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

# Setup Chrome driver

driver = webdriver.Chrome()

driver.maximize\_window()

# Open Instagram login page

driver.get("https://www.instagram.com/accounts/login/")

# Wait until the username field is present (explicit wait)

wait = WebDriverWait(driver, 15)

username\_input = wait.until(EC.presence\_of\_element\_located((By.NAME, "username")))

password\_input = wait.until(EC.presence\_of\_element\_located((By.NAME, "password")))

# Enter login details

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

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

# Click the login button

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

login\_button.click()

# Wait to ensure the page loads after login

time.sleep(5)

# Check if redirected (Optional: Based on what’s loaded post login)

print("Login attempted.")

driver.quit()

### 

### 

### 

### 

### 

### **Aim:**

To analyze a sample web application project and identify the modules that are most appropriate for automation testing based on factors such as repetitiveness, stability, and business criticality.

### **Objective:**

* Understand how to break down a software project into modules.
* Determine which modules are ideal for automation testing.
* Learn to justify automation choices with technical reasons.

### **Tools & Technologies Used:**

* Selenium WebDriver
* Google Chrome Browser
* Visual Studio Code / Eclipse IDE
* Sample Project: **"Online Book Store"** web application

### **Theory:**

Automation testing is a process in which test cases are executed using automation tools, eliminating the need for manual intervention. Automating the right modules can save time and effort, especially for:

* Repetitive tasks
* Regression testing
* Critical workflows
* Data-driven testing

### **Key Criteria for Selecting Modules for Automation:**

| **Criteria** | **Description** |
| --- | --- |
| **Repetitive Tests** | Executed repeatedly (e.g., login, search). |
| **Stable Modules** | Code/UX doesn’t change often. |
| **Data-Driven Modules** | Requires many data inputs (e.g., checkout). |
| **High-Risk Modules** | Core business functionality (e.g., payment). |
| **Time-Consuming Cases** | Manual testing takes a lot of time. |

### **Sample Project for Analysis: Online Book Store**

The project allows users to browse, search, purchase books, and includes an admin panel for managing books.

### **Step-by-Step Module Analysis**

| **Module Name** | **Description** | **Frequency** | **Stability** | **Automation Suitability** | **Reason** |
| --- | --- | --- | --- | --- | --- |
| **User Registration** | Allows users to sign up with username, email, password | Low | Medium | ❌ Manual Preferred | One-time task per user |
| **User Login/Logout** | Allows users to securely log in and out | High | High | ✅ Yes | Used in almost all test flows |
| **Book Search** | Search for books by title, author, genre | High | High | ✅ Yes | Repetitive with multiple input data |
| **Add to Cart** | Adds selected book to shopping cart | Medium | High | ✅ Yes | Simple workflow, consistent |
| **Remove from Cart** | Remove books from cart before purchase | Medium | High | ✅ Yes | Useful for negative testing |
| **Checkout** | Enter address, payment, confirm order | Medium | High | ✅ Yes | Critical for business |
| **View Order History** | Displays previous orders made by user | Medium | High | ✅ Yes | Data-driven and repetitive |
| **Admin: Add Books** | Admin can add new books | Medium | High | ✅ Yes | Structured CRUD operation |
| **Admin: Delete Books** | Admin can remove books from inventory | Medium | High | ✅ Yes | Easy to automate |
| **Captcha** | Prevents bots from logging in/registering | Low | Varies | ❌ No | Difficult for automation tools |
| **UI Visual Check** | Checks visual layout, spacing, color schemes | Medium | Low | ❌ No | Needs human visual confirmation |

### **Modules Best Suited for Automation:**

1. **Login/Logout**
2. **Search Functionality**
3. **Add to Cart / Remove from Cart**
4. **Checkout Flow**
5. **View Order History**
6. **Admin Book Management (Add/Delete)**

### **Modules Better for Manual Testing:**

1. **User Registration**
2. **Captcha**
3. **UI/UX Visual Testing**
4. **Email Confirmation Testing**

### **Justification Example for Automation:**

📌 **Login/Logout Module:**

* Executed before any other action
* Simple steps: Enter credentials → Click login → Validate dashboard
* Can be reused in every test case  
   ✅ Best for automation using Selenium.

📌 **Search Module:**

* Test inputs like “Harry Potter”, “Science Fiction”, “J.K. Rowling”
* Output changes dynamically
* Validating search results with expected data  
   ✅ Automation can test multiple search inputs using Data-Driven Testing in Selenium.