**Problem Statement**

Web Application:

- User Registration with valid and invalid inputs.

- User Login with valid and invalid credentials.

- Basic functionality of the User Dashboard and Navigation

**Solution**

**Setting up Selenium (C#)**

**1. Create a new test project in Visual Studio**:

* Open Visual Studio.
* Create a new project -> Select "NUnit Test Project" or "MSTest Test Project".
* Name your project and click "Create".

**Install Selenium WebDriver NuGet packages**:

* Right-click on your project in the Solution Explorer.
* Select "Manage NuGet Packages".
* Search for "Selenium.WebDriver" and install it.
* Search for "Selenium.WebDriver.ChromeDriver" and install it (or the driver for your preferred browser).

**2. Basic Selenium Test for User Registration with Valid and Invalid Inputs**

using NUnit.Framework;

using OpenQA.Selenium;

using OpenQA.Selenium.Chrome;

using System;

public class UserRegistrationTests

{

private IWebDriver driver;

[SetUp]

public void Setup()

{

driver = new ChromeDriver();

driver.Manage().Timeouts().ImplicitWait = TimeSpan.FromSeconds(10); driver.Manage().Window.Maximize();

}

//User Registration with valid input

[Test]

public void TestUserRegistrationWithValidInputs()

{

//Intialising the WebDriver

IWebDriver driver = new ChromeDriver();

//Navigating to bookstoreapp registrstation

driver.Navigate().GoToUrl("http://bookstoreapp.com/register");

//User with valid username,password and email

driver.FindElement(By.Id("username")).SendKeys("testuser");

driver.FindElement(By.Id("password")).SendKeys("Password123!");

driver.FindElement(By.Id("email")).SendKeys("testuser@example.com");

driver.FindElement(By.Id("submit")).Click();

//validating if user is getting success message

var successMessage = driver.FindElement(By.Id("successMessage")).Text;

Assert.Equal("Registration successful", successMessage);

**}**

[Test]

public void TestUserRegistrationWithInValidInputs()

{

IWebDriver driver = new ChromeDriver();

//Navigating to bookstoreapp registrstation

driver.Navigate().GoToUrl("http://bookstoreapp.com/register");

//User with valid username,password and invalid email

driver.FindElement(By.Id("username")).SendKeys("testuser");

driver.FindElement(By.Id("password")).SendKeys("Password123!");

driver.FindElement(By.Id("email")).SendKeys("testuserexample");

driver.FindElement(By.Id("submit")).Click();

//validating if user is getting invalid mail message

var InvalidMail= driver.FindElement(By.Id("Invalidmail")).Text;

Assert.Equal("Invalid mail", successMessage);

}

[TearDown]

//Quitting the driver

public void TearDown()

{

driver.Quit();

}

}

**3. Basic Selenium Test for User Login with Valid and Invalid Inputs**

public class UserLoginTests

{

private IWebDriver driver;

[SetUp]

public void Setup()

{

driver = new ChromeDriver();

driver.Manage().Timeouts().ImplicitWait = TimeSpan.FromSeconds(10); driver.Manage().Window.Maximize();

}

//User Login with valid input

[Test]

public void TestUserLoginWithValidInputs()

{

//Intialising the WebDriver

IWebDriver driver = new ChromeDriver();

//Navigating to bookstoreapp login

driver.Navigate().GoToUrl("http://bookstoreapp.com/Login");

//User with registered valid username,password driver.FindElement(By.Id("username")).SendKeys("testuser");

driver.FindElement(By.Id("password")).SendKeys("Password123!");

driver.FindElement(By.Id("submit")).Click();

//validating if user is getting success message

var LoginMessage = driver.FindElement(By.Id(" LoginMessage ")).Text;

Assert.Equal("Login successful", LoginMessage);

}

//User Login with Invalid input

[Test]

public void TestUserLoginWithInValidInputs()

{

//Intialising the WebDriver

IWebDriver driver = new ChromeDriver();

//Navigating to bookstoreapp login

driver.Navigate().GoToUrl("http://bookstoreapp.com/Login");

//User with registered valid username and Invalid password driver.FindElement(By.Id("username")).SendKeys("testuser");

driver.FindElement(By.Id("password")).SendKeys(“!@#$%\*(");

driver.FindElement(By.Id("submit")).Click();

//validating if user is getting Invalid username and password

var InvaliduserPass= driver.FindElement(By.Id(" InvaliduserPass ")).Text;

Assert.Equal("Invalid User/Passl", InvaliduserPass);

}

[TearDown]

//Quitting the driver

public void TearDown()

{

driver.Quit();

}

**3. Basic Selenium Test for Data Display and Navigation**

public class DashboardTests

{

private IWebDriver driver;

[SetUp]

public void Setup()

{

driver = new ChromeDriver();

driver.Manage().Timeouts().ImplicitWait = TimeSpan.FromSeconds(10);

driver.Manage().Window.Maximize();

// Login steps

driver.Navigate().GoToUrl("http://bookstorewebapp.com/login");

driver.FindElement(By.Id("username")).SendKeys("testuser");

driver.FindElement(By.Id("password")).SendKeys("password");

driver.FindElement(By.Id("loginButton")).Click();

// Navigate to User Dashboard

driver.Navigate().GoToUrl("http://bookstorewebapp.com/dashboard");

}

[Test]

public void VerifyDataDisplay()

{

// Check if the user's name is displayed

IWebElement userNameElement = driver.FindElement(By.Id("userName"));

Assert.IsTrue(userNameElement.Displayed, "User name is not displayed");

// Check if the recent activity section is displayed

IWebElement recentActivityElement = driver.FindElement(By.Id("recentActivity"));

Assert.IsTrue(recentActivityElement.Displayed, "Recent activity section is not displayed");

// Check if the recommended books section is displayed

IWebElement recommendedBooksElement = driver.FindElement(By.Id("recommendedBooks"));

Assert.IsTrue(recommendedBooksElement.Displayed, "Recommended books section is not displayed");

}

[Test]

public void VerifyNavigation()

{

// Navigate to Profile

IWebElement profileLink = driver.FindElement(By.Id("profileLink"));

profileLink.Click();

Assert.AreEqual("http://bookstorewebapp.com/profile", driver.Url);

// Navigate to Settings

IWebElement settingsLink = driver.FindElement(By.Id("settingsLink"));

settingsLink.Click();

Assert.AreEqual("http://bookstorewebapp.com/settings", driver.Url);

// Navigate back to Dashboard

IWebElement dashboardLink = driver.FindElement(By.Id("dashboardLink"));

dashboardLink.Click();

Assert.AreEqual("http://bookstorewebapp.com/dashboard", driver.Url);

}

[TearDown]

public void TearDown()

{

driver.Quit();

}

}

}

### **Running the Tests**

1. **Build the project**: Ensure there are no errors in the code.
2. **Run the tests**: Use the Test Explorer in Visual Studio to run the tests. Ensure the tests pass and verify that data is displayed correctly and navigation works as expected on the User Dashboard.