

Functional Testing (Automated Testing): [2 Hours]

- Write the Automated Test cases by using Selenium for your application. And generate the test report using NUnit report/Selenium IDE.

Program.cs

```
using OpenQA.Selenium.Chrome;
using OpenQA.Selenium;
using Utils.Reports;
namespace SeleniumAssessment
{
    internal class Program
    {
        static void Main(string[] args)
        {

            IWebDriver chrome = new ChromeDriver();
            LoginPage loginPage = new LoginPage(chrome);
            loginPage.CheckLoginButtonwithvalidinput();
            ////loginPage.Check_Login_Button_with_invalid_account_input();
            loginPage.Check_login_with_empty_data_return_fails();
            loginPage.Check_Login_Button_with_invalid_password_input();
            loginPage.Check_work_for_Register_button();
            RegisterPage registerPage = new RegisterPage(chrome);
            registerPage.Check_work_for_Login_button();
            // registerPage.Check_register_with_empty_data_return_fails();
            registerPage.CheckRegisterButtonwithvalidinput();
            registerPage.Check_Register_Button_with_exists_data_input();
            HomePage homePage = new HomePage(chrome);
            homePage.Check_list_movie_button_redirect_to_list_page();
            homePage.Check_add_movie_button_redirect_to_add_page();
            homePage.Check_value_in_total_movie_not_null();
            CreatePage createPage = new CreatePage(chrome);
            createPage.Check_work_for_View_button();
            createPage.CheckAddButtonwithvalidinput();
            createPage.Check_add_movie_with_empty_data_return_fails();
            ListPage listPage = new ListPage(chrome);
            listPage.Check_work_for_Delete_button();
            listPage.Check_work_for_Update_button();
            listPage.Check_work_for_Add_button();
            UpdatePage updatePage = new UpdatePage(chrome);
            updatePage.CheckUpdateButtonwithvalidinput();
            updatePage.Check_update_movie_with_empty_data_return_fails();
            updatePage.Check_work_for_View_button();
```

```

        DeletePage deletePage = new DeletePage(chrome);
        deletePage.Check_delete_button_display_Dialog_box();
        deletePage.Check_dialog_box_Delete_button_delete_element();
        deletePage.Check_dialog_box_Cancel_button_return_list_page();

        ExtentReporting.EndReporting();

    }
}

```

RegisterPage.cs

```

using OpenQA.Selenium;
using OpenQA.Selenium.Support.UI;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Utils.Reports;

namespace SeleniumAssessment
{
    internal class RegisterPage
    {
        private readonly IWebDriver driver;

        public RegisterPage(IWebDriver webDriver)
        {
            driver = webDriver;
        }

        public void CheckRegisterButtonwithvalidinput()
        {
            ExtentReporting.CreateTest("CheckRegisterButtonwithvalidinput");

            string url = "http://localhost:3000/register";
            driver.Navigate().GoToUrl(url);
            Console.WriteLine(driver.Title.ToString());
            string pagesource = driver.PageSource;
            if (pagesource != null)
            {
                driver.Manage().Window.Maximize();
                driver.FindElement(By.Id("UserEmail")).SendKeys("newsnht1@gmail.com");
            }
        }
    }
}

```

```

        driver.FindElement(By.Id("UserName")).SendKeys("newnsAhcp1u99nt");
        driver.FindElement(By.Id("UserPassword")).SendKeys("Acount231");
        driver.FindElement(By.Id("RegisterButton")).Click();
        WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));
        wait.Until(driver => driver.Url.Contains("/home"));
        string currenturl = driver.Url;
        if (currenturl == "http://localhost:3000/home")
        {
            Console.WriteLine("Test case passed for login with correct data redirect to next
page");
        }
        else
        {
            Console.WriteLine("Test case failed for login with correct not redirect to nextpage ");
            Console.WriteLine(currenturl);
        }
    }
}
public void Check_register_with_empty_data_return_fails()
{
    ExtentReporting.CreateTest("Check_register_with_empty_data_return_fails");

    string url = "http://localhost:3000/register";
    driver.Navigate().GoToUrl(url);
    Console.WriteLine(driver.Title.ToString());
    string pagesource = driver.PageSource;
    if (pagesource != null)
    {
        driver.Manage().Window.Maximize();
        driver.FindElement(By.Id("UserEmail")).SendKeys("");
        driver.FindElement(By.Id("UserName")).SendKeys("");
        driver.FindElement(By.Id("UserPassword")).SendKeys("");
        driver.FindElement(By.Id("RegisterButton")).Click();
        string useremailerror = driver.FindElement(By.Id("useremailerror")).Text;
        string userpwderror = driver.FindElement(By.Id("userpwderror")).Text;
        string usernameerror = driver.FindElement(By.Id("usernameerror")).Text;

        string currenturl = driver.Url;
        if (currenturl == "http://localhost:3000/" || useremailerror == "Please enter a email" ||
userpwderror == "Please enter a password" || usernameerror == "Please enter a username")
        {
            Console.WriteLine("Test case passed for regiter with no data not redirect to next
page");
        }
    }
}

```

```

        else
        {
            Console.WriteLine("Test case failed for register with correct data redirect to
nextpage");
            Console.WriteLine(currenturl);
        }
    }
}
public void Check_Register_Button_with_exists_data_input()
{
    ExtentReporting.CreateTest("Check_Register_Button_with_exists_data_input");

    string url = "http://localhost:3000/register";
    driver.Navigate().GoToUrl(url);
    Console.WriteLine(driver.Title.ToString());
    string pagesource = driver.PageSource;
    if (pagesource != null)
    {
        driver.Manage().Window.Maximize();
        driver.FindElement(By.Id("UserEmail")).SendKeys("Karthick@gmail.com");
        driver.FindElement(By.Id("UserName")).SendKeys("Karthick Subburaj");
        driver.FindElement(By.Id("UserPassword")).SendKeys("Karthick123");
        driver.FindElement(By.Id("RegisterButton")).Click();
        WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));
        wait.Until(driver => driver.Url.Contains("/register"));
        string currenturl = driver.Url;
        string servererror = driver.FindElement(By.Id("servererror")).Text;
        if (currenturl == "http://localhost:3000/register" || servererror == "Email is already
exists")
        {
            Console.WriteLine("Test case passed for register with exists data not redirect to
next page and show error");
        }
        else
        {
            Console.WriteLine("Test case failed for login with exists data redirect to nextpage
and not show error");
            Console.WriteLine(currenturl);
        }
    }
}
public void Check_work_for_Login_button()
{
    ExtentReporting.CreateTest("Check_work_for_Login_button");

```

```

string url = "http://localhost:3000/register";
driver.Navigate().GoToUrl(url);
Console.WriteLine(driver.Title.ToString());
string pagesource = driver.PageSource;

if (pagesource != null)
{
    driver.Manage().Window.Maximize();
    driver.FindElement(By.Id("LoginButton")).Click();
    WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));
    wait.Until(driver => driver.Url.Contains("/"));
    string currenturl = driver.Url;
    if (currenturl == "http://localhost:3000/")
    {
        Console.WriteLine("Test case passed for login button in login page redirect to
login page");
    }
    else
    {
        Console.WriteLine("Test case failed for login button in login page not redirect to
login page");
    }
}
}
}
}
}
}
}
}

```

```

LoginPage.cs
using OpenQA.Selenium;
using OpenQA.Selenium.Support.UI;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using OpenQA.Selenium.Chrome;
using Utils.Reports;

```

```

namespace SeleniumAssessment
{
    internal class LoginPage
    {
        private readonly IWebDriver driver;

        public LoginPage(IWebDriver webDriver)
        {
            driver = webDriver;
        }

        public void CheckLoginButtonwithvalidinput()
        {
            ExtentReporting.CreateTest("CheckLoginButtonwithvalidinput");

            string url = "http://localhost:3000";
            driver.Navigate().GoToUrl(url);
            Console.WriteLine(driver.Title.ToString());
            string pagesource = driver.PageSource;
            if (pagesource != null)
            {
                driver.Manage().Window.Maximize();
                driver.FindElement(By.Id("UserEmail")).SendKeys("Karthick@gmail.com");
                driver.FindElement(By.Id("UserPassword")).SendKeys("Karthick123");
                driver.FindElement(By.Id("LoginButton")).Click();
                WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));
                wait.Until(driver => driver.Url.Contains("/home"));
                string currenturl= driver.Url;
                if (currenturl == "http://localhost:3000/home")
                {
                    Console.WriteLine("Test case passed for login with correct data redirect to next
page");
                }
                else
                {
                    Console.WriteLine("Test case failed for login with correct not redirect to nextpage");
                    Console.WriteLine(currenturl);
                }
            }
        }

        public void Check_login_with_empty_data_return_fails()
        {
            ExtentReporting.CreateTest("Check_login_with_empty_data_return_fails");
        }
    }
}

```

```

string url = "http://localhost:3000";
driver.Navigate().GoToUrl(url);
Console.WriteLine(driver.Title.ToString());
string pagesource = driver.PageSource;
if (pagesource != null)
{
    driver.Manage().Window.Maximize();
    driver.FindElement(By.Id("UserEmail")).SendKeys("");
    driver.FindElement(By.Id("UserPassword")).SendKeys("");
    driver.FindElement(By.Id("LoginButton")).Click();
    string useremailerror=driver.FindElement(By.Id("useremailerror")).Text;
    string userpwderror = driver.FindElement(By.Id("userpwderror")).Text;

    string currenturl = driver.Url;
    if (currenturl == "http://localhost:3000/" && useremailerror== "Please enter a email"
&& userpwderror== "Please enter a password")
    {
        Console.WriteLine("Test case passed for login with no data not redirect to next
page");
    }
    else
    {
        Console.WriteLine("Test case failed for login with correct data redirect to
nextpage");
        Console.WriteLine(currenturl);
    }
}

}
//public void Check_Login_Button_with_invalid_account_input()
//{
//    string url = "http://localhost:3000";
//    driver.Navigate().GoToUrl(url);
//    Console.WriteLine(driver.Title.ToString());
//    string pagesource = driver.PageSource;

//    if (pagesource != null)
//    {
//        driver.Manage().Window.Maximize();
//        driver.FindElement(By.Id("UserEmail")).SendKeys("noaccount@gmail.com");
//        driver.FindElement(By.Id("UserPassword")).SendKeys("12345678");
//        driver.FindElement(By.Id("LoginButton")).Click();
//        string errormsg=driver.FindElement(By.Id("servererror")).Text;

```

```

//    string currenturl = driver.Url;
//    if (currenturl == "http://localhost:3000" && errormsg == "No account in this email")
//    {
//        Console.WriteLine("Test case passed for login with wrong data not redirect and
show error");
//    }
//    else
//    {
//        Console.WriteLine("Test case failed for login with wrong data not show error");

//    }

// }

//}
public void Check_Login_Button_with_invalid_password_input()
{
    ExtentReporting.CreateTest("Check_Login_Button_with_invalid_password_input");

    string url = "http://localhost:3000";
    driver.Navigate().GoToUrl(url);
    Console.WriteLine(driver.Title.ToString());
    string pagesource = driver.PageSource;

    if (pagesource != null)
    {
        driver.Manage().Window.Maximize();
        driver.FindElement(By.Id("UserEmail")).SendKeys("Karthick@gmail.com");
        driver.FindElement(By.Id("UserPassword")).SendKeys("12345678");
        driver.FindElement(By.Id("LoginButton")).Click();
        string errormsg = driver.FindElement(By.Id("servererror")).Text;

        string currenturl = driver.Url;
        if (currenturl == "http://localhost:3000" && errormsg == "Wrong Password")
        {
            Console.WriteLine("Test case passed for login with wrong password not redirect
and show error");
        }
        else
        {
            Console.WriteLine("Test case failed for login with wrong password not show error");

```



```
using OpenQA.Selenium.Support.UI;
using System;
```

```
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Utils.Reports;
```

```
namespace SeleniumAssessment
```

```
{
    internal class ListPage
    {
        private IWebDriver driver;
        public ListPage(IWebDriver driver) {
            this.driver = driver;
        }
        public void Check_work_for_Add_button()
        {
            ExtentReporting.CreateTest(" Check_work_for_Add_button");

            string url = "http://localhost:3000/list";
            driver.Navigate().GoToUrl(url);
            Console.WriteLine(driver.Title.ToString());
            string pagesource = driver.PageSource;

            if (pagesource != null)
            {
                driver.Manage().Window.Maximize();
                driver.FindElement(By.Id("AddButton")).Click();
                WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));
                wait.Until(driver => driver.Url.Contains("/create"));
                string currenturl = driver.Url;
                if (currenturl == "http://localhost:3000/create")
                {
                    Console.WriteLine("Test case passed for add button in list page redirect to create
page");
                }
                else
                {
                    Console.WriteLine("Test case failed for add button in list page not redirect to
create page");
                }
            }
        }
    }
}
```

```

    }

}

}
public void Check_work_for_Update_button()
{
    ExtentReporting.CreateTest("Check_work_for_Update_butto");

    string url = "http://localhost:3000/list";
    driver.Navigate().GoToUrl(url);
    Console.WriteLine(driver.Title.ToString());
    string pagesource = driver.PageSource;

    if (pagesource != null)
    {
        driver.Manage().Window.Maximize();
        driver.FindElements(By.Id("UpdateButton")).First().Click();
        WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));
        wait.Until(driver => driver.Url.Contains("/create/25"));
        string currenturl = driver.Url;
        if (currenturl == "http://localhost:3000/create/25")
        {
            Console.WriteLine("Test case passed for update button in list page redirect to
update page");
        }
        else
        {
            Console.WriteLine("Test case failed for update button in list page not redirect to
update page");
        }
    }

}

}
public void Check_work_for_Delete_button()
{
    ExtentReporting.CreateTest("Check_work_for_Delete_button");

    string url = "http://localhost:3000/list";
    driver.Navigate().GoToUrl(url);
    Console.WriteLine(driver.Title.ToString());

```

```

string pagesource = driver.PageSource;

if (pagesource != null)
{
    driver.Manage().Window.Maximize();
    driver.FindElements(By.Id("DeleteButton")).First().Click();
    WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));
    wait.Until(driver => driver.FindElement(By.Id("dialogbox")).Displayed);
    IWebElement modal = driver.FindElement(By.Id("dialogbox"));
    if (modal.Displayed)
    {
        Console.WriteLine("Test passed for delete button display the delete dialog box");
    }
    else
    {
        Console.WriteLine("Test failed for delete button not display the delete dialog box");
    }
}
}
}
}

```

```

HomePage.cs
using OpenQA.Selenium;

using OpenQA.Selenium.Support.UI;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Utils.Reports;

namespace SeleniumAssessment
{
    internal class HomePage
    {
        private readonly IWebDriver driver;
    }
}

```

```

public HomePage(IWebDriver webDriver)
{
    driver = webDriver;
}
public void Check_add_movie_button_redirect_to_add_page()
{
    ExtentReporting.CreateTest(" Check_add_movie_button_redirect_to_add_page");

    string url = "http://localhost:3000/home";
    driver.Navigate().GoToUrl(url);
    Console.WriteLine(driver.Title.ToString());
    string pagesource = driver.PageSource;

    if (pagesource != null)
    {
        driver.Manage().Window.Maximize();
        driver.FindElement(By.Id("AddMovie")).Click();
        WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));
        wait.Until(driver => driver.Url.Contains("/create"));
        string currenturl = driver.Url;
        if (currenturl == "http://localhost:3000/create")
        {
            Console.WriteLine("Test case passed for add movie button in home page redirect
to add movie page");
        }
        else
        {
            Console.WriteLine("Test case failed for add movie button in home page not
redirect to add movie page");
        }
    }
}
public void Check_list_movie_button_redirect_to_list_page()
{
    ExtentReporting.CreateTest(" Check_list_movie_button_redirect_to_list_page");

    string url = "http://localhost:3000/home";
    driver.Navigate().GoToUrl(url);
    Console.WriteLine(driver.Title.ToString());
    string pagesource = driver.PageSource;

```

```

        if (pagesource != null)
        {
            driver.Manage().Window.Maximize();
            driver.FindElement(By.Id("ListMovie")).Click();
            WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));
            wait.Until(driver => driver.Url.Contains("/list"));
            string currenturl = driver.Url;
            if (currenturl == "http://localhost:3000/list")
            {
                Console.WriteLine("Test case passed for list movie button in home page redirect
to list movie page");
            }
            else
            {
                Console.WriteLine("Test case failed for list movie button in home page not redirect
to list movie page");
            }
        }
    }
}

public void Check_value_in_total_movie_not_null()
{
    ExtentReporting.CreateTest(" Check_value_in_total_movie_not_null");

    string url = "http://localhost:3000/home";
    driver.Navigate().GoToUrl(url);
    Console.WriteLine(driver.Title.ToString());
    string pagesource = driver.PageSource;

    if (pagesource != null)
    {
        driver.Manage().Window.Maximize();
        int total = Convert.ToInt32(driver.FindElement(By.Id("TotalMovie")).Text);
        WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));
        wait.Until(driver => driver.Url.Contains("/home"));
        string currenturl = driver.Url;
        if (currenturl == "http://localhost:3000/home" && total>0)
        {

```

```

        Console.WriteLine("Test case passed for total movie column show number of movie
in home page ");
    }
    else
    {
        Console.WriteLine("Test case failed for total movie column not show number of
movie in home page ");
    }
}
}
}
}
}

```

ListPage.cs

```

using OpenQA.Selenium;
using OpenQA.Selenium.DevTools.V119.WebAuthn;
using OpenQA.Selenium.Support.UI;
using System;

using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Utils.Reports;

namespace SeleniumAssessment
{
    internal class ListPage
    {
        private IWebDriver driver;
        public ListPage(IWebDriver driver) {
            this.driver = driver;
        }
        public void Check_work_for_Add_button()
        {
            ExtentReporting.CreateTest(" Check_work_for_Add_button");

            string url = "http://localhost:3000/list";
            driver.Navigate().GoToUrl(url);
            Console.WriteLine(driver.Title.ToString());
        }
    }
}

```

```

string pagesource = driver.PageSource;

if (pagesource != null)
{
    driver.Manage().Window.Maximize();
    driver.FindElement(By.Id("AddButton")).Click();
    WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));
    wait.Until(driver => driver.Url.Contains("/create"));
    string currenturl = driver.Url;
    if (currenturl == "http://localhost:3000/create")
    {
        Console.WriteLine("Test case passed for add button in list page redirect to create
page");
    }
    else
    {
        Console.WriteLine("Test case failed for add button in list page not redirect to
create page");
    }
}

}

public void Check_work_for_Update_button()
{
    ExtentReporting.CreateTest("Check_work_for_Update_butto");

    string url = "http://localhost:3000/list";
    driver.Navigate().GoToUrl(url);
    Console.WriteLine(driver.Title.ToString());
    string pagesource = driver.PageSource;

    if (pagesource != null)
    {
        driver.Manage().Window.Maximize();
        driver.FindElements(By.Id("UpdateButton")).First().Click();
        WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));
        wait.Until(driver => driver.Url.Contains("/create/25"));
        string currenturl = driver.Url;
        if (currenturl == "http://localhost:3000/create/25")
        {

```



```

        Console.WriteLine("Test case passed for update button in list page redirect to
update page");
    }
    else
    {
        Console.WriteLine("Test case failed for update button in list page not redirect to
update page");
    }
}
}
}
public void Check_work_for_Delete_button()
{
    ExtentReporting.CreateTest("Check_work_for_Delete_button");

    string url = "http://localhost:3000/list";
    driver.Navigate().GoToUrl(url);
    Console.WriteLine(driver.Title.ToString());
    string pagesource = driver.PageSource;

    if (pagesource != null)
    {
        driver.Manage().Window.Maximize();
        driver.FindElements(By.Id("DeleteButton")).First().Click();
        WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));
        wait.Until(driver => driver.FindElement(By.Id("dialogbox")).Displayed);
        IWebElement modal = driver.FindElement(By.Id("dialogbox"));
        if (modal.Displayed)
        {
            Console.WriteLine("Test passed for delete button display the delete dialog box");
        }
        else
        {
            Console.WriteLine("Test failed for delete button not display the delete dialog box");
        }
    }
}
}
}

```

```
}
```

CreatePage.cs

```
using OpenQA.Selenium.Support.UI;
using OpenQA.Selenium;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Newtonsoft.Json;
using Utils.Reports;
```

```
namespace SeleniumAssessment
```

```
{
    internal class CreatePage(IWebDriver webDriver)
    {
        private readonly IWebDriver driver = webDriver;

        public void CheckAddButtonwithvalidinput()
        {
            ExtentReporting.CreateTest("CheckAddButtonwithvalidinput");

            string url = "http://localhost:3000/create";
            driver.Navigate().GoToUrl(url);
            Console.WriteLine(driver.Title.ToString());
            string pagesource = driver.PageSource;
            if (pagesource != null)
            {
                driver.Manage().Window.Maximize();
                driver.FindElement(By.Id("movieName")).SendKeys("Leo 2");
                driver.FindElement(By.Id("movieType")).SendKeys("Action");
                driver.FindElement(By.Id("movieLanguage")).SendKeys("Tamil");
                driver.FindElement(By.Id("movieDurations")).SendKeys("2.54 hour");
                driver.FindElement(By.Id("AddUpdateButton")).Click();
                WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));
                wait.Until(driver => driver.Url.Contains("/list"));
                string currenturl = driver.Url;
                if (currenturl == "http://localhost:3000/list")
                {
                    Console.WriteLine("Test case passed for add movie with correct data redirect to
next page");
                }
            }
        }
    }
}
```

```

    }
    else
    {
        Console.WriteLine("Test case failed for add movie with correct not redirect to
nextpage ");
        Console.WriteLine(currenturl);
    }
}
}
public void Check_add_movie_with_empty_data_return_fails()
{
    ExtentReporting.CreateTest(" Check_add_movie_with_empty_data_return_fails");

    string url = "http://localhost:3000/create";
    driver.Navigate().GoToUrl(url);
    Console.WriteLine(driver.Title.ToString());
    string pagesource = driver.PageSource;
    if (pagesource != null)
    {
        driver.Manage().Window.Maximize();
        driver.FindElement(By.Id("movieName")).SendKeys("");
        driver.FindElement(By.Id("movieType")).SendKeys("");
        driver.FindElement(By.Id("movieLanguage")).SendKeys("");
        driver.FindElement(By.Id("movieDurations")).SendKeys("");
        driver.FindElement(By.Id("AddUpdateButton")).Click();
        string nameerror= driver.FindElement(By.Id("nameerror")).Text;
        string typeerror= driver.FindElement(By.Id("typeerror")).Text;
        string langerror= driver.FindElement(By.Id("langerror")).Text;
        string duraerror= driver.FindElement(By.Id("duraerror")).Text;

        WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));
        wait.Until(driver => driver.Url.Contains("/create"));
        string currenturl = driver.Url;
        if (nameerror== "***Required ! Must fill the field ***" && typeerror== "***Required ! Must
fill the field ***" && langerror== "***Required ! Must fill the field ***" && duraerror == "***Required !
Must fill the field ***")
        {
            Console.WriteLine("Test case passed for add movie with no data not redirect to
next page and show error");
        }
    }
}

```

```

        else
        {
            Console.WriteLine("Test case failed for add movie with no data or not show error");
            Console.WriteLine(currenturl);
        }
    }
}

public void Check_work_for_View_button()
{
    ExtentReporting.CreateTest("Check_work_for_View_button");

    string url = "http://localhost:3000/create";
    driver.Navigate().GoToUrl(url);
    Console.WriteLine(driver.Title.ToString());
    string pagesource = driver.PageSource;

    if (pagesource != null)
    {
        driver.Manage().Window.Maximize();
        driver.FindElement(By.Id("ViewButton")).Click();
        WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));
        wait.Until(driver => driver.Url.Contains("/list"));
        string currenturl = driver.Url;
        if (currenturl == "http://localhost:3000/list")
        {
            Console.WriteLine("Test case passed for view button in create page redirect to list
page");
        }
        else
        {
            Console.WriteLine("Test case failed for view button in create page not redirect to
list page");
        }
    }
}
}
}

```

```
}
```

UpdatePage.cs

```
using OpenQA.Selenium;
using OpenQA.Selenium.Support.UI;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Utils.Reports;
```

```
namespace SeleniumAssessment
```

```
{
```

```
    internal class UpdatePage
```

```
    {
```

```
        private readonly IWebDriver driver;
```

```
        public UpdatePage(IWebDriver webDriver)
```

```
        {
```

```
            driver = webDriver;
```

```
        }
```

```
        public void CheckUpdateButtonwithvalidinput()
```

```
        {
```

```
            ExtentReporting.CreateTest("CheckUpdateButtonwithvalidinput");
```

```
            string url = "http://localhost:3000/create/13";
```

```
            driver.Navigate().GoToUrl(url);
```

```
            Console.WriteLine(driver.Title.ToString());
```

```
            string pagesource = driver.PageSource;
```

```
            if (pagesource != null)
```

```
            {
```

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

```
                driver.FindElement(By.Id("movieName")).Clear();
```

```
                driver.FindElement(By.Id("movieType")).Clear();
```

```
                driver.FindElement(By.Id("movieLanguage")).Clear();
```

```
                driver.FindElement(By.Id("movieDurations")).Clear();
```

```
                driver.FindElement(By.Id("movieName")).SendKeys("Leo 2");
```

```
                driver.FindElement(By.Id("movieType")).SendKeys("Action");
```

```
                driver.FindElement(By.Id("movieLanguage")).SendKeys("Tamil");
```

```
                driver.FindElement(By.Id("movieDurations")).SendKeys("2.54 hour");
```

```
                driver.FindElement(By.Id("AddUpdateButton")).Click();
```

```

        WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));
        wait.Until(driver => driver.Url.Contains("/"));
        string currenturl = driver.Url;
        if (currenturl == "http://localhost:3000/list")
        {
            Console.WriteLine("Test case passed for update movie with correct data redirect to
next page");
        }
        else
        {
            Console.WriteLine("Test case failed for update movie with correct not redirect to
nextpage ");
            Console.WriteLine(currenturl);
        }
    }
}

public void Check_update_movie_with_empty_data_return_fails()
{
    ExtentReporting.CreateTest("Check_update_movie_with_empty_data_return_fails");

    string url = "http://localhost:3000/create/13";
    driver.Navigate().GoToUrl(url);
    Console.WriteLine(driver.Title.ToString());
    string pagesource = driver.PageSource;
    if (pagesource != null)
    {
        driver.Manage().Window.Maximize();
        driver.FindElement(By.Id("movieName")).Clear();
        driver.FindElement(By.Id("movieType")).Clear();
        driver.FindElement(By.Id("movieLanguage")).Clear();
        driver.FindElement(By.Id("movieDurations")).Clear();
        driver.FindElement(By.Id("movieName")).SendKeys("");
        driver.FindElement(By.Id("movieType")).SendKeys("");
        driver.FindElement(By.Id("movieLanguage")).SendKeys("");
        driver.FindElement(By.Id("movieDurations")).SendKeys("");
        driver.FindElement(By.Id("AddUpdateButton")).Click();
        WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(100));
        wait.Until(driver => driver.Url.Contains("/create"));
    }
}

```

```

        string currenturl = driver.Url;
        if (currenturl == "http://localhost:3000/create/24")
        {
            Console.WriteLine("Test case passed for update movie with no data not redirect to
next page and show error");

        }
        else
        {
            Console.WriteLine("Test case failed for update movie with no data or not show
error");
            Console.WriteLine(currenturl);
        }
    }
}

public void Check_work_for_View_button()
{
    ExtentReporting.CreateTest("Check_work_for_View_button");

    string url = "http://localhost:3000/create";
    driver.Navigate().GoToUrl(url);
    Console.WriteLine(driver.Title.ToString());
    string pagesource = driver.PageSource;

    if (pagesource != null)
    {
        driver.Manage().Window.Maximize();
        driver.FindElement(By.Id("ViewButton")).Click();
        WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));
        wait.Until(driver => driver.Url.Contains("/list"));
        string currenturl = driver.Url;
        if (currenturl == "http://localhost:3000/list")
        {
            Console.WriteLine("Test case passed for view button in create page redirect to list
page");

        }
        else
        {
            Console.WriteLine("Test case failed for view button in create page not redirect to
list page");

```

```

    }
}
}
}
}

```

DeletePage.cs

```

using OpenQA.Selenium;
using OpenQA.Selenium.Support.UI;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Utils.Reports;

```

namespace SeleniumAssessment

```

{
    internal class DeletePage
    {
        private IWebDriver driver;
        public DeletePage(IWebDriver driver)
        {
            this.driver = driver;
        }
        public void Check_delete_button_display_Dialog_box()
        {
            ExtentReporting.CreateTest(" Check_delete_button_display_Dialog_box");

            string url = "http://localhost:3000/list";
            driver.Navigate().GoToUrl(url);
            Console.WriteLine(driver.Title.ToString());
            string pagesource = driver.PageSource;

            if (pagesource != null)
            {
                driver.Manage().Window.Maximize();
                driver.FindElements(By.Id("DeleteButton")).First().Click();
            }
        }
    }
}

```



```

        WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));
        wait.Until(driver => driver.FindElement(By.Id("dialogbox")).Displayed);
        IWebElement modal = driver.FindElement(By.Id("dialogbox"));
        if (modal.Displayed)
        {
            Console.WriteLine("Test passed for delete button display the delete dialog box");
        }
        else
        {
            Console.WriteLine("Test failed for delete button not display the delete dialog box");
        }
    }
}
public void Check_dialog_box_Delete_button_delete_element()
{
    ExtentReporting.CreateTest(" Check_dialog_box_Delete_button_delete_element(");

    string url = "http://localhost:3000/list";
    driver.Navigate().GoToUrl(url);
    Console.WriteLine(driver.Title.ToString());
    string pagesource = driver.PageSource;

    if (pagesource != null)
    {
        driver.Manage().Window.Maximize();
        driver.FindElements(By.Id("DeleteButton")).First().Click();
        WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));
        wait.Until(driver => driver.FindElement(By.Id("dialogbox")).Displayed);
        IWebElement modal = driver.FindElement(By.Id("dialogbox"));
        if (modal.Displayed)
        {
            driver.FindElement(By.Id("dialogdeletebutton")).Click();
            wait.Until(driver => driver.Url.Contains("/list"));
            if(modal==null)
            {
                Console.WriteLine("Test failed for delete button in dialog box not return to list
page");
            }
        }
    }
}

```

```

        else
        {
            Console.WriteLine("Test passed for delete button in dialog box return to list
page");

        }
    }
}

public void Check_dialog_box_Cancel_button_return_list_page()
{
    ExtentReporting.CreateTest(" Check_dialog_box_Cancel_button_return_list_page");

    string url = "http://localhost:3000/list";
    driver.Navigate().GoToUrl(url);
    Console.WriteLine(driver.Title.ToString());
    string pagesource = driver.PageSource;

    if (pagesource != null)
    {
        driver.Manage().Window.Maximize();
        driver.FindElements(By.Id("DeleteButton")).First().Click();
        WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));
        wait.Until(driver => driver.Url.Contains("/list"));

        IWebElement modal = driver.FindElement(By.Id("dialogbox"));

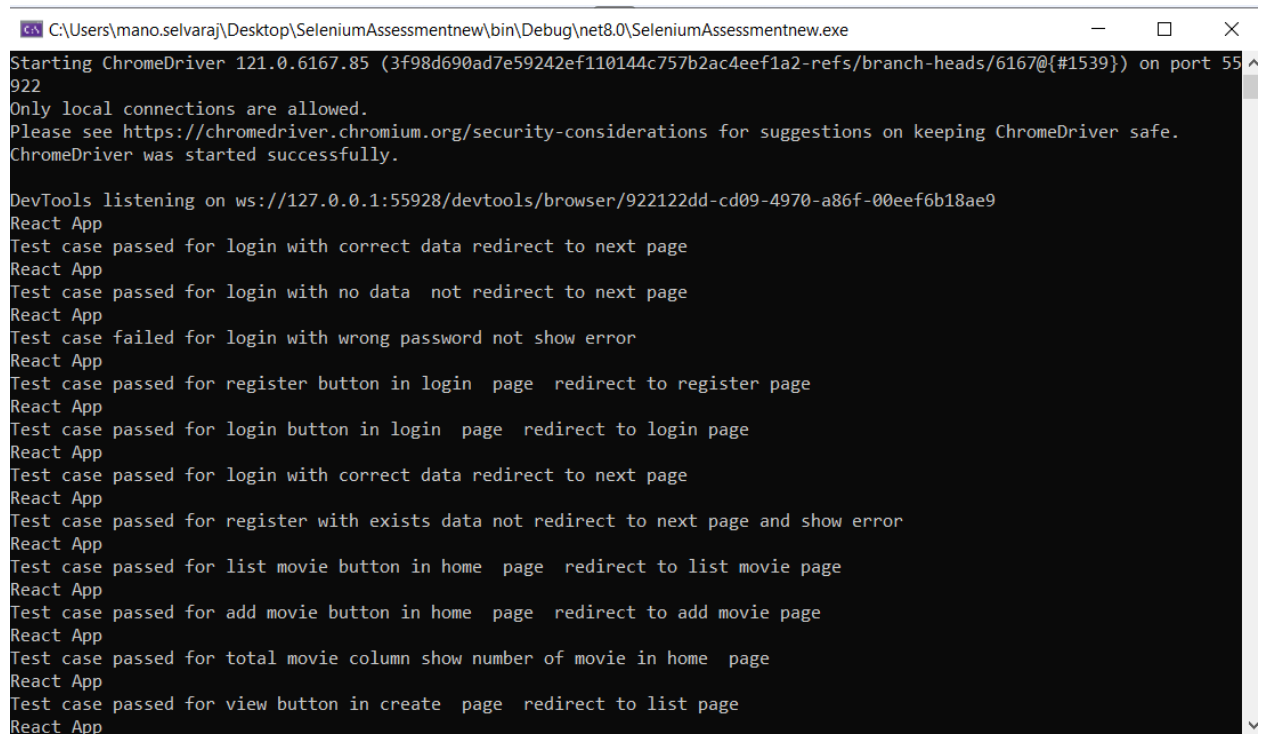
        if (modal.Displayed)
        {
            driver.FindElement(By.Id("dialogcancelbutton")).Click();
            wait.Until(driver => driver.Url.Contains("/list"));
            if (modal==null)
            {
                Console.WriteLine("Test failed for delete button in dialog box not return to list
page");
            }
        }
        else
        {

```

```
Console.WriteLine("Test passed for delete button in dialog box return to list  
page");
```

```
    }  
  }  
}  
}
```

Console Result



A screenshot of a Windows command prompt window. The title bar shows the file path: C:\Users\mano.selvaraj\Desktop\SeleniumAssessmentnew\bin\Debug\net8.0\SeleniumAssessmentnew.exe. The window contains the following text:

```
Starting ChromeDriver 121.0.6167.85 (3f98d690ad7e59242ef110144c757b2ac4eef1a2-refs/branch-heads/6167@{#1539}) on port 55922
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.

DevTools listening on ws://127.0.0.1:55928/devtools/browser/922122dd-cd09-4970-a86f-00eef6b18ae9
React App
Test case passed for login with correct data redirect to next page
React App
Test case passed for login with no data not redirect to next page
React App
Test case failed for login with wrong password not show error
React App
Test case passed for register button in login page redirect to register page
React App
Test case passed for login button in login page redirect to login page
React App
Test case passed for login with correct data redirect to next page
React App
Test case passed for register with exists data not redirect to next page and show error
React App
Test case passed for list movie button in home page redirect to list movie page
React App
Test case passed for add movie button in home page redirect to add movie page
React App
Test case passed for total movie column show number of movie in home page
React App
Test case passed for view button in create page redirect to list page
React App
```

```
C:\Users\mano.selvaraj\Desktop\SeleniumAssessmentnew\bin\Debug\net8.0\SeleniumAssessmentnew.exe
Test case passed for view button in create page redirect to list page
React App
Test case passed for add movie with correct data redirect to next page
React App
Test case passed for add movie with no data not redirect to next page and show error
React App
Test passed for delete button display the delete dialog box
React App
Test case passed for update button in list page redirect to update page
React App
Test case passed for add button in list page redirect to create page
React App
Test case failed for update movie with correct not redirect to nextpage
http://localhost:3000/create/13
React App
Test case failed for update movie with no data or not show error
http://localhost:3000/create/13
React App
Test case passed for view button in create page redirect to list page
React App
Test passed for delete button display the delete dialog box
React App
Test passed for delete button in dialog box return to list page
React App
Test passed for delete button in dialog box return to list page
```

Extent Report Screenshot

| Tests | | | CheckLoginButtonwithvalidinput | |
|-------|--|--------------------|--------------------------------|-------------------------------------|
| | CheckLoginButtonwithvalidinput | 17:17:10 / 0 ms | 31-01-2024 17:17:10 | 31-01-2024 17:17:10 0 ms #test-id=1 |
| | Check_login_with_empty_data_return_fails | 17:17:12 / 0 ms | | |
| | Check_Login_Button_with_invalid_password_input | 17:17:13 / 0 ms | | |
| | Check_work_for_Register_button | 17:17:14 / 0 ms | | |
| | Check_work_for_Login_button | 17:17:14 / 0 ms | | |
| | CheckRegisterButtonwithvalidinput | 17:17:14 / 0 ms | | |
| | Check_Register_Button_with_exists_data_input | 17:17:15 / 0 ms | | |
| | Check_list_movie_button_redirect_to_list_page | 17:17:16 / 0 ms | | |
| | Check_add_movie_button_redirect_to_add_page | 17:17:16 / 0 ms | | |
| | Check_value_in_total_movie_not_null | 17:17:17 / 0.01 ms | | |

31-01-2024 17:17:10

Tests

Check_work_for_Delete_button
17:17:19 / 0 ms

Check_work_for_Update_butto
17:17:20 / 0 ms

Check_work_for_Add_button
17:17:20 / 0 ms

CheckUpdateButtonwithvalidinput
17:17:20 / 0 ms

Check_update_movie_with_empty_data_return_fails
17:17:21 / 0 ms

Check_work_for_View_button
17:17:22 / 0 ms

Check_delete_button_display_Dialog_box
17:17:22 / 0 ms

Check_dialog_box_Delete_button_delete_element(
17:17:22 / 0 ms

Check_dialog_box_Cancel_button_return_list_page
17:17:23 / 0 ms

CheckLoginButtonwithvalidinput

31-01-2024 17:17:1031-01-2024 17:17:300 ms#test-id=1

31-01-2024 17:17:10

Started
31-01-2024 17:17:10

Ended
31-01-2024 17:17:24

Tests Passed
22

Tests Failed
0

Tests

Pass

22 tests passed
0 tests failed, 0 skipped, 0 others

Timeline

00.0050.0100.0150.0200.0250.0300.035