));

    }

}

**Output**

****

**Exercise 3:**

**AssertionsTest.java**

// File: src/test/java/com/example/junitdemo/AssertionsTest.java

package com.example.junitdemo;

import static org.junit.jupiter.api.Assertions.assertEquals;

import static org.junit.jupiter.api.Assertions.assertFalse;

import static org.junit.jupiter.api.Assertions.assertNotNull;

import static org.junit.jupiter.api.Assertions.assertNull;

import static org.junit.jupiter.api.Assertions.assertTrue;

import org.junit.jupiter.api.Test;

public class AssertionsTest {

    @Test

    void testAssertions() {

        // assertEquals

        assertEquals(5, 2 + 3);

        // assertTrue

        assertTrue(5 > 3);

        // assertFalse

        assertFalse(5 < 3);

        // assertNull

        String str = null;

        assertNull(str);

        // assertNotNull

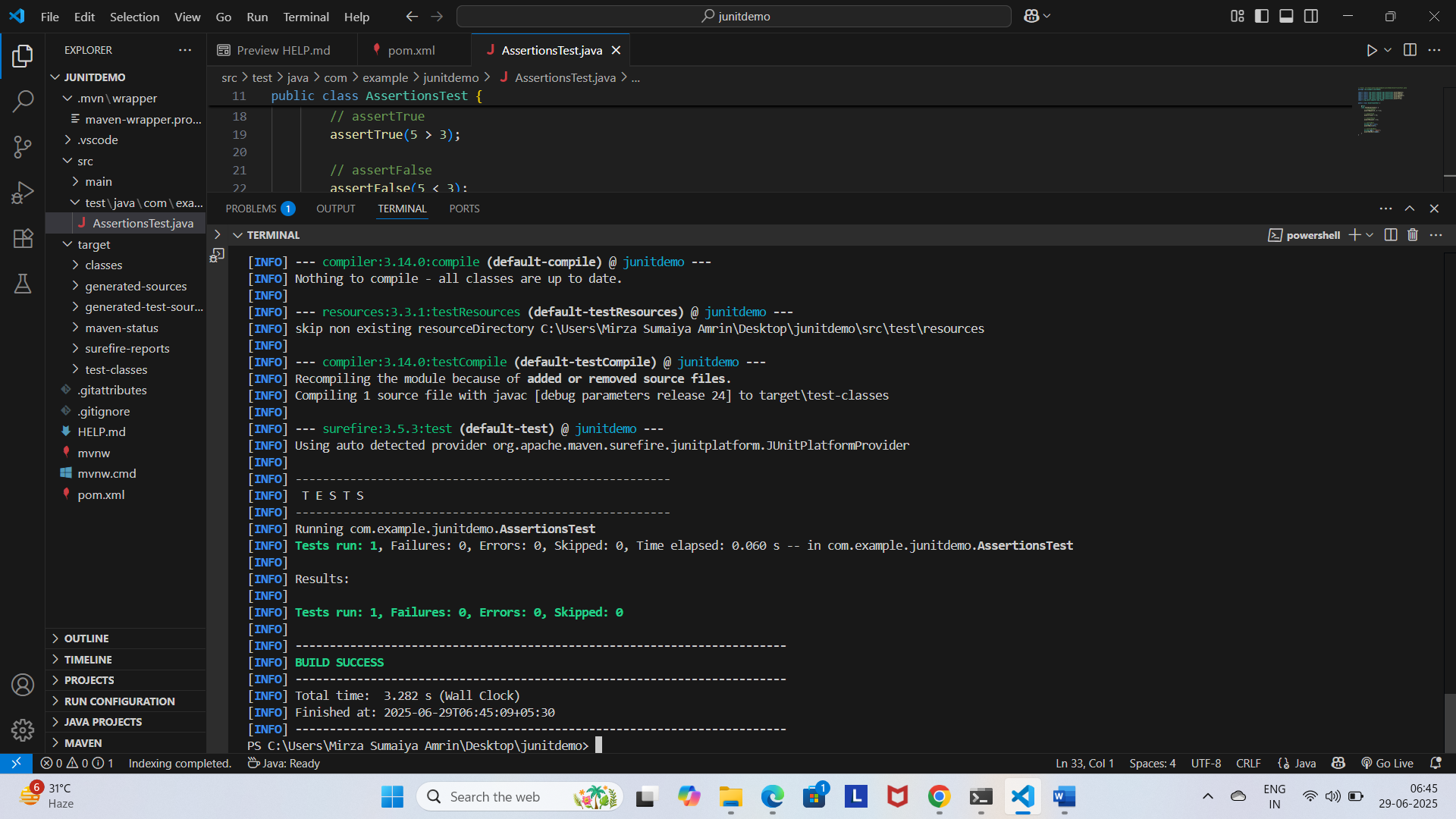
        String name = "JUnit";

        assertNotNull(name);

    }

}

**Output:**

****

**Exercise 4:**

**Calculator.java**

// File: src/main/java/com/example/junitdemo/Calculator.java

package com.example.junitdemo;

public class Calculator {

    public int add(int a, int b) {

        return a + b;

    }

    public int subtract(int a, int b) {

        return a - b;

    }

}

**CalculatorLifecycleTest.java**

// File: src/test/java/com/example/junitdemo/CalculatorLifecycleTest.java

package com.example.junitdemo;

import org.junit.jupiter.api.AfterEach;

import static org.junit.jupiter.api.Assertions.assertEquals;

import org.junit.jupiter.api.BeforeEach;

import org.junit.jupiter.api.Test;

public class CalculatorLifecycleTest {

    private Calculator calculator;

    @BeforeEach

    void setUp() {

        // Arrange: Create Calculator instance before each test

        calculator = new Calculator();

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

    }

    @AfterEach

    void tearDown() {

        // Cleanup: This runs after each test

        calculator = null;

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

    }

    @Test

    void testAddition() {

        // Act

        int result = calculator.add(10, 20);

        // Assert

        assertEquals(30, result);

    }

    @Test

    void testSubtraction() {

        // Act

        int result = calculator.subtract(10, 4);

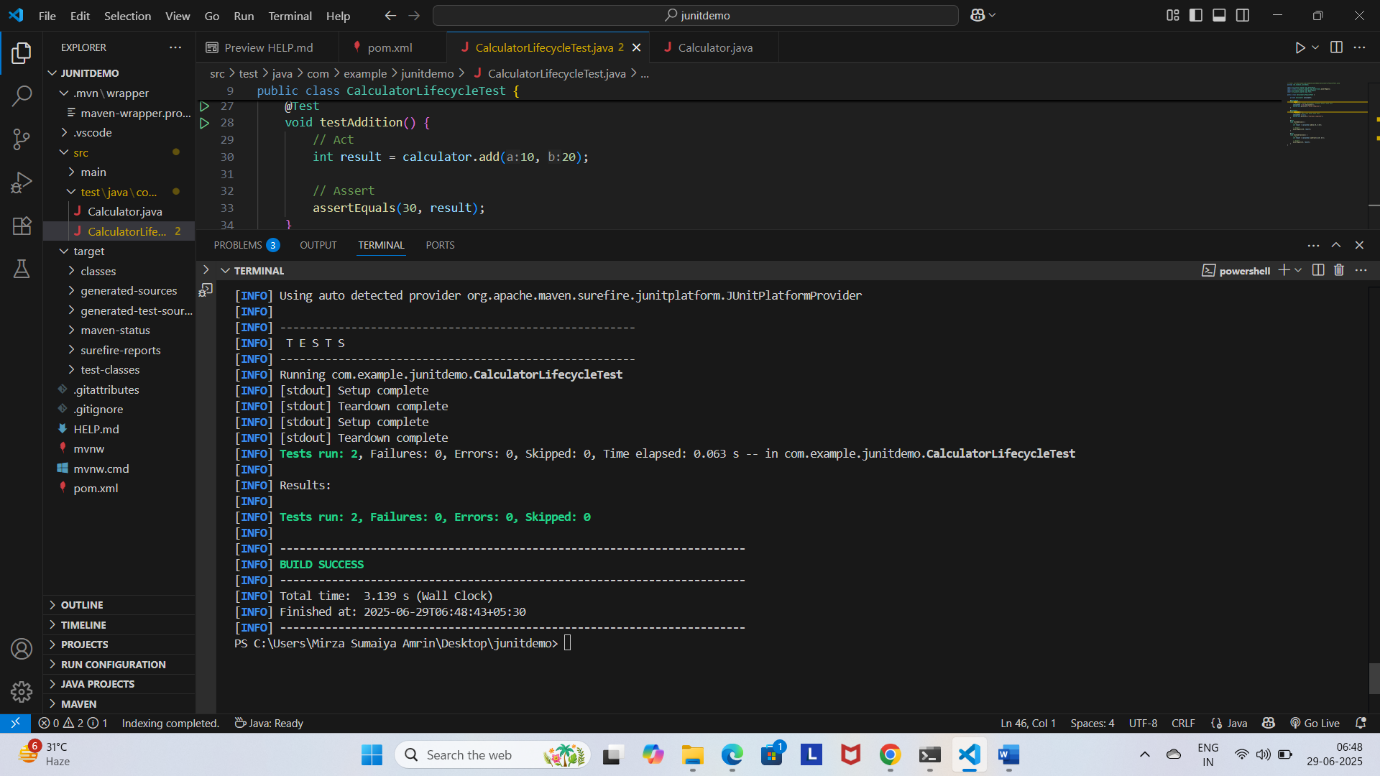
        // Assert

        assertEquals(6, result);

    }

}

**Output:**

****

**Exercise 1:**

**EvenChecker.java**

ckage com.example.junitdemo;

public class EvenChecker {

    public boolean isEven(int number) {

        return number % 2 == 0;

    }

}

**EvenCheckerTest.java**

package com.example.junitdemo;

import static org.junit.jupiter.api.Assertions.assertFalse;

import static org.junit.jupiter.api.Assertions.assertTrue;

import org.junit.jupiter.params.ParameterizedTest;

import org.junit.jupiter.params.provider.ValueSource;

public class EvenCheckerTest {

    private final EvenChecker checker = new EvenChecker();

    @ParameterizedTest

    @ValueSource(ints = {2, 4, 6, 8, 10})

    void testIsEven(int number) {

        assertTrue(checker.isEven(number));

    }

    @ParameterizedTest

    @ValueSource(ints = {1, 3, 5, 7, 9})

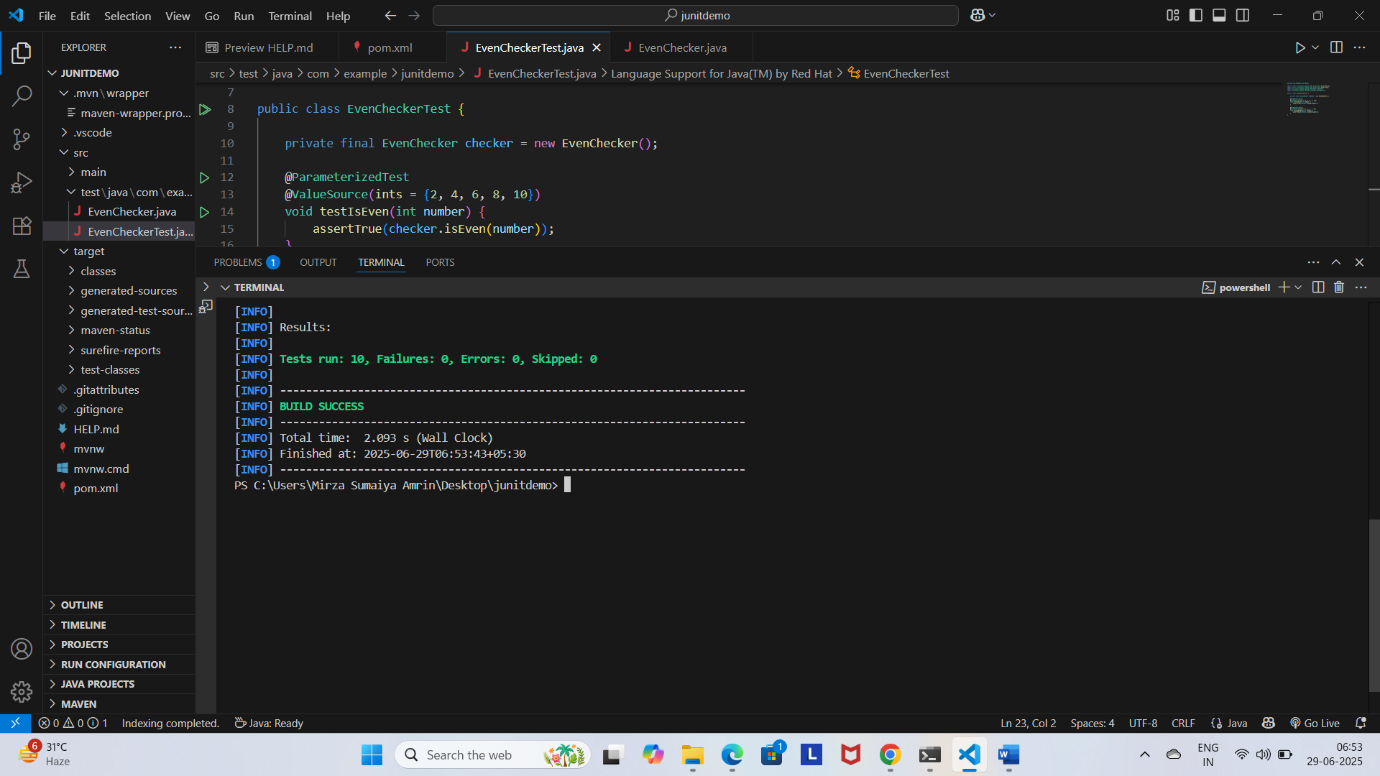
    void testIsNotEven(int number) {

        assertFalse(checker.isEven(number));

    }

}

**Output:**

****

**Exercise 2:**

**pom.xml**

<dependency>

    <groupId>org.junit.platform</groupId>

    <artifactId>junit-platform-suite</artifactId>

     <!-- Match your JUnit version -->

    <scope>test</scope>

</dependency>

    <dependency>

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

        <artifactId>junit-jupiter</artifactId>

        <scope>test</scope>

    </dependency>

**Calculator.java**

// File: src/main/java/com/example/junitdemo/Calculator.java

package com.example.junitdemo;

public class Calculator {

    public int add(int a, int b) {

        return a + b;

    }

    public int subtract(int a, int b) {

        return a - b;

    }

}

**CalculatorTest.java**

// File: src/test/java/com/example/junitdemo/CalculatorTest.java

package com.example.junitdemo;

import static org.junit.jupiter.api.Assertions.assertEquals;

import org.junit.jupiter.api.Test;

public class CalculatorTest {

    @Test

    void testAdd() {

        Calculator calc = new Calculator();

        assertEquals(5, calc.add(2, 3));

    }

    @Test

    void testSubtract() {

        Calculator calc = new Calculator();

        assertEquals(1, calc.subtract(4, 3));

    }

}

**AssertionsTest.java**

// File: src/test/java/com/example/junitdemo/AssertionsTest.java

package com.example.junitdemo;

import static org.junit.jupiter.api.Assertions.assertEquals;

import static org.junit.jupiter.api.Assertions.assertFalse;

import static org.junit.jupiter.api.Assertions.assertNotNull;

import static org.junit.jupiter.api.Assertions.assertNull;

import static org.junit.jupiter.api.Assertions.assertTrue;

import org.junit.jupiter.api.Test;

public class AssertionsTest {

    @Test

    void testAssertions() {

        // assertEquals

        assertEquals(5, 2 + 3);

        // assertTrue

        assertTrue(5 > 3);

        // assertFalse

        assertFalse(5 < 3);

        // assertNull

        String str = null;

        assertNull(str);

        // assertNotNull

        String name = "JUnit";

        assertNotNull(name);

    }

}

**EvenChecker.java**

package com.example.junitdemo;

public class EvenChecker {

    public boolean isEven(int number) {

        return number % 2 == 0;

    }

}

**EvenCheckerTest.java**

package com.example.junitdemo;

import static org.junit.jupiter.api.Assertions.assertFalse;

import static org.junit.jupiter.api.Assertions.assertTrue;

import org.junit.jupiter.params.ParameterizedTest;

import org.junit.jupiter.params.provider.ValueSource;

public class EvenCheckerTest {

    private final EvenChecker checker = new EvenChecker();

    @ParameterizedTest

    @ValueSource(ints = {2, 4, 6, 8, 10})

    void testIsEven(int number) {

        assertTrue(checker.isEven(number));

    }

    @ParameterizedTest

    @ValueSource(ints = {1, 3, 5, 7, 9})

    void testIsNotEven(int number) {

        assertFalse(checker.isEven(number));

    }

}

**AllTests.java**

package com.example.junitdemo;

import org.junit.platform.suite.api.SelectClasses;

import org.junit.platform.suite.api.Suite;

@Suite

@SelectClasses({

        CalculatorTest.class,

        AssertionsTest.class,

        EvenCheckerTest.class

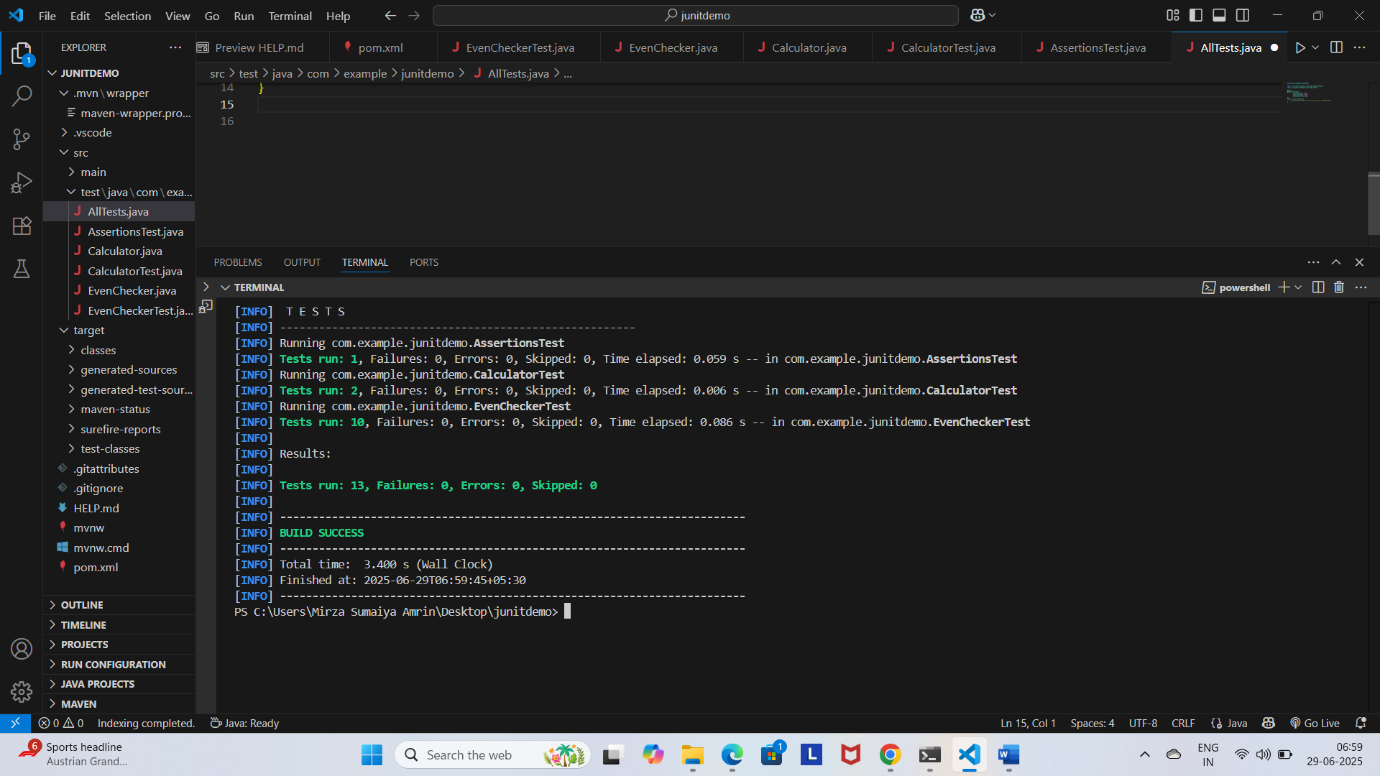
})

public class AllTests {

    // No methods needed – this class runs all included tests

}

**Output:**

****

**Exercise 3:**

**OrderedTests.java**

package com.example.junitdemo;

import static org.junit.jupiter.api.Assertions.assertTrue;

import org.junit.jupiter.api.MethodOrderer;

import org.junit.jupiter.api.Order;

import org.junit.jupiter.api.Test;

import org.junit.jupiter.api.TestMethodOrder;

@TestMethodOrder(MethodOrderer.OrderAnnotation.class)

public class OrderedTests {

    @Test

    @Order(1)

    void testFirst() {

        System.out.println("First test");

        assertTrue(true);

    }

    @Test

    @Order(2)

    void testSecond() {

        System.out.println("Second test");

        assertTrue(true);

    }

    @Test

    @Order(3)

    void testThird() {

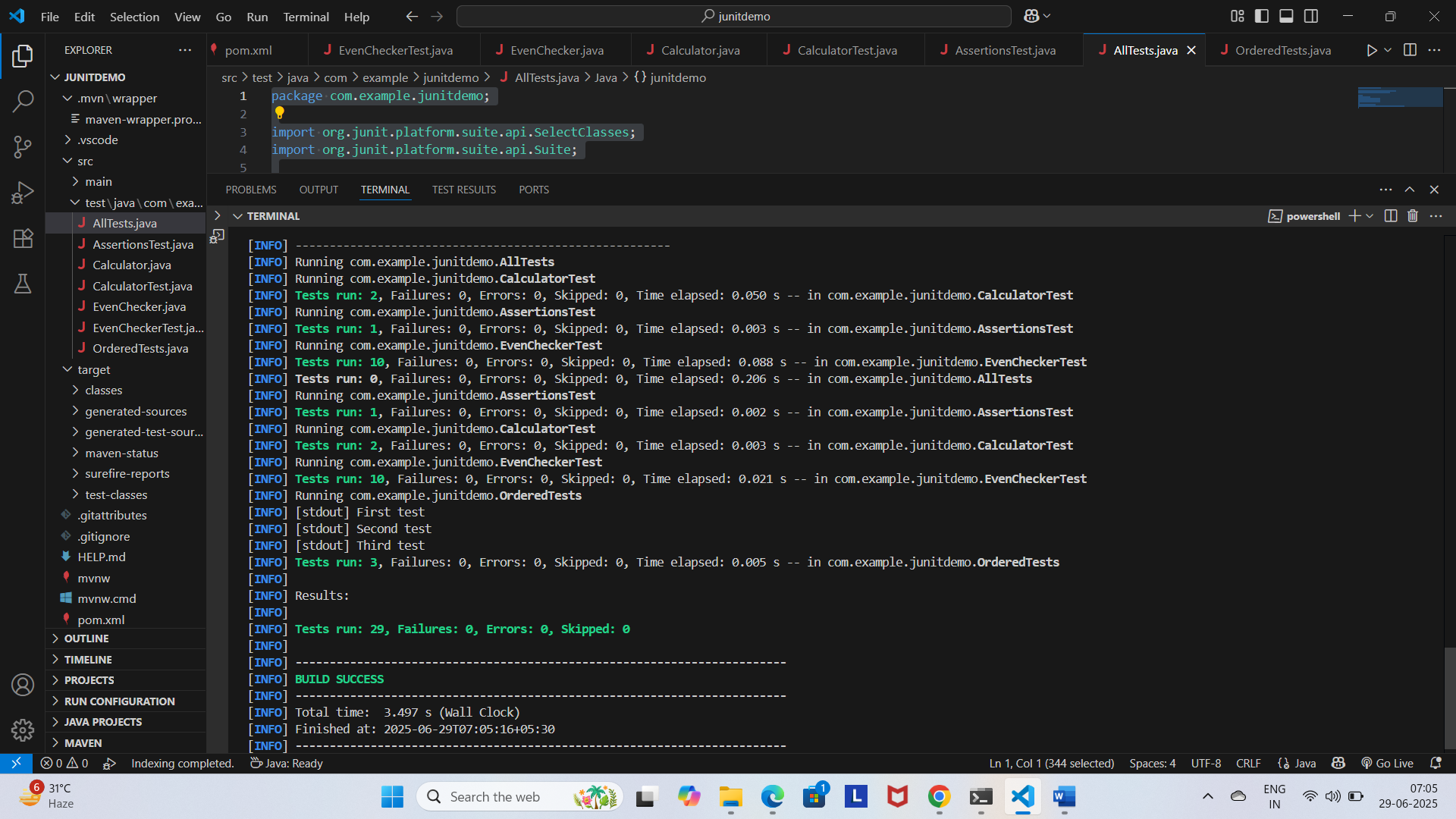
        System.out.println("Third test");

        assertTrue(true);

    }

}

**Output:**

****

**Exercise 4:**

**ExceptionThrower.java**

package com.example.junitdemo;

public class ExceptionThrower {

    public void throwException() {

        throw new IllegalArgumentException("Invalid argument!");

    }

}

**ExceptionThrowerTest.java**

package com.example.junitdemo;

import org.junit.jupiter.api.Test;

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

public class ExceptionThrowerTest {

    @Test

    void testExceptionIsThrown() {

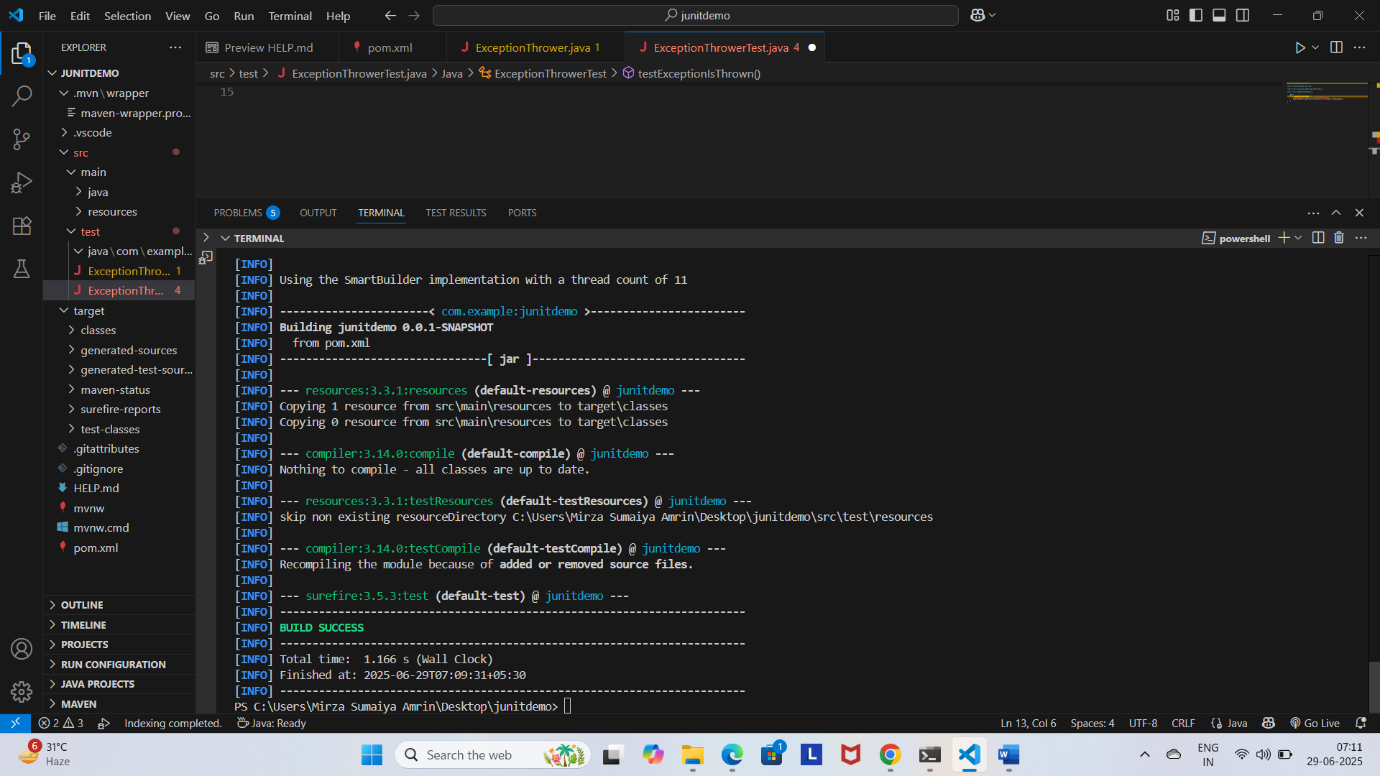
        ExceptionThrower thrower = new ExceptionThrower();

        assertThrows(IllegalArgumentException.class, thrower::throwException);

    }

}

**Output:**

****

**Exercise 5:**

**PerformanceTester.java**

package com.example.junitdemo;

public class PerformanceTester {

    public void performTask() throws InterruptedException {

        // Simulate some work

        Thread.sleep(500); // 500 ms

    }

}

**PerformanceTesterTest.java**

package com.example.junitdemo;

import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.assertTimeout;

import java.time.Duration;

public class PerformanceTesterTest {

    @Test

    void testPerformTaskWithinTime() {

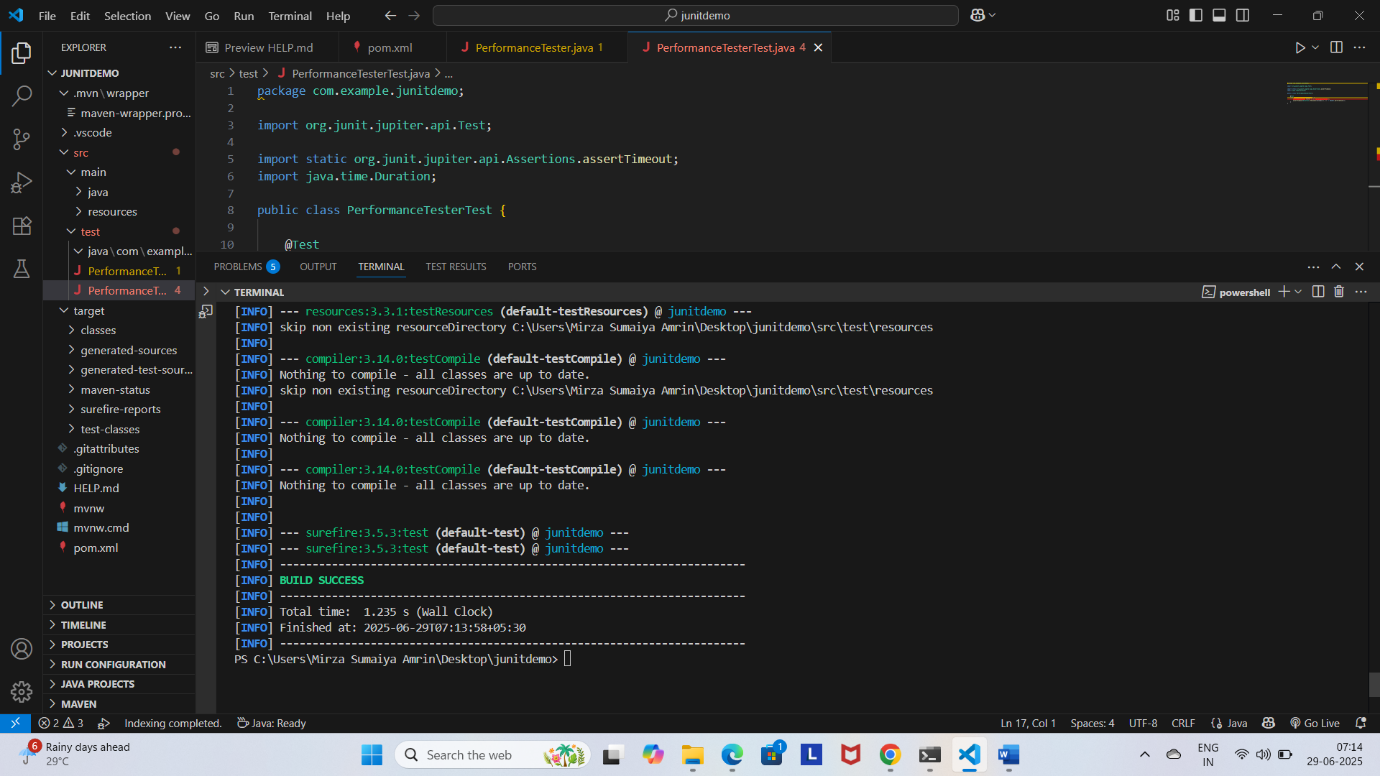
        PerformanceTester tester = new PerformanceTester();

        assertTimeout(Duration.ofSeconds(1), () -> tester.performTask());

    }

}

**Output:**

****

**3.**

**Exercise 1:**

**MyServiceTest.java**

import static org.mockito.Mockito.\*;

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

import org.junit.jupiter.api.Test;

class ExternalApi {

    public String getData() {

        return "Real Data";

    }

}

class MyService {

    private final ExternalApi api;

    public MyService(ExternalApi api) {

        this.api = api;

    }

    public String fetchData() {

        return api.getData();

    }

}

public class MyServiceTest {

    @Test

    public void testExternalApi() {

        ExternalApi mockApi = mock(ExternalApi.class);

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

        MyService service = new MyService(mockApi);

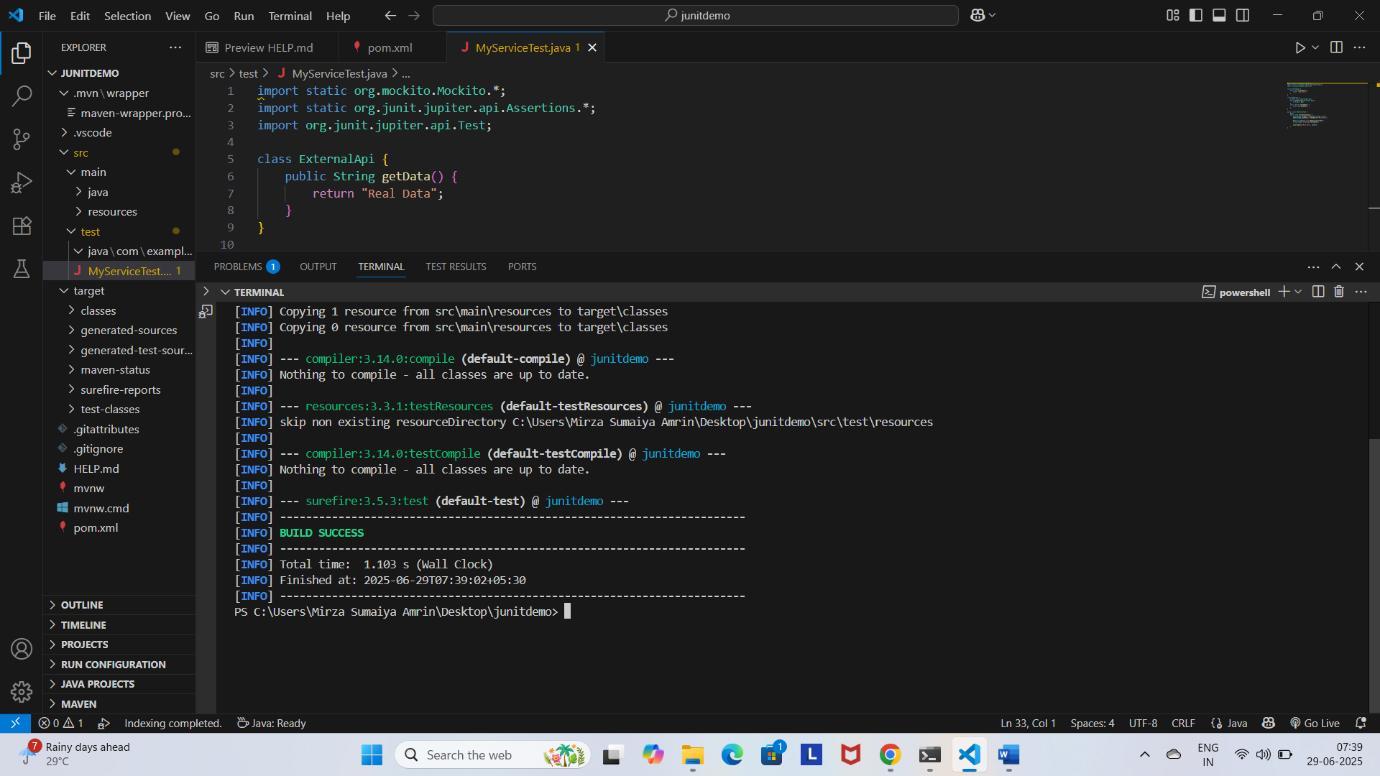
        String result = service.fetchData();

        assertEquals("Mock Data", result);

    }

}

**Output:**

****

**Exercise 2:**

**Exercise 3:**

**Exercise 4:**

**Exercise 5:**

**Exercise 6:**

**Exercise 7:**