TASK 3:

**Automated** **Login Test for** **a Web** **Application**

Write a test suite in library of your choice that automates testing login functionality  
on the given website. You may use other websites too for this task.

You have to Include positive test cases (valid credentials) and negative case (invalid credentials, empty fields).

import pytest

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.keys import Keys

# Replace with the actual URL of the website

BASE\_URL = "https://www.example.com"

# Replace with the actual username and password fields

USERNAME\_FIELD = "username"

PASSWORD\_FIELD = "password"

# Replace with the actual login button locator

LOGIN\_BUTTON = "login\_button"

# Positive test cases with valid credentials

@pytest.mark.parametrize("username, password", [

("valid\_user1", "valid\_password1"),

("valid\_user2", "valid\_password2")

])

def test\_login\_success(username, password):

driver = webdriver.Chrome() # Replace with your preferred driver

driver.get(BASE\_URL)

# Locate and enter username

username\_field = driver.find\_element(By.ID, USERNAME\_FIELD)

username\_field.send\_keys(username)

# Locate and enter password

password\_field = driver.find\_element(By.ID, PASSWORD\_FIELD)

password\_field.send\_keys(password)

# Click the login button

login\_button = driver.find\_element(By.ID, LOGIN\_BUTTON)

login\_button.click()

# Assert successful login (e.g., check for a welcome message)

assert "Welcome" in driver.page\_source

driver.quit()

# Negative test cases with invalid credentials

@pytest.mark.parametrize("username, password, expected\_error", [

("", "", "Please enter both username and password."),

("invalid\_user", "invalid\_password", "Invalid credentials."),

("valid\_user", "", "Please enter password."),

("", "valid\_password", "Please enter username.")

])

def test\_login\_failure(username, password, expected\_error):

driver = webdriver.Chrome()

driver.get(BASE\_URL)

# Enter username and password

username\_field = driver.find\_element(By.ID, USERNAME\_FIELD)

username\_field.send\_keys(username)

password\_field = driver.find\_element(By.ID, PASSWORD\_FIELD)

password\_field.send\_keys(password)

# Click the login button

login\_button = driver.find\_element(By.ID, LOGIN\_BUTTON)

login\_button.click()

# Assert the error message

assert expected\_error in driver.page\_source

driver.quit()

**Explanation:**

1. **Import necessary libraries:**
   * pytest for test framework
   * webdriver from selenium for browser interaction
   * By and Keys from selenium.webdriver.common for element locating and key actions
2. **Define constants:**
   * BASE\_URL: URL of the website
   * USERNAME\_FIELD: ID or other locator of the username field
   * PASSWORD\_FIELD: ID or other locator of the password field
   * LOGIN\_BUTTON: ID or other locator of the login button
3. **Positive test cases:**
   * @pytest.mark.parametrize decorator allows testing with multiple sets of valid credentials.
   * Each test case:
     + Creates a webdriver instance.
     + Navigates to the website.
     + Locates and enters username and password.
     + Clicks the login button.
     + Asserts successful login by checking for a welcome message or any other indicator.
     + Closes the browser.
4. **Negative test cases:**
   * @pytest.mark.parametrize decorator allows testing with various invalid credentials and empty fields.
   * Each test case:
     + Creates a webdriver instance.
     + Navigates to the website.
     + Enters username and password (or empty fields).
     + Clicks the login button.
     + Asserts the expected error message on the page.
     + Closes the browser.

**To run the tests:**

* Save the code as a Python file (e.g., test\_login.py).
* Open a terminal or command prompt.
* Navigate to the directory where you saved the file.
* Run the command pytest test\_login.py.