

TUTORIAL 8

8. Automate the login functionality test case using Selenium.

Objective:

To **automate a test case** for login functionality using **Selenium WebDriver (Python)**.

Tools & Technologies Used:

- Language: **Python**
- Automation Library: **Selenium**
- Browser: **Chrome**
- Test Site: Use demo site like <http://parabank.parasoft.com>

Step-by-Step Guide

Step 1: Install Required Tools

1. Install Python (from python.org)

Install Selenium:

```
pip install selenium
```

2. Download **ChromeDriver** compatible with your Chrome browser:
<https://sites.google.com/chromium.org/driver/>
3. Place chromedriver.exe in your project folder or system PATH.

Step 2: Write Test Script for Login

Scenario:

- Open login page
- Enter username: john
- Enter password: demo
- Click Login
- Verify successful login (e.g., check welcome message or redirect)

Python Code Using Selenium

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
import time

# Setup
driver = webdriver.Chrome() # Launch Chrome
driver.maximize_window()

# Step 1: Open Login Page
driver.get("http://parabank.parasoft.com/parabank/index.htm")

# Step 2: Enter Username
username = driver.find_element(By.NAME, "username")
username.send_keys("john") # Replace with valid user if needed

# Step 3: Enter Password
password = driver.find_element(By.NAME, "password")
password.send_keys("demo") # Replace with valid password

# Step 4: Click Login
login_button = driver.find_element(By.XPATH, "//input[@value='Log In']")
login_button.click()

# Step 5: Wait and Verify Login
time.sleep(3)
```

```
# Verify: Check for welcome message
try:
    welcome_message = driver.find_element(By.XPATH, "//h1[text()='Accounts Overview']")
    print("✅ Login Test Passed: Accounts Overview page displayed.")
except:
    print("❌ Login Test Failed: Accounts Overview page not found.")

# Close browser
driver.quit()
```

Step 3: Understand Key Selenium Commands

Selenium Code	Purpose
<code>webdriver.Chrome()</code>	Launches Chrome browser
<code>driver.get(URL)</code>	Opens web page
<code>find_element(By.X, "locator")</code>	Finds element (By ID, NAME, XPATH, etc.)
<code>.send_keys("value")</code>	Types into input field
<code>.click()</code>	Clicks on button
<code>driver.quit()</code>	Closes the browser

Step 4: Customize and Extend

- Add negative test case (wrong password)
- Add assertions using unittest or pytest
- Export results to a log or report file

Step 5: Expected Output

If successful:

Login Test Passed: Accounts Overview page displayed.

✗ If failed:

✗ Login Test Failed: Accounts Overview page not found.

Assignment for Students

Task	Description
Test 1	Automate valid login
Test 2	Automate invalid login and check for error message
Test 3	Add screenshot on failure
Bonus	Use unittest framework to create multiple test cases

Optional: Add Screenshot on Failure

except:

```
print("✗ Login Test Failed")  
driver.save_screenshot("login_failure.png")
```