**FINAL QA PROJECT**

**Submitted by : Aaditi Singh**

**(Fynd QA Batch-6)**

**Manual Testing :**

**User Story 0001:** Add “About us” link option in footer of the application

Scenario 1: Verify the “About us “link is available and displayed

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SLNo | Test Case | Expected Result | Test Data | Test Status |
| 1 | Launch the application | The application is launched successfully. | [**https://test-nf.com/english.html**](https://test-nf.com/english.html) | Pass |
|  |  |  |  |  |

Scenario 2: Verify the position of the “About us” link.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SL.No | Test Case | Expected Result | Test Data | Test Status |
| 1 | Launch the application | The application is launched successfully. | [**https://test-nf.com/english.html**](https://test-nf.com/english.html) | Pass |
| 2 | Position of the link | The link is always available in footer of the application. |  |  |

Scenario 3: Verify the “About us” link is highlighted

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl No | Test Case | Expected Result | Test Data | Test Status |
| 1 | Launch the application | The application is launched successfully. | [**https://test-nf.com/english.html**](https://test-nf.com/english.html) | Pass |
| 2 | Position of the link | The link is always available in footer of the application. |  |  |
| 3 | Check the colour of the text | The link is highlighted and underlined |  |  |

Scenario 4: Verify the “About us” link is clickable

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl No | Test Case | Expected Result | Test Data | Test Status |
| 1 | Launch the application | The application is launched successfully. | [**https://test-nf.com/english.html**](https://test-nf.com/english.html) | Pass |
| 2 | Position of the link | The link is always available in footer of the application. |  |  |
| 3 | Click on the link | The user should be able to click |  |  |

Scenario 5: Verify the “About us” link is navigating to a new page

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl No | Test Case | Expected Result | Test Data | Test Status |
| 1 | Launch the application | The application is launched successfully. | [**https://test-nf.com/english.html**](https://test-nf.com/english.html) | Pass |
| 2 | Position of the link | The link is always available in footer of the application. |  |  |
| 3 | Click on the link | User should be able to click the link |  |  |
| 4 | Check that link is navigating to new page | Link is navigating to a new page |  |  |

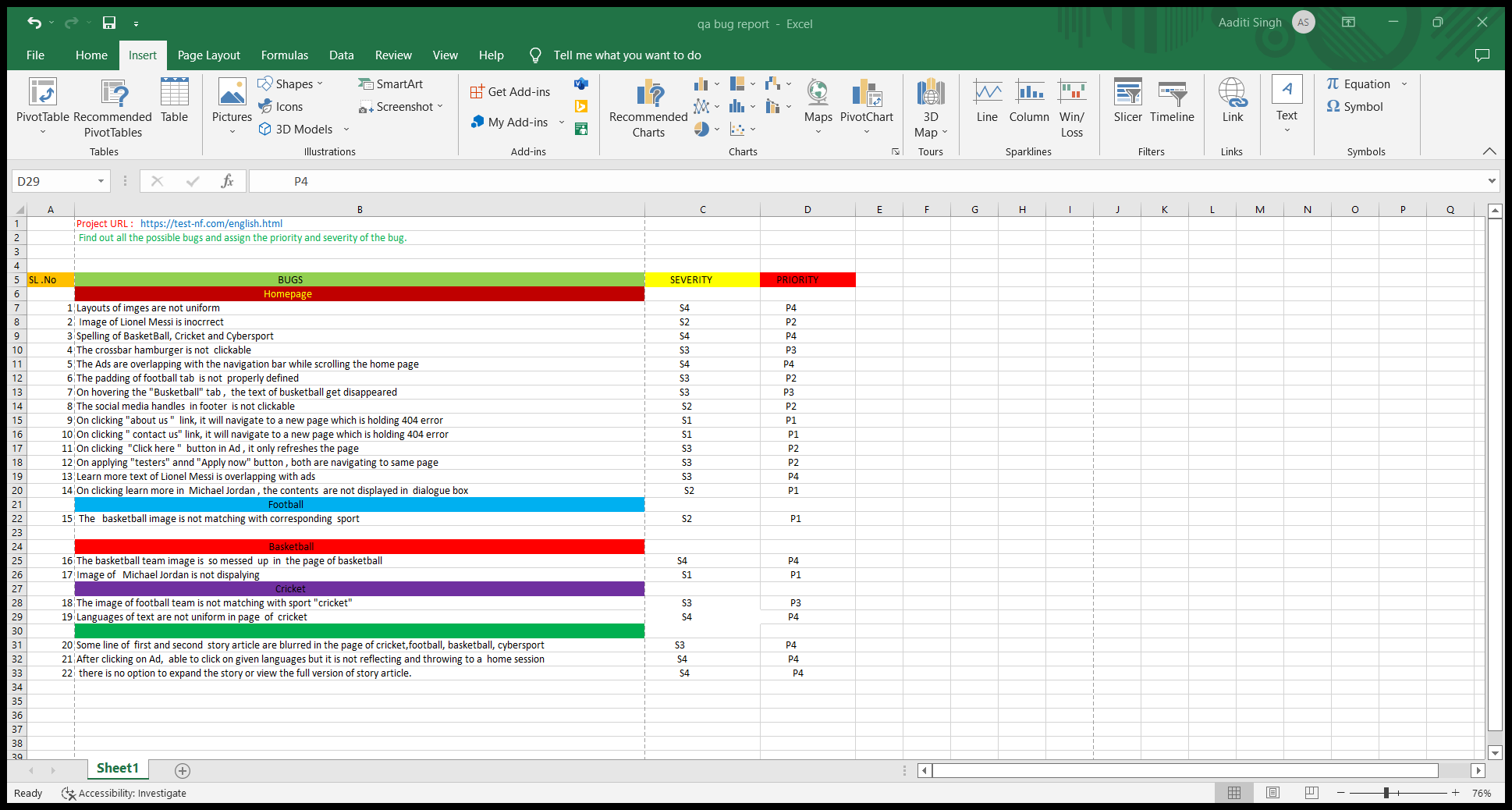
Scenario 6: Verify that on clicking the “About us” link, the link is displaying the Organization vision and Mission

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl No | Test Case | Expected Result | Test Data | Test Status |
| 1 | Launch the application | The application is launched successfully. | [**https://test-nf.com/english.html**](https://test-nf.com/english.html) | Pass |
| 2 | Position of the link | The link is always available in footer of the application. |  |  |
| 3 | Click on the link | User should be able to click the link |  |  |
| 4 | Check the link is directing to new page | Link is navigating to a new page |  |  |
| 5 | Check the contents and data int the page | Link is displaying the Organization vision and Mission |  |  |

Scenario 7: Verify “About us “link should be available for all tabs.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl No | Test Case | Expected Result | Test Data | Test Status |
| 1 | Launch the application | The application is launched successfully. | [**https://test-nf.com/english.html**](https://test-nf.com/english.html) | Pass |
| 2 | Click on all tabs | All tabs are navigating to new page. |  |  |
| 3 | Check the availability of link in footer | The link is available in footer for all tabs. |  |  |

Bug Report:



**Automation Testing Scenario:**

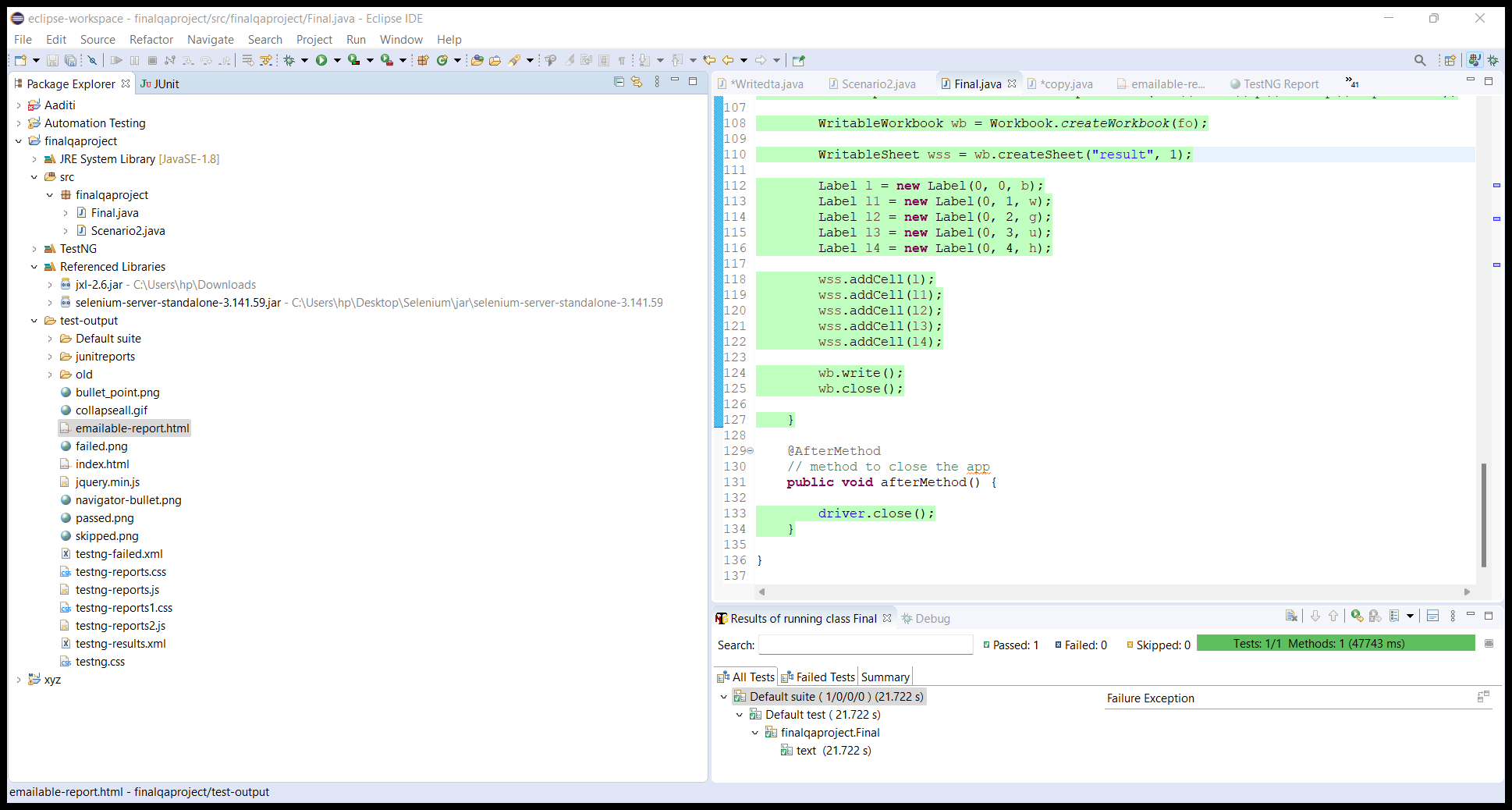
1. Launch the application: <https://test-nf.com/english.html>

2. Verify the availability of each tab (Home, Cricket, Football, Basketball and Ciber Sports)

3. Verify URL of each tab contains the tab name. 4. Store all the current URL in excel sheet. Note: Implement Page Object model, Assertion for Validation, Excel Sheet Integration for passing System properties and application URL.

Answer –

Folder Structure



Automation script to verify the given scenarios

This Scenario2.java class to store the store the location of the tab.

package finalqaproject;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindBy;

//create page object class for storing the locators of the tabs

public class Scenario2 {

@FindBy(xpath="/html/body/header/div/ul/li[1]/a")WebElement Home;

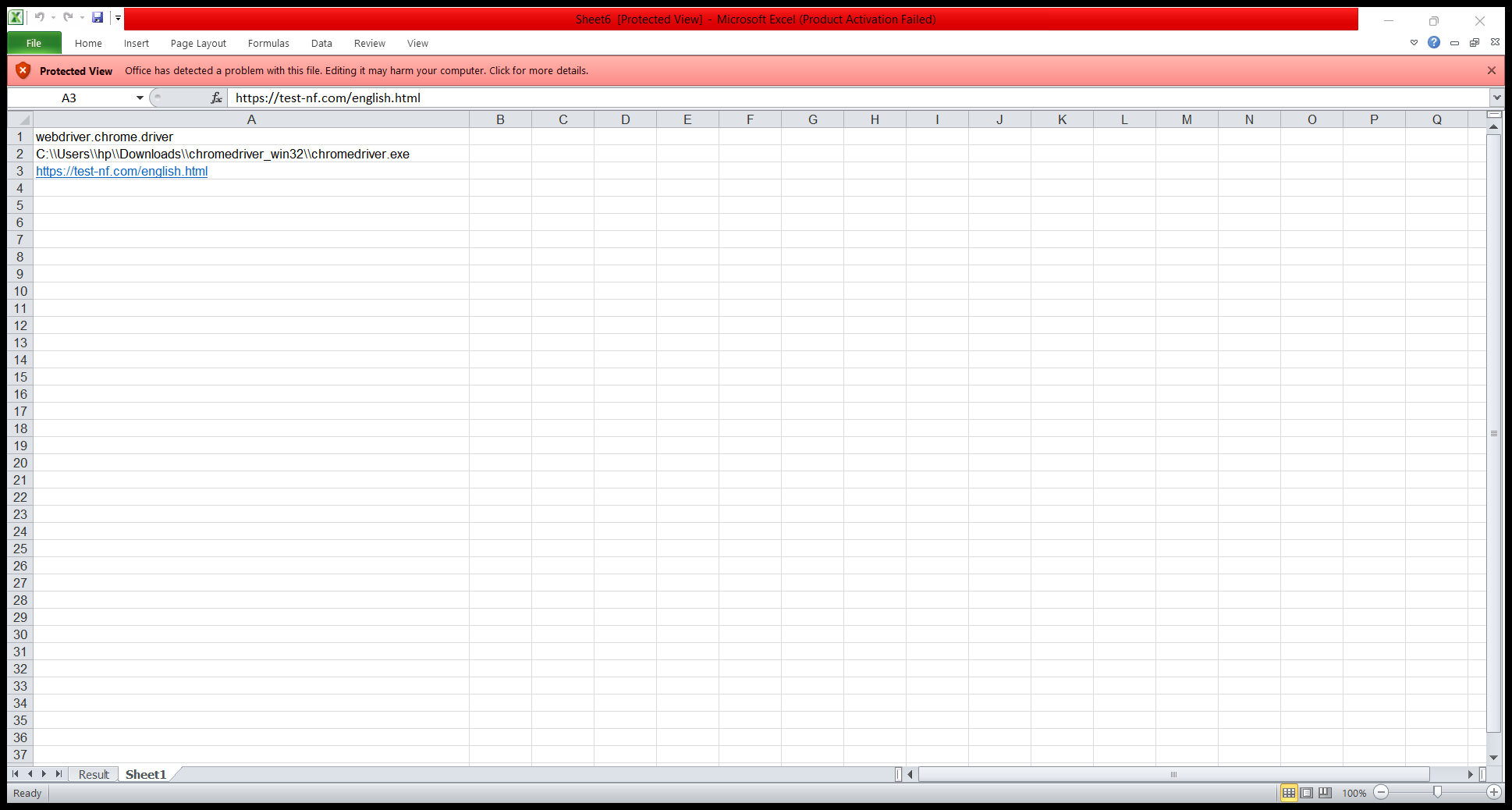
@FindBy(xpath="/html/body/header/div/ul/li[2]/a")WebElement Football;

@FindBy(xpath="/html/body/header/div/ul/li[3]/a")WebElement BasketBall;

@FindBy(xpath="/html/body/header/div/ul/li[4]/a")WebElement Cricket;

@FindBy(xpath="/html/body/header/div/ul/li[5]/a")WebElement Cibersport;

}

Source path of Browser Driver and Link URL of the website

This is the final.java class where we use Excel sheet Integration, TestNg, Assertion validation and Pageobject model.

**package** finalqaproject;

**import** java.io.FileInputStream;

**import** java.io.FileOutputStream;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.openqa.selenium.support.PageFactory;

**import** org.testng.annotations.AfterMethod;

**import** org.testng.annotations.BeforeMethod;

**import** org.testng.annotations.Test;

**import** org.testng.asserts.SoftAssert;

**import** jxl.Sheet;

**import** jxl.Workbook;

**import** jxl.write.Label;

**import** jxl.write.WritableSheet;

**import** jxl.write.WritableWorkbook;

**public** **class** Final {

**public** WebDriver driver;

@BeforeMethod

**public** **void** beforeMethod()**throws** Exception {

// Here we use excel integration to pass the data from excel

FileInputStream f = **new** FileInputStream("C:\\Users\\hp\\Desktop\\Sheet6.xls");

Workbook wb = Workbook.*getWorkbook*(f);

Sheet s = wb.getSheet("Sheet1");

System.*setProperty*(s.getCell(0, 0).getContents(), s.getCell(0, 1).getContents());

driver = **new** ChromeDriver();

driver.get(s.getCell(0, 2).getContents());

// to launch the app

driver.manage().window().maximize();

}

@Test

**public** **void** text()**throws** Exception {

// import the page factory method by page factory property

Scenario2 s = PageFactory.*initElements*(driver, Scenario2.**class**);

// use of soft assertion to validate the given scenarios

SoftAssert t = **new** SoftAssert();

// 2. Verify the availability of each tab(Home, Cricket, Football, Basketball

// and Ciber Sports)

**boolean** check = s.Home.isDisplayed();

t.assertEquals(check, **true**);

**boolean** check1 = s.Football.isDisplayed();

t.assertEquals(check1, **true**);

**boolean** check2 = s.Cricket.isDisplayed();

t.assertEquals(check2, **true**);

**boolean** check3 = s.Cibersport.isDisplayed();

t.assertEquals(check3, **true**);

// Scenario 3. Verify URL of each tab contains the tab name.

// checking for home tab

s.Home.click();

Thread.*sleep*(2000);

String b = driver.getCurrentUrl();

**boolean** v = b.contains("english");

t.assertEquals(v, **true**);

System.***out***.println("contain same name");

// Checking for Football tab

s.Football.click();

Thread.*sleep*(2000);

String w = driver.getCurrentUrl();

**boolean** a = w.contains("football");

t.assertEquals(a, **true**);

System.***out***.println("contain same name");

// Checking for busketball tab

s.BasketBall.click();

Thread.*sleep*(2000);

String g = driver.getCurrentUrl();

**boolean** d = g.contains("busketball");

t.assertEquals(d, **true**);

System.***out***.println("contain same name");

// checking for cricket tab

s.Cricket.click();

Thread.*sleep*(2000);

String u = driver.getCurrentUrl();

**boolean** a2 = u.contains("kriket");

t.assertEquals(a2, **true**);

System.***out***.println("contain same name");

// checking for cybersport tab

s.Cibersport.click();

Thread.*sleep*(2000);

String h = driver.getCurrentUrl();

**boolean** a3 = h.contains("cibersport");

t.assertEquals(a3, **true**);

System.***out***.println("contain same name");

// 4.Store all the current URL of tabs in excel sheet

FileOutputStream fo = **new** FileOutputStream("C:\\Users\\hp\\Desktop\\output2.xls");

//creating workbook

WritableWorkbook wb = Workbook.*createWorkbook*(fo);

//creating sheet in workbook

WritableSheet wss = wb.createSheet("result", 1);

//creating cell using Label class

Label l = **new** Label(0, 0, b);

Label l1 = **new** Label(0, 1, w);

Label l2 = **new** Label(0, 2, g);

Label l3 = **new** Label(0, 3, u);

Label l4 = **new** Label(0, 4, h);

wss.addCell(l);

wss.addCell(l1);

wss.addCell(l2);

wss.addCell(l3);

wss.addCell(l4);

//saving and closing workbook

wb.write();

wb.close();

}

@AfterMethod

// method to close the app

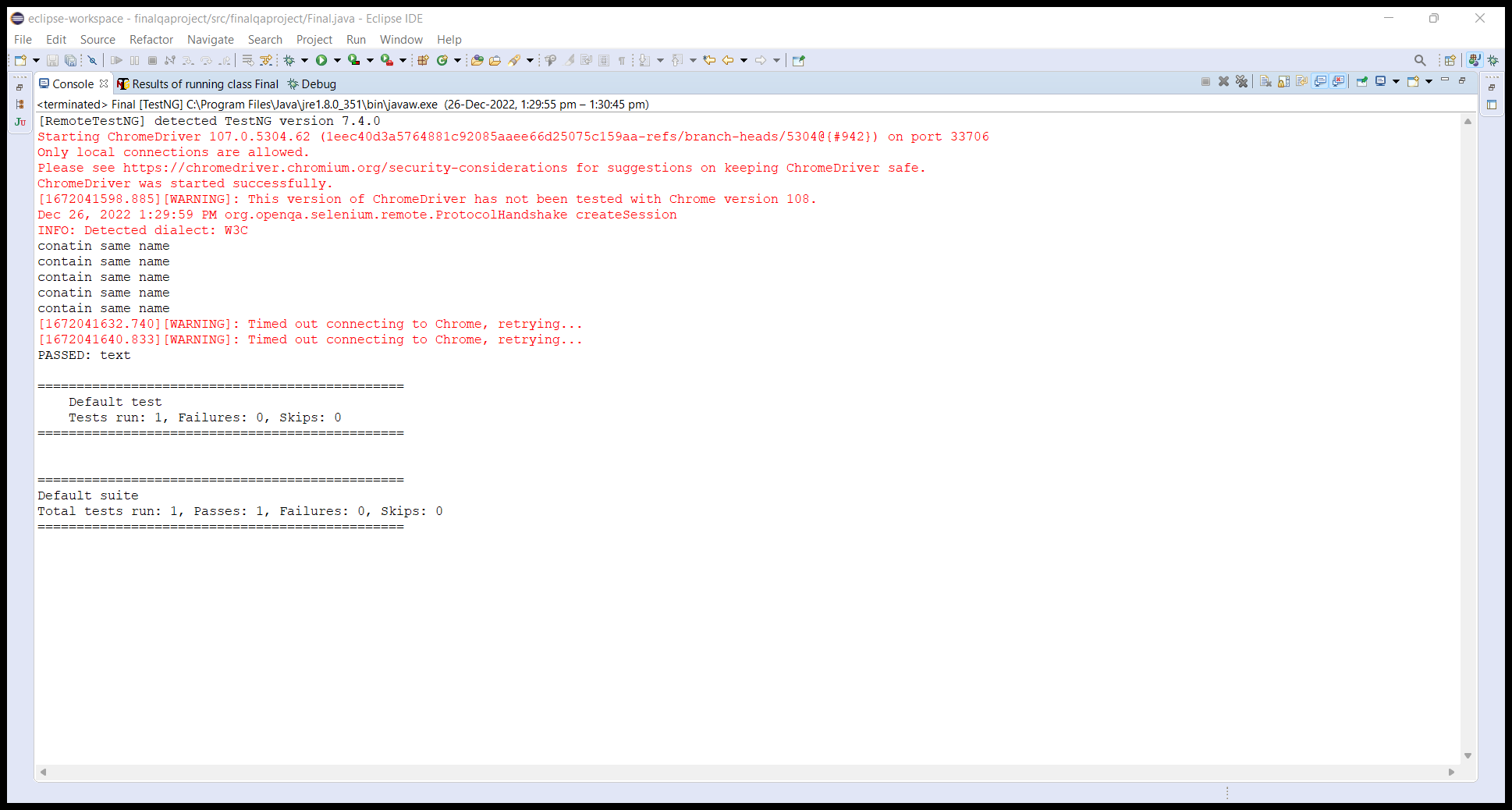
**public** **void** afterMethod() {

driver.close();

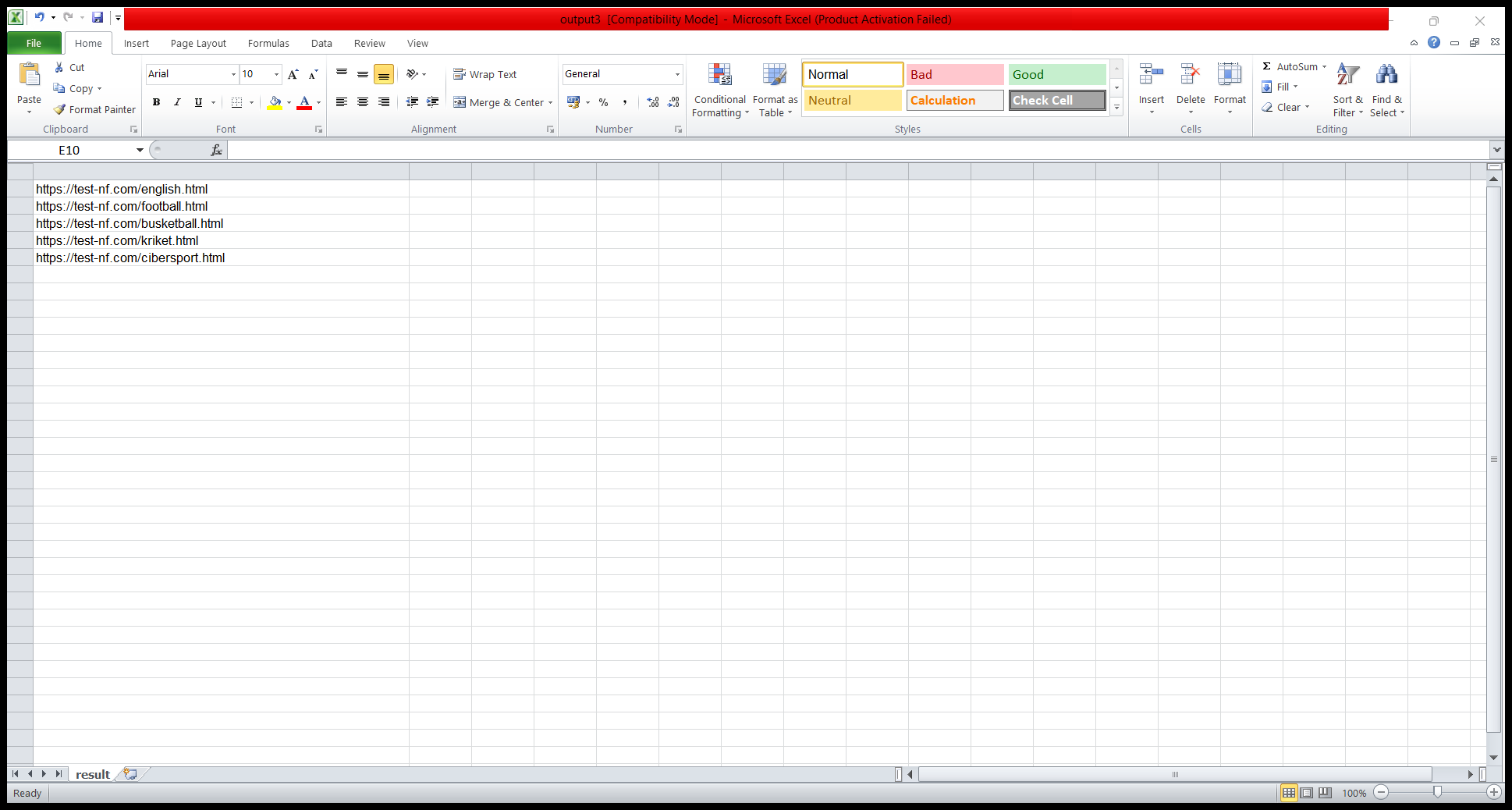
}

}

OUTPUT



The URL’s are successfully stored in excel sheet in .xls format.



TESTNG Report

