**DIGITAL-NURTURE-4.0- JAVA FSE**

**WEEK-2 MOCKITO**

**Exercise 1: Mocking and Stubbing**

**PaymentGateway** – This is a third-party system (like Razorpay or PayPal) that actually processes the payment.

**PaymentService** – This is the code that **uses** the gateway to make a payment.

***PaymentGateway.java***

package net.javaguides;

public interface PaymentGateway {

    boolean processPayment(String accountNumber, double amount);

}

***PaymentService.java***

package net.javaguides;

public class PaymentService {

    private final PaymentGateway gateway;

public PaymentService(PaymentGateway gateway) {

        this.gateway = gateway;

    }

public String makePayment(String accountNumber, double amount) {

        boolean success = gateway.processPayment(accountNumber, amount);

        return success ? "Payment Successful" : "Payment Failed";

    }

}

***PaymentService.java***

package net.javaguides;

import org.junit.jupiter.api.Test;

import static org.mockito.Mockito.\*;

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

public class PaymentServiceTest {

    @Test

void testMakePayment\_Success() {

        PaymentGateway mockGateway = *mock*(PaymentGateway.class);

*when*(mockGateway.processPayment("12345", 1000.0)).thenReturn(true);

        PaymentService service = new PaymentService(mockGateway);

        String result = service.makePayment("12345", 1000.0);

*assertEquals*("Payment Successful", result);

    }

@Test

void testMakePayment\_Failure() {

        PaymentGateway mockGateway = *mock*(PaymentGateway.class);

*when*(mockGateway.processPayment("67890", 2000.0)).thenReturn(false);

        PaymentService service = new PaymentService(mockGateway);

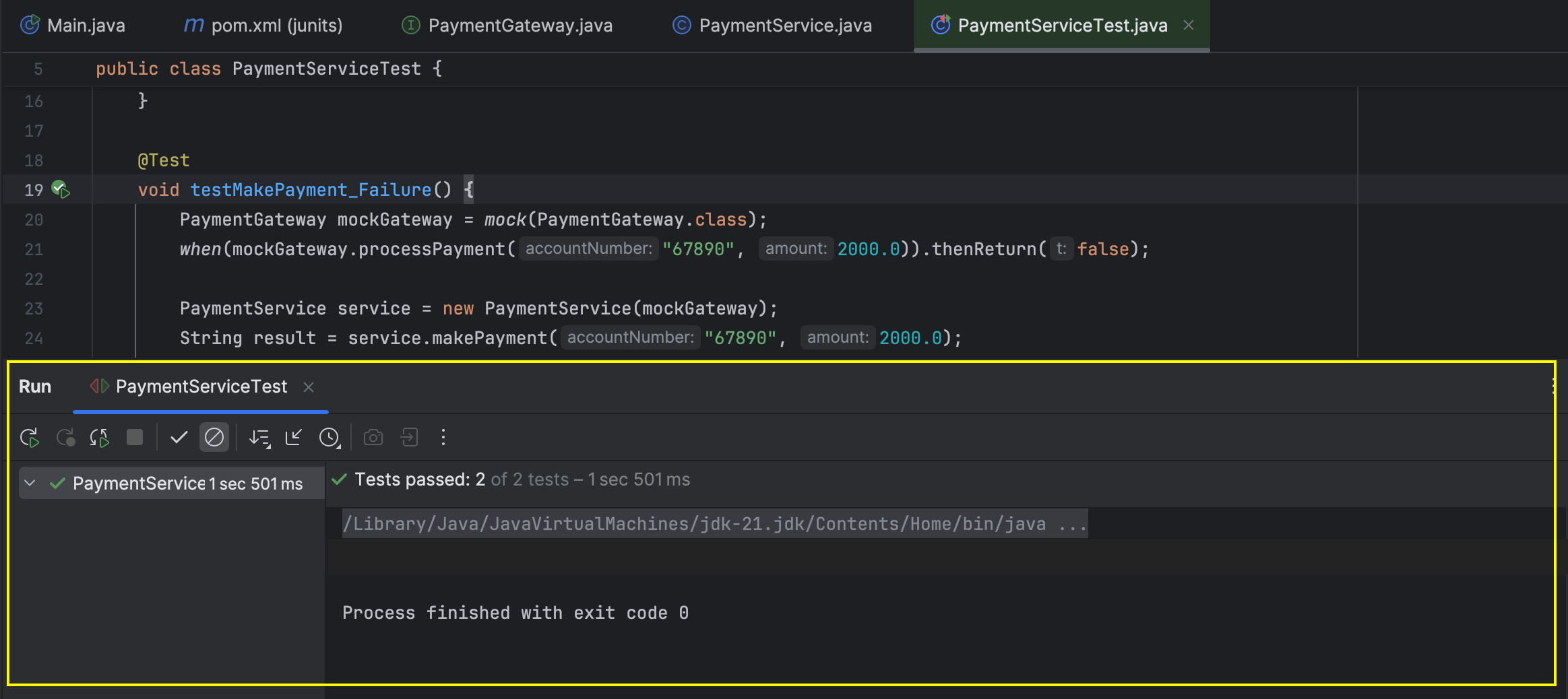
        String result = service.makePayment("67890", 2000.0);

*assertEquals*("Payment Failed", result);

    }

}

***OUPUT***



**Exercise 2: Verifying Interactions**

***EmailClient.java* [the interface that has to be mocked]**

public interface EmailClient {

void sendEmail(String email, String message);

}

***NotificationService.java* [the class we are going to test]**

public class NotificationService {

private final EmailClient emailClient;

public NotificationService(EmailClient emailClient) {

this.emailClient = emailClient;

}

public void notifyUser(String email, String message) {

emailClient.sendEmail(email, message);

}

}

***NotificationService.java***

import org.junit.jupiter.api.Test;

import static org.mockito.Mockito.\*;

public class NotificationServiceTest {

@Test

void testNotifyUserCalledWithCorrectArguments() {

EmailClient mockEmailClient = mock(EmailClient.class);

NotificationService service = new NotificationService(mockEmailClient);

service.notifyUser("user@example.com", "Welcome to our app!");

verify(mockEmailClient).sendEmail("user@example.com", "Welcome to our app!");

}

}

***Output***

