NAME : PRATIKSHA SHIVSHARNE

ROLL NO : D2235008

**Functional Test Cases**

|  |  |  |  |
| --- | --- | --- | --- |
| **TC ID** | **Test Scenario** | **Steps** | **Expected Result** |
| TC\_01 | Valid Login | 1. Enter **valid username & password** 2. Click **Login** | User should be redirected to the **dashboard** |
| TC\_02 | Invalid Login (Wrong Password) | 1. Enter **correct username** but **wrong password** 2. Click **Login** | Error message: **"Invalid credentials"** |
| TC\_03 | Invalid Login (Non-existent User) | 1. Enter a **random/non-existent username** 2. Enter **any password** 3. Click **Login** | Error: **"User does not exist"** |
| TC\_04 | Empty Username & Password | 1. Keep **username & password empty** 2. Click **Login** | Error: **"Fields cannot be empty"** |
| TC\_05 | Password Masking | 1. Enter **password** 2. Check if it appears as \*\*\*\* | Password should be **masked** |
| TC\_06 | Forgot Password Link | 1. Click on **Forgot Password?** | Should redirect to **password recovery page** |
| TC\_07 | Case Sensitivity | 1. Enter **correct username** but use different **letter cases** (e.g., Admin vs. admin) | System should be **case-sensitive** as per requirements |
| TC\_08 | Session Timeout Test | 1. Log in successfully 2. Remain inactive for **X minutes** | User should be **logged out automatically** |
| TC\_09 | Login Button Disabled | 1. Enter **only username or only password** 2. Observe the **Login button** | It should remain **disabled** until both fields are filled |
| TC\_10 | Logout Functionality | 1. Log in successfully 2. Click **Logout** | User should be redirected to the **login page** |

**Selenium Framework Code for Login Page Testing**

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.keys import Keys

import time

def test\_valid\_login():

driver = webdriver.Chrome()

driver.get("http://hotel-management-system.com/login")

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

driver.find\_element(By.ID, "password").send\_keys("password123")

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

time.sleep(2)

assert "dashboard" in driver.current\_url

driver.quit()

def test\_invalid\_login():

driver = webdriver.Chrome()

driver.get("http://hotel-management-system.com/login")

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

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

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

time.sleep(2)

error\_message = driver.find\_element(By.ID, "error").text

assert error\_message == "Invalid credentials"

driver.quit()

def test\_empty\_fields():

driver = webdriver.Chrome()

driver.get("http://hotel-management-system.com/login")

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

time.sleep(2)

error\_message = driver.find\_element(By.ID, "error").text

assert error\_message == "Fields cannot be empty"

driver.quit()

def test\_password\_masking():

driver = webdriver.Chrome()

driver.get("http://hotel-management-system.com/login")

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

password\_field.send\_keys("password123")

assert password\_field.get\_attribute("type") == "password"

driver.quit()

if \_\_name\_\_ == "\_\_main\_\_":

test\_valid\_login()

test\_invalid\_login()

test\_empty\_fields()

test\_password\_masking()