A blue circle with text and symbols

Description automatically generated

**AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)**

**FACULTY OF SCIENCE AND TECHNOLOGY**

Course Name: SOFTWARE QUALITY AND TESTING

Course Code: CSC4271

Section: C

Semester: Spring 2024-25

Supervisor Name: ABHIJIT BHOWMIK

|  |  |  |
| --- | --- | --- |
| SL | NAME | ID |
| 1 | MD. SHOHANUR RAHMAN SHOHAN | 22-46013-1 |
| 2 | FARJANA YESMIN OPI | 22-47018-1 |
| 3 | MD. ABU TOWSIF | 22-47019-1 |
| 4 | A. F. M. RAFIUL HASSAN | 22-47048-1 |

Submission date: May 19th, 2025

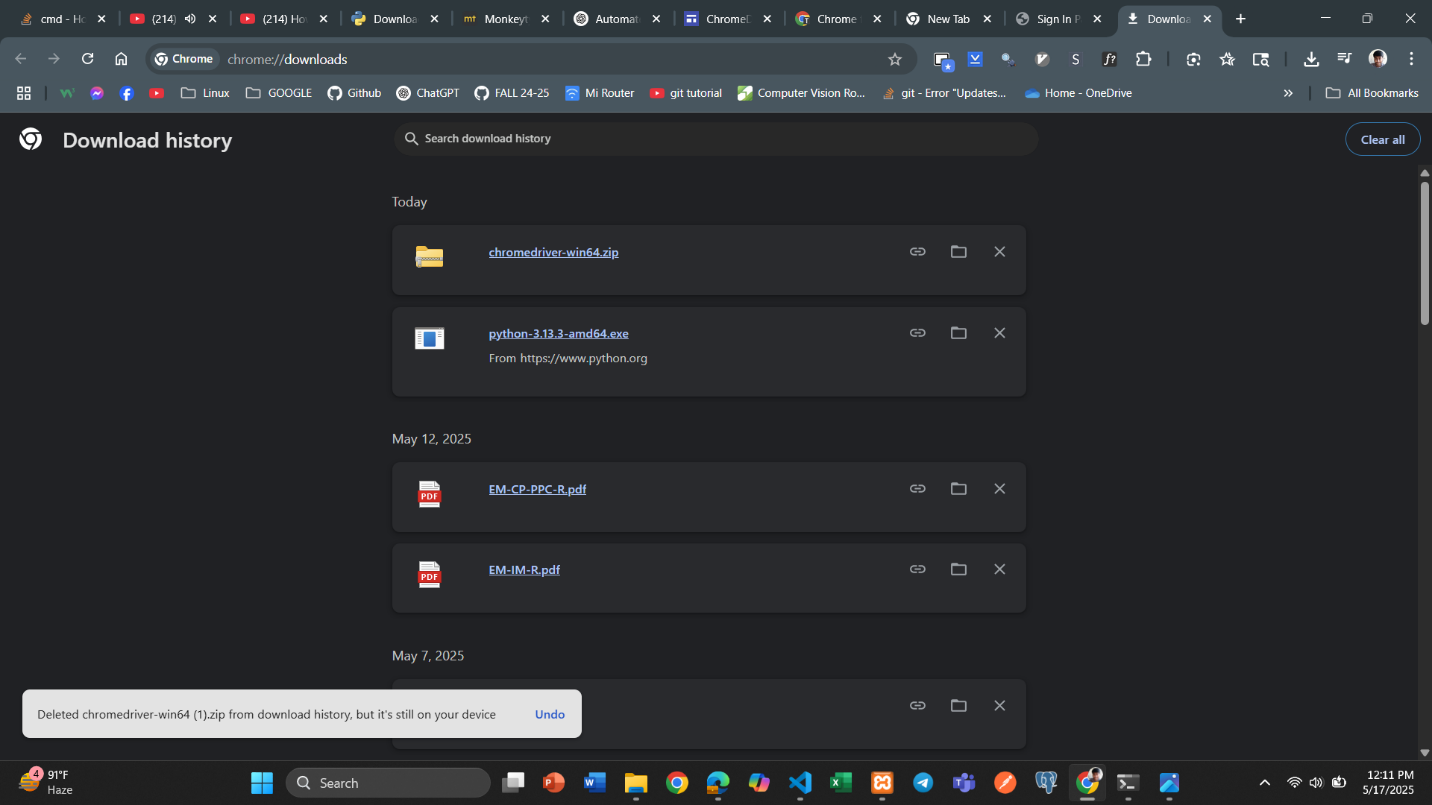
Automated Testing using Selenium(Python)

Figure 1: Downloading the correct version of Chrome Driver

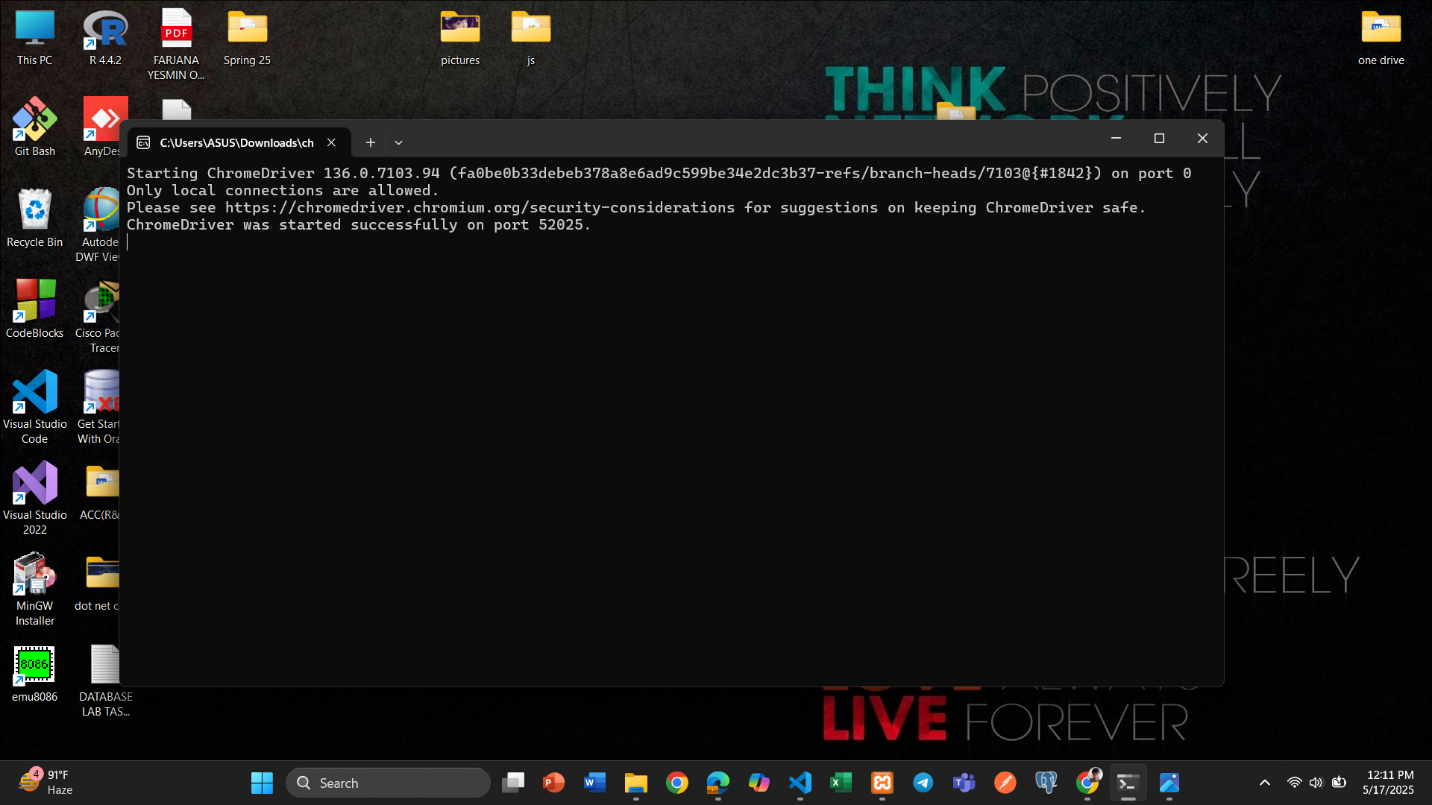


Figure 2: Chrome Driver installed and Chrome Driver started Selenium

## 

Figure 3: Checking the Chrome Driver in CLI

## Python Environment Setup

Figure 4: Installing required library in order to use Selenium

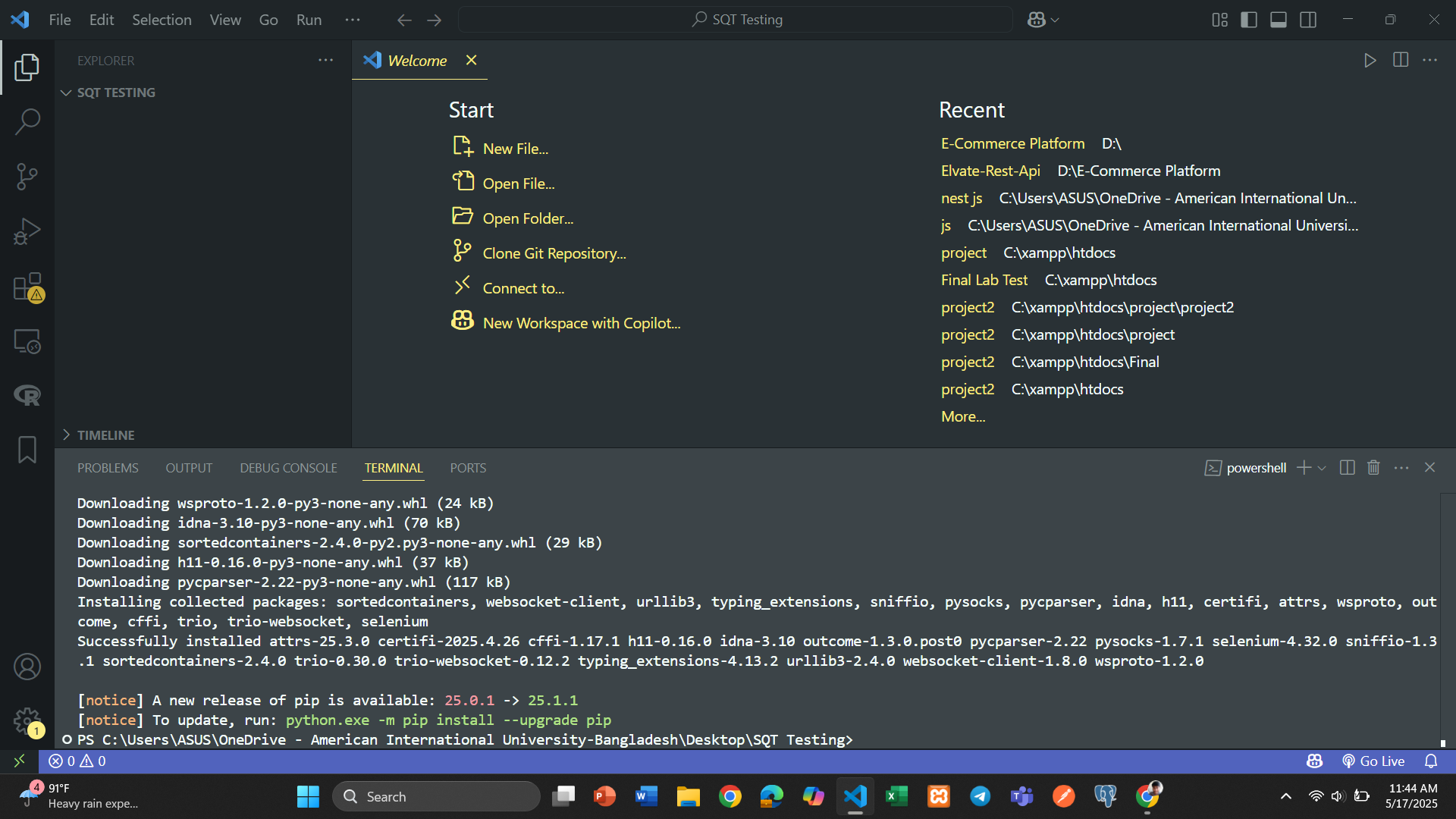


Figure 5: Required library has been installed in order to use Selenium

Test Case 1: Sign Up check

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Adventure | | | Test Designed by: A. F. M. RAFIUL HASSAN | | |
| Test Case ID: FR\_02 | | | Test Designed Date:5/17/2025 | | |
| Test Priority (Low, Medium, High): Medium | | | Test Executed by: A. F. M. RAFIUL HASSAN | | |
| Module Name: Signup Session | | | Test Execution Date:5/18/2025 | | |
| Test Title: Verify signup with random valid credentials | | |  | | |
| Description: Test the website signup page with random valid data. | | |  | | |
| Precondition: User is on the signup page and database is ready to accept new users.  Dependencies: Internet connection, server running, form field IDs correctly set. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Open Chrome browser 2. Navigate to the signup page 3. Enter random valid test data 4. Select user type 5. Click the signup button | Name: testuser  Email: testXXXX@example.com (randomized)  Password: testpass123  User type: User (radio button) | User account should be created successfully. Page may redirect to sign in page. | | As expected | Pass |
| Post Condition: User is registered successfully with randomized credentials. A new account is added to the database. The browser is closed properly. | | | | | |

## Python code using Selenium Library

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected\_conditions as EC

import time

import random

driver = webdriver.Chrome()

driver.get("http://localhost/project/view/opi\_features/auth\_feature/signup.html")

driver.maximize\_window()

WebDriverWait(driver, 10).until(EC.presence\_of\_element\_located((By.ID, "name")))

rand\_num = random.randint(1000, 9999)

test\_name = f"testuser"

test\_email = f"test{rand\_num}@example.com"

test\_password = "testpass123"

driver.find\_element(By.ID, "name").send\_keys(test\_name)

driver.find\_element(By.ID, "email").send\_keys(test\_email)

driver.find\_element(By.ID, "password").send\_keys(test\_password)

driver.find\_element(By.ID, "confirm\_password").send\_keys(test\_password)

driver.find\_element(By.ID, "type1").click()

driver.save\_screenshot("C:/Users/ASUS/Pictures/SS/signup\_before\_submit.png")

print('Screenshot saved: "signup\_before\_submit.png"')

submit\_button = driver.find\_element(By.XPATH, "//input[@type='submit' and @name='signup']")

submit\_button.click()

time.sleep(3)

driver.save\_screenshot("C:/Users/ASUS/Pictures/SS/signup\_after\_submit.png")

print('Screenshot saved: "signup\_after\_submit.png"')

driver.quit()



Figure 6: Sign Up before submit



Figure 7: Sign Up after submit

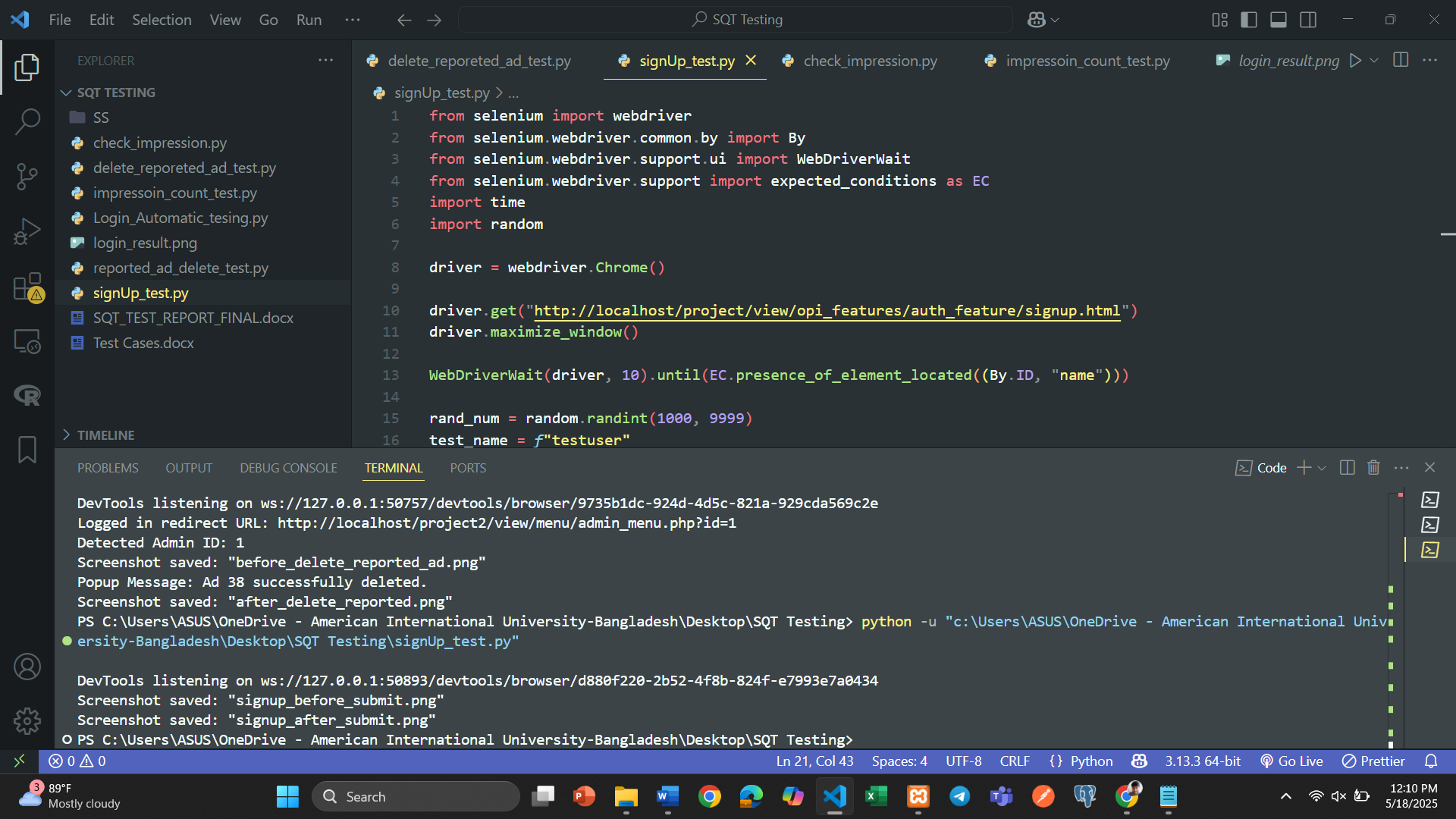


Figure 8: Terminal View

Test Case 2: login check

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Adventure | | | Test Designed by: MD. SHOHANUR RAHMAN SHOHAN | | |
| Test Case ID: FR\_01 | | | Test Designed Date: 5/17/2025 | | |
| Test Priority (Low, Medium, High): Medium | | | Test Executed by: MD. SHOHANUR RAHMAN SHOHAN | | |
| Module Name: login session | | | Test Execution Date: 5/18/2025 | | |
| Test Title: verify login functionality with username and password using Selenium | | |  | | |
| Description: Test the website login page using Selenium automation script | | |  | | |
| Precondition: User has access to localhost login page with valid credentials  Dependencies: Chrome WebDriver installed and reachable via system path | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Open Chrome browser 2. Navigate to login page 3. Enter valid username 4. Enter valid password 5. Click the submit button | Username: advertiser Password: 123456 | User should be logged in successfully and redirected to the menu page. | | As expected | Pass |
| Post Condition: User is validated and logged in successfully if credentials are correct. Browser session is closed automatically. | | | | | |

## Python code using Selenium Library

from selenium import webdriver

from selenium.webdriver.common.by import By

import time

driver = webdriver.Chrome()

driver.get("http://localhost/project2/view/auth\_feature/signin.html")

driver.maximize\_window()

time.sleep(2)

username\_input = driver.find\_element(By.ID, "username")

password\_input = driver.find\_element(By.ID, "password")

submit\_button = driver.find\_element(By.XPATH, "//input[@type='submit' and @name='login']")

username\_input.send\_keys("advertiser")

password\_input.send\_keys("123456")

submit\_button.click()

time.sleep(3)

current\_url = driver.current\_url

print("Current URL after login attempt:", current\_url)

if "menu" in current\_url:

    screenshot\_name = "login\_successful.png"

    print("Login successful")

else:

    screenshot\_name = "login\_failed.png"

    print("Login failed")

screenshot\_path = *f*"C:/Users/ASUS/Pictures/SS/{screenshot\_name}"

driver.save\_screenshot(screenshot\_path)

print(*f*"Screenshot saved as: {screenshot\_path}")

driver.quit()

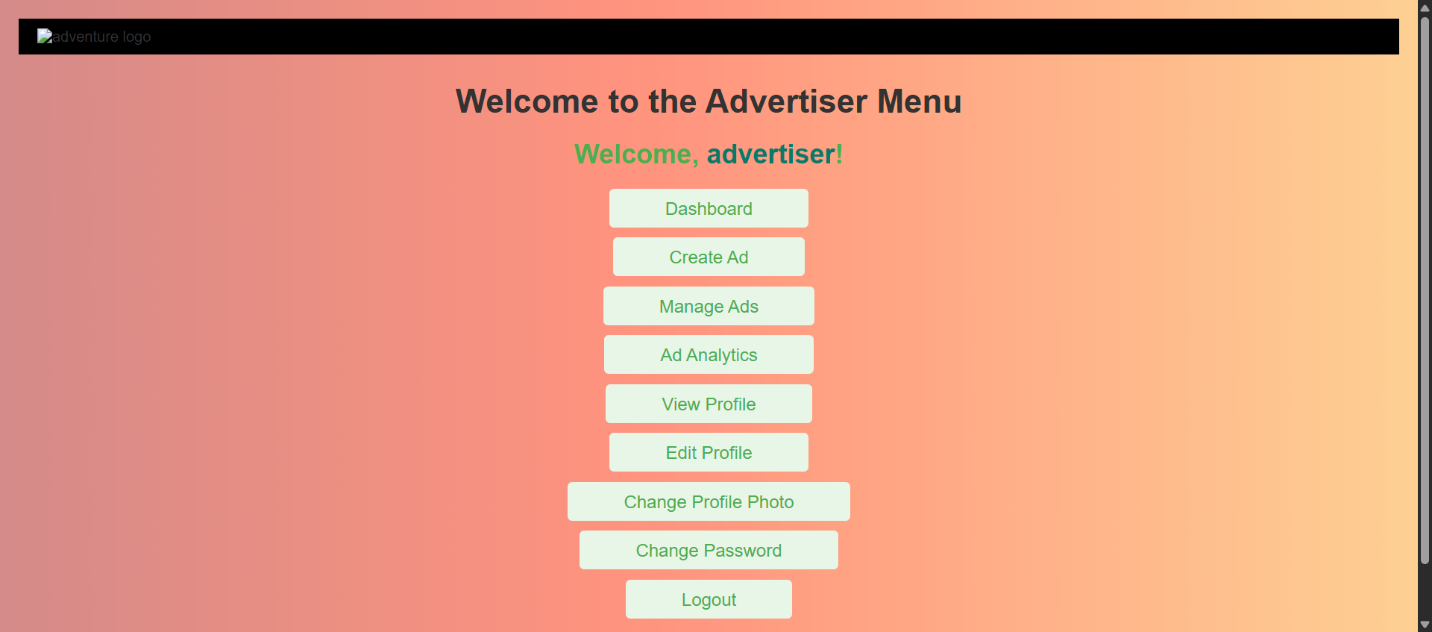


Figure 9: Login Successful



Figure 10: Login Failed

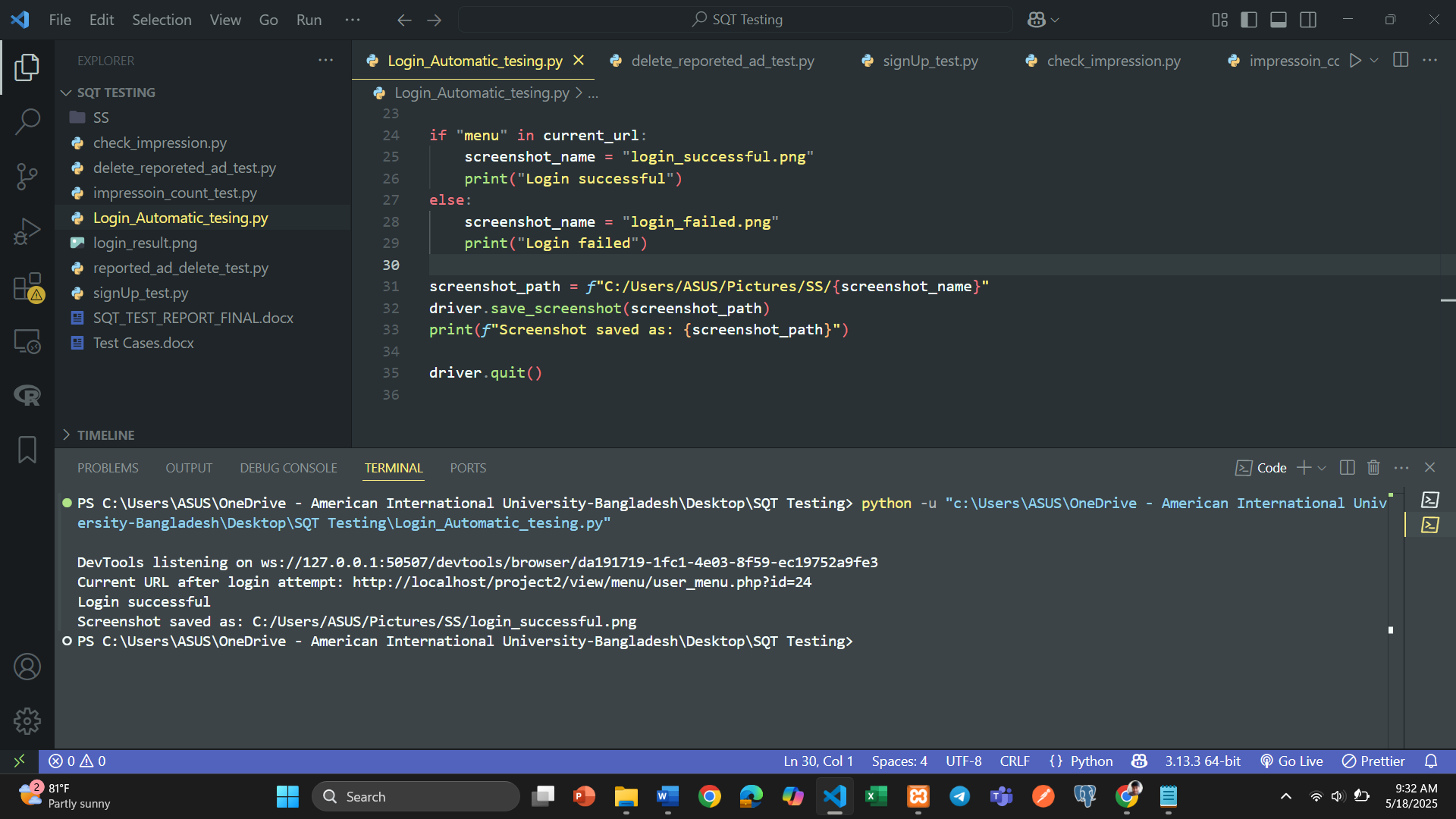


Figure 11: Terminal View For Login Successful

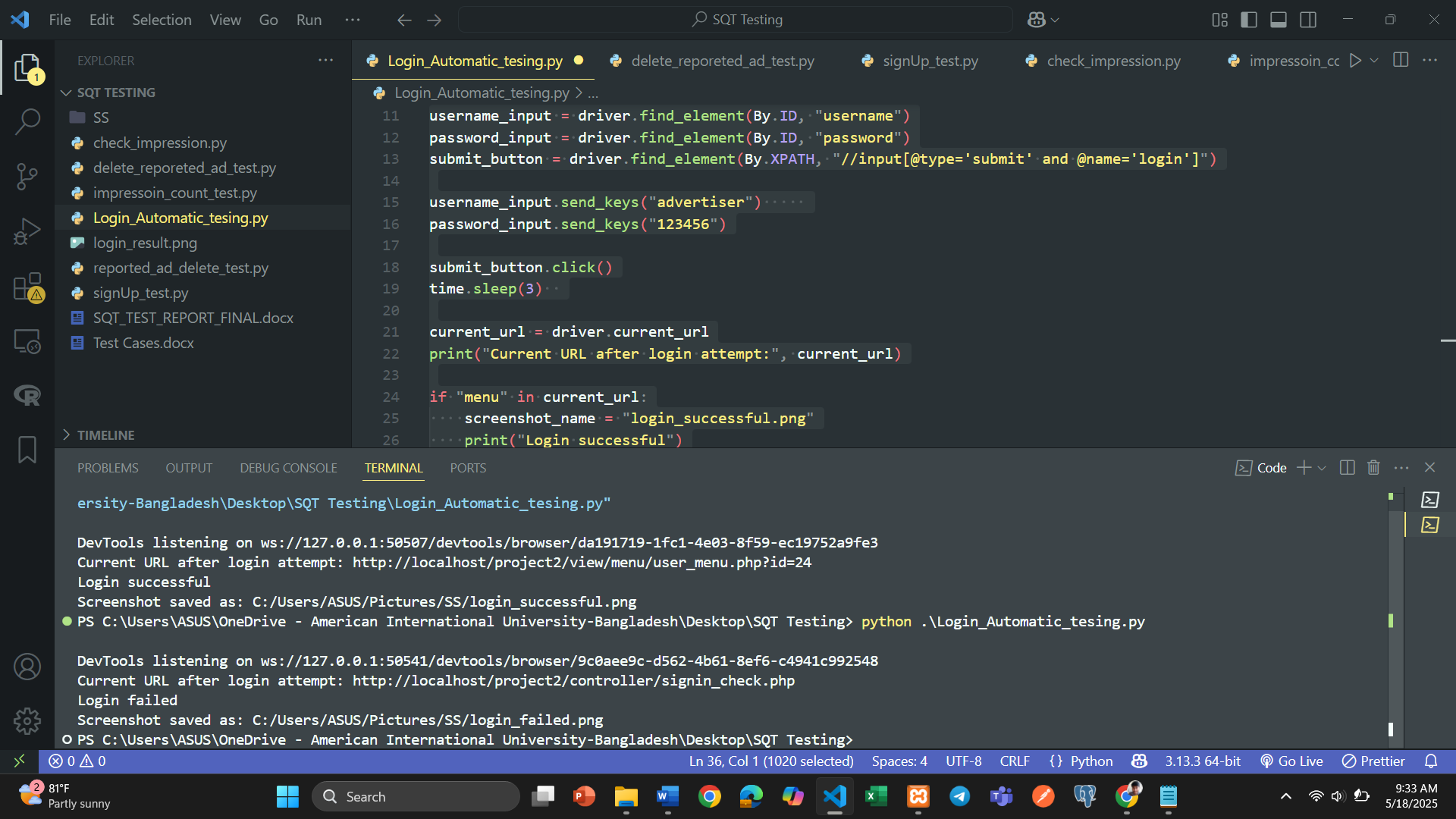


Figure 12: Terminal View For Login Failed

Test Case 3: Delete Reported Ad Functionality

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Adventure | | | Test Designed by: MD. ABU TOWSIF | | |
| Test Case ID: FR\_03 | | | Test Designed Date: 5/17/2025 | | |
| Test Priority (Low, Medium, High): Medium | | | Test Executed by: MD. ABU TOWSIF | | |
| Module Name: Reported Ads | | | Test Execution Date: 5/18/2025 | | |
| Test Title: Verify deletion of a reported ad with confirmation popup using Selenium | | |  | | |
| Description: Test deleting a reported ad and confirm the success message via the confirmation popup. | | |  | | |
| Precondition: User must have access to the reported ads page with at least one reported ad present. Chrome WebDriver should be installed and accessible in the system path.  Dependencies: Chrome WebDriver installed and accessible via system path. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1.Open Sign in page  2. Sign in as Admin  2. Open browser and navigate to reported ads page  3. Wait for ads to load  4. Click first delete button  5. Confirm deletion  6. Verify success message  7. Close popup  8. Close browser | User Name: adminadmin  Password: 110918 | Page loads with ads. Deletion confirmed by success message. Browser closes. | | As expected | Pass |
| Post Condition: The reported ad is deleted, and the browser session is closed. | | | | | |

## Python code using Selenium Library

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected\_conditions as EC

import time

import re

driver = webdriver.Chrome()

driver.get("http://localhost/project2/view/auth\_feature/signin.html")

driver.maximize\_window()

WebDriverWait(driver, 10).until(EC.presence\_of\_element\_located((By.ID, "username")))

driver.find\_element(By.ID, "username").send\_keys("adminadmin")

driver.find\_element(By.ID, "password").send\_keys("110918")

driver.find\_element(By.XPATH, "//input[@type='submit' and @name='login']").click()

WebDriverWait(driver, 10).until(EC.url\_contains("admin\_menu.php"))

admin\_url = driver.current\_url

print("Logged in redirect URL:", admin\_url)

match = re.search(r'id=(\d+)', admin\_url)

admin\_id = match.group(1) if match else "1"

print("Detected Admin ID:", admin\_id)

reported\_ad\_url = f"http://localhost/project/view/tishat\_features/report\_ads/reported\_ads.php?id={admin\_id}"

driver.get(reported\_ad\_url)

WebDriverWait(driver, 10).until(

EC.presence\_of\_element\_located((By.CLASS\_NAME, "reported\_ad\_info\_container"))

)

driver.save\_screenshot("C:/Users/ASUS/Pictures/SS/before\_delete\_fullflow.png")

print('Screenshot saved: "before\_delete\_fullflow.png"')

delete\_button = driver.find\_element(By.XPATH, "(//div[@class='delete\_btn']/button)[1]")

delete\_button.click()

WebDriverWait(driver, 5).until(EC.visibility\_of\_element\_located((By.ID, "confirmPopup")))

WebDriverWait(driver, 5).until(EC.element\_to\_be\_clickable((By.ID, "confirmYes"))).click()

popup\_message = WebDriverWait(driver, 10).until(EC.visibility\_of\_element\_located((By.ID, "popupMessage")))

print("Popup Message:", popup\_message.text)

time.sleep(1)

driver.save\_screenshot("C:/Users/ASUS/Pictures/SS/after\_delete\_fullflow.png")

print('Screenshot saved: "after\_delete\_fullflow.png"')

WebDriverWait(driver, 5).until(EC.element\_to\_be\_clickable((By.ID, "confirmOk"))).click()

driver.quit()

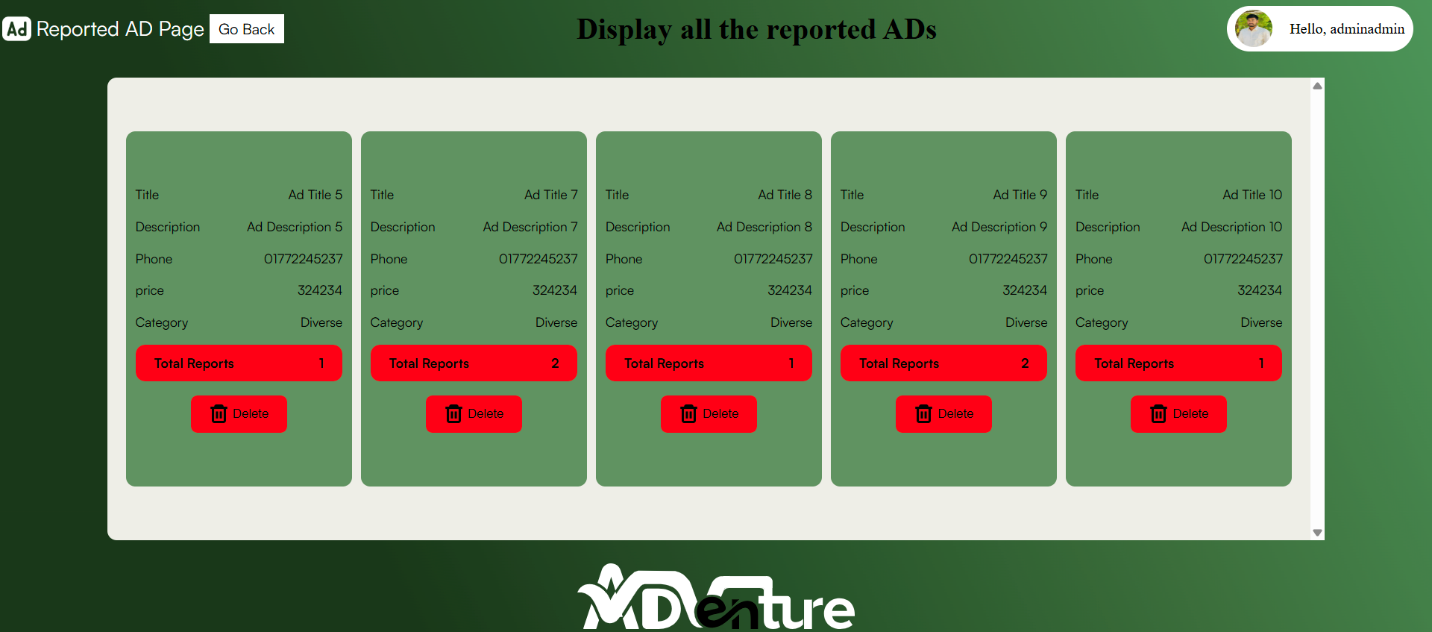


Figure 13: Before Delete reported ads



Figure 14: After Delete reported ads

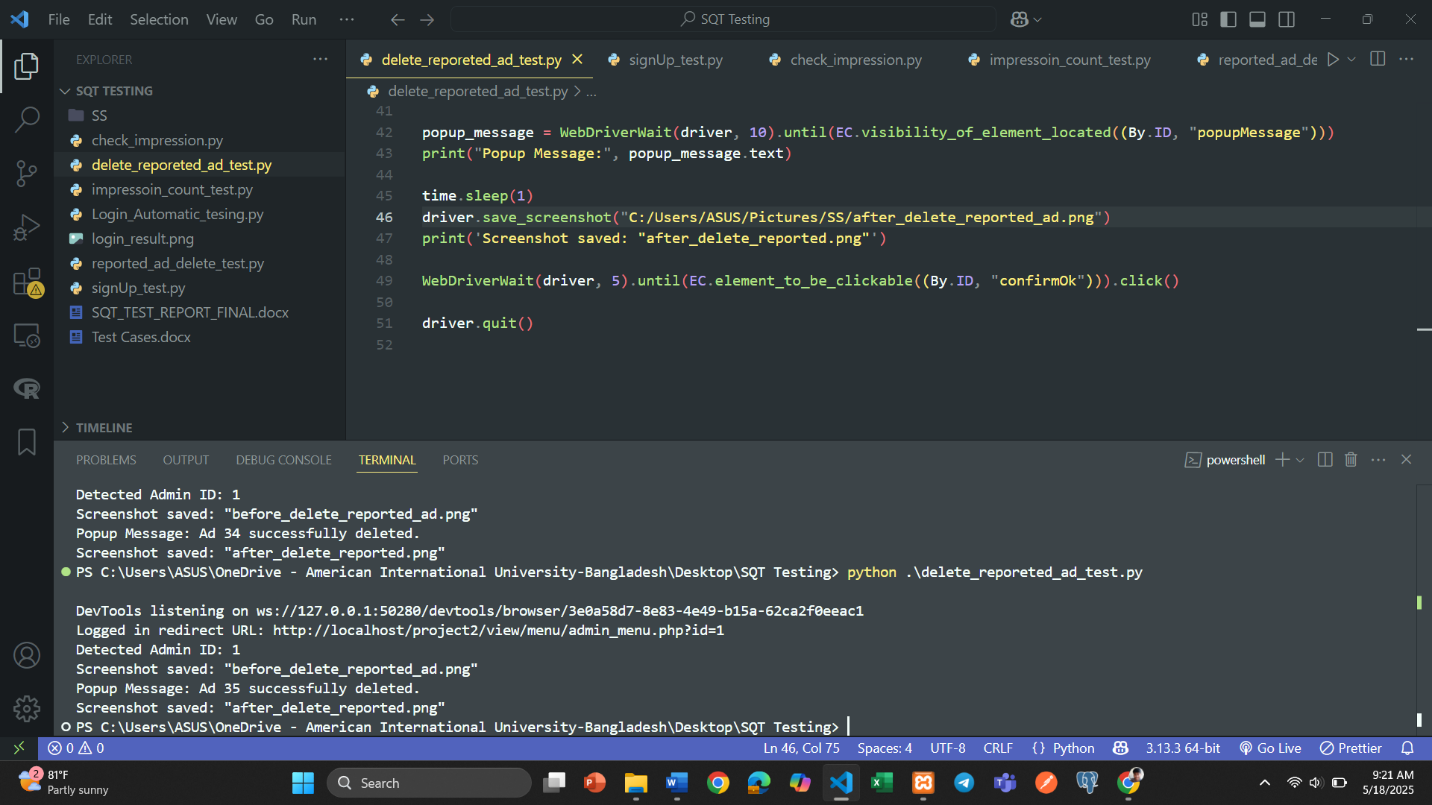


Figure 15: Terminal View for reported ad deletion test

Test Case 4: In landing page, impression button is working.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Adventure | | | Test Designed by: FARJANA YESMIN OPI | | |
| Test Case ID: FR\_04 | | | Test Designed Date: 5/17/2025 | | |
| Test Priority (Low, Medium, High): Medium | | | Test Executed by: FARJANA YESMIN OPI | | |
| Module Name: Ad Impression Tracking | | | Test Execution Date: 5/18/2025 | | |
| Test Title: Validate Ad Impression Count Increment on Click | | |  | | |
| Description: Ensure clicking the "Impression" button increases the ad impression count. | | |  | | |
| Precondition: Landing page is accessible; impression count is visible and functional.  Dependencies: Impression button must trigger backend update; internet/local server must be running. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Open Chrome browser 2. Navigate to landing page 3. Capture current impression count 4. Click the impression button 5. Wait and verify count | URL: http://  localhost/project/ | Impression count should increase by 1 after clicking the button. | | As expected | Pass |
| Post Condition: Impression count is successfully incremented by one, verifying the click was registered and the browser session ends properly. | | | | | |

## Python code using Selenium Library

from selenium import webdriver

from selenium.webdriver.common.by import By

import time

driver = webdriver.Chrome()

try:

driver.get('http://localhost/project/')

time.sleep(7)

driver.save\_screenshot("C:/Users/ASUS/Pictures/SS/before\_impression\_increase.png")

print(f'Screenshot saved: "before\_impression\_increase.png"')

impression\_stat\_div = driver.find\_element(By.CLASS\_NAME, 'ad\_impression\_class')

impression\_count\_elem = impression\_stat\_div.find\_element(By.TAG\_NAME, 'p')

count\_before = int(impression\_count\_elem.text.strip())

impression\_button = driver.find\_element(By.CLASS\_NAME, 'impression')

impression\_button.click()

time.sleep(3)

driver.save\_screenshot("C:/Users/ASUS/Pictures/SS/after\_impression\_increase.png")

print(f'Screenshot saved: "after\_impression\_increase.png"')

updated\_stat\_div = driver.find\_element(By.CLASS\_NAME, 'ad\_impression\_class')

updated\_count\_elem = updated\_stat\_div.find\_element(By.TAG\_NAME, 'p')

count\_after = int(updated\_count\_elem.text.strip())

print("Impression count before click:", count\_before)

print("Impression count after click: ", count\_after)

if count\_after > count\_before:

print("Success: Impression count increased.")

else:

print("Failed: Impression count did not increase.")

finally:

driver.quit()



Figure 17: After Impression Increase

Figure 16: Before Impression Increase

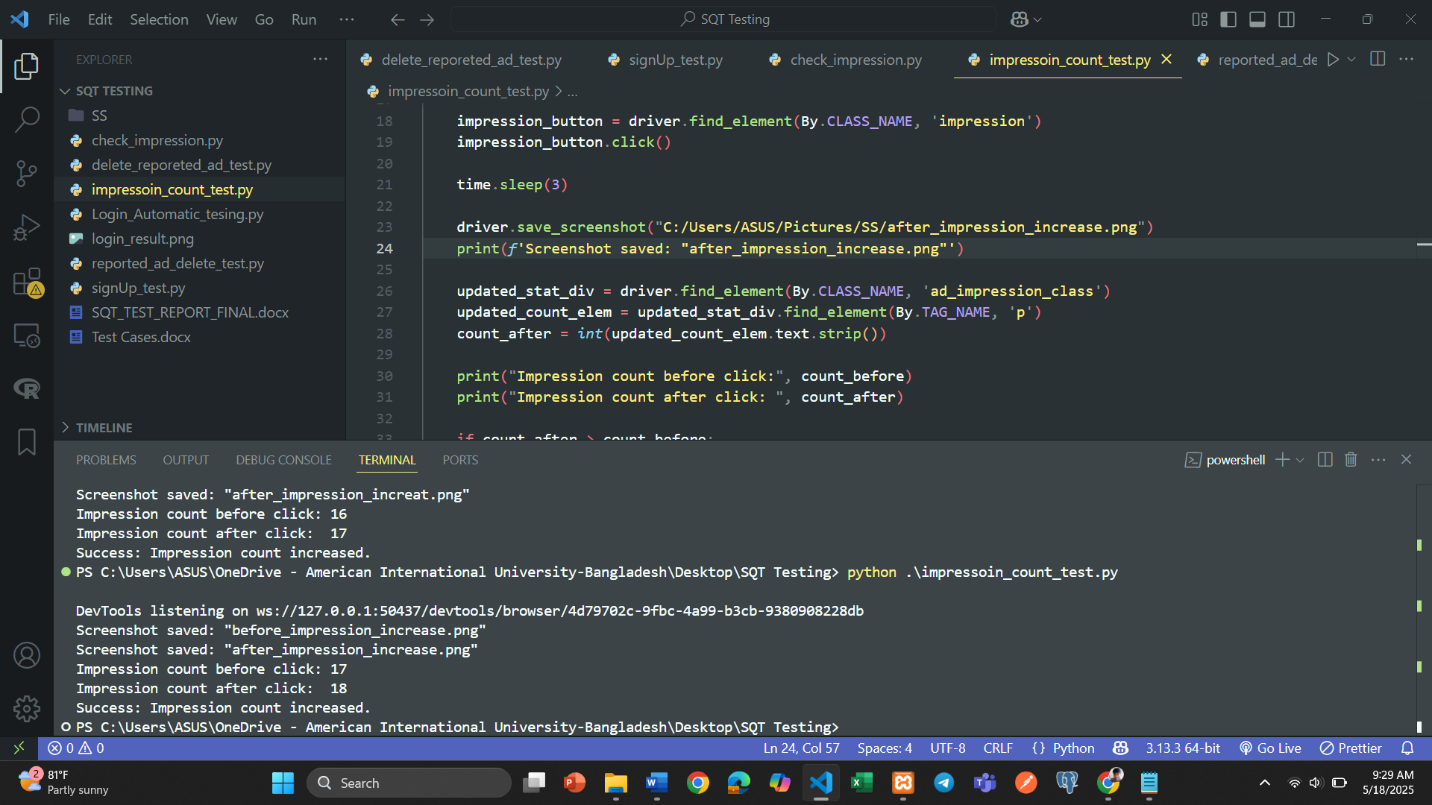


Figure 18: Terminal View for impression button test