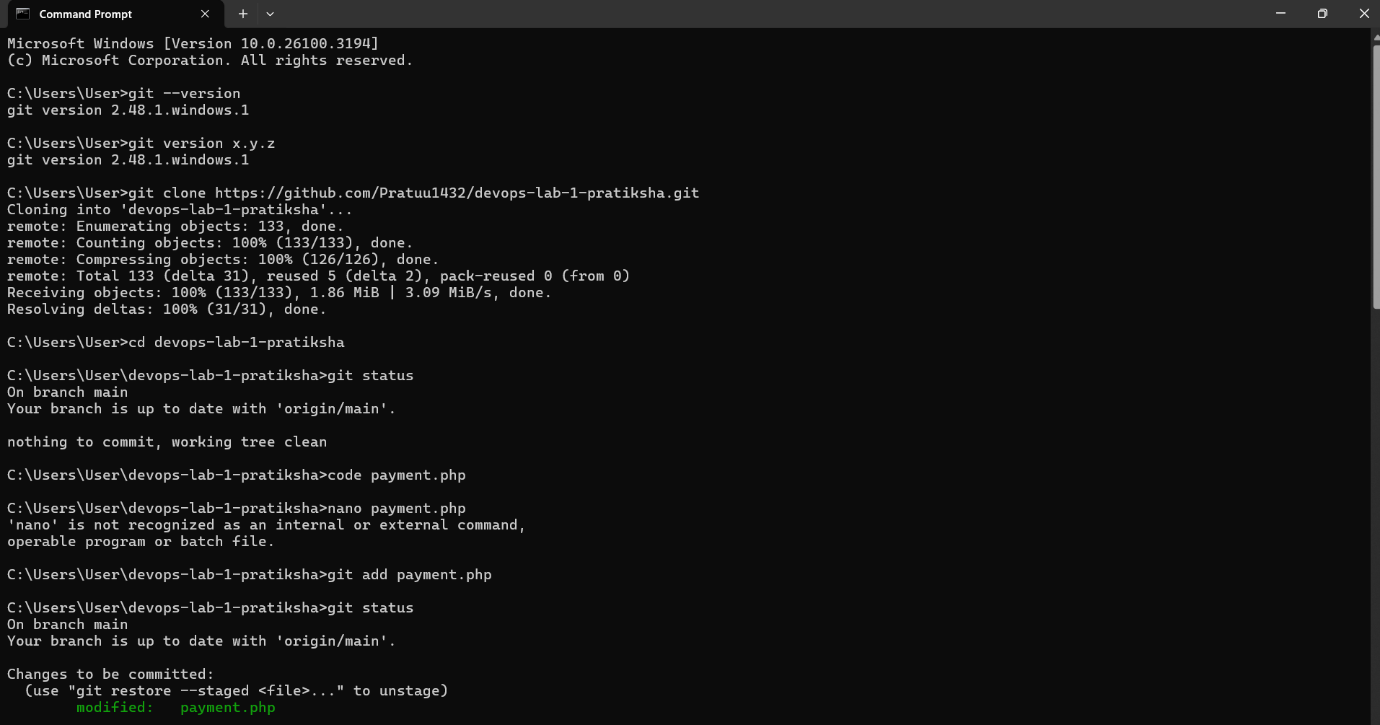
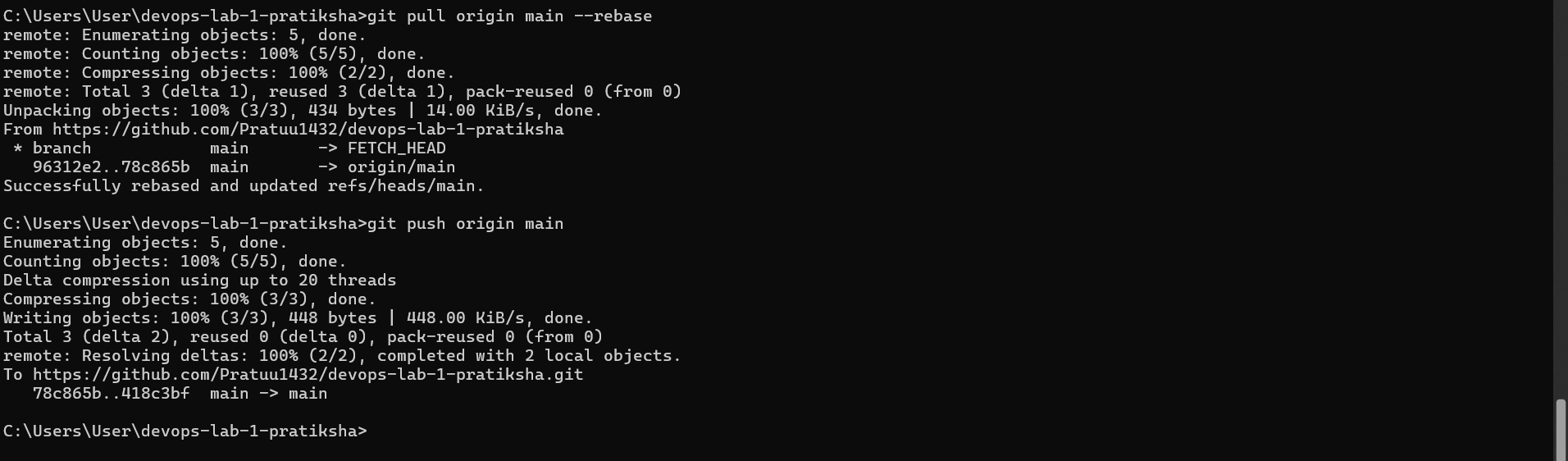
Name :- Pooja Mule

Enrollment no :- MITU23BTITD007

# Phase 1





# Phase 2

# Test Case Document for payment.php

## 1. Introduction

This document contains test cases for the `payment.php` file, focusing on functional, UI, and security testing. The execution results indicate the pass/fail status.

## 2. Test Cases

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Scenario | Steps to Execute | Expected Result | Actual Result | Status |
| TC-01 | Page Load | Open `payment.php` in a browser | Page should load without errors |  |  |
| TC-02 | Form Display | Check if the payment form is visible | Payment form should be displayed correctly |  |  |
| TC-03 | Required Fields | Leave fields empty and submit | Error messages should appear |  |  |
| TC-04 | Valid Card Number | Enter a valid credit/debit card number and submit | Payment should process successfully |  |  |
| TC-05 | Invalid Card Number | Enter an invalid card number | Error message should be displayed |  |  |
| TC-06 | Expired Card | Enter an expired card number | Error message should appear |  |  |
| TC-07 | Security Testing | Try SQL Injection (`' OR '1'='1` in fields) | Input should be sanitized |  |  |
| TC-08 | UI Testing | Check alignment, font, and colors of the page | UI should be visually correct |  |  |
| TC-09 | Network Failure | Disconnect from the internet and submit payment | Appropriate error should appear |  |  |
| TC-10 | Automation Using Selenium | Run an automated test script | Payment workflow should be tested |  |  |

## 3. Automated Testing Approach

Tool: Selenium WebDriver

Script: Automates form filling and payment submission

Verification: Checks success/failure messages

## 4. Selenium Test Scripts

### Test Case 1: Valid Payment Processing

from selenium import webdriver  
from selenium.webdriver.common.keys import Keys  
  
# Set up WebDriver  
driver = webdriver.Chrome()  
driver.get("http://localhost/payment.php")  
  
# Fill in the payment form  
driver.find\_element("name", "card\_number").send\_keys("4111111111111111")  
driver.find\_element("name", "expiry").send\_keys("12/25")  
driver.find\_element("name", "cvv").send\_keys("123")  
driver.find\_element("name", "submit").click()  
  
# Verify success message  
assert "Payment Successful" in driver.page\_source  
  
driver.quit()

### Test Case 2: Invalid Card Number Handling

from selenium import webdriver  
from selenium.webdriver.common.keys import Keys  
  
# Set up WebDriver  
driver = webdriver.Chrome()  
driver.get("http://localhost/payment.php")  
  
# Enter invalid card details  
driver.find\_element("name", "card\_number").send\_keys("1234567890123456")  
driver.find\_element("name", "expiry").send\_keys("12/25")  
driver.find\_element("name", "cvv").send\_keys("123")  
driver.find\_element("name", "submit").click()  
  
# Verify error message  
assert "Invalid Card Number" in driver.page\_source  
  
driver.quit()

## 5. Conclusion

This document ensures that `payment.php` meets functional, UI, and security standards through manual and automated testing.