QUALITY ASSURANCE FINAL PROJECT

Submitted By – Roshan Paswan

[Fynd QA Batch- 7]

Manual Testing:

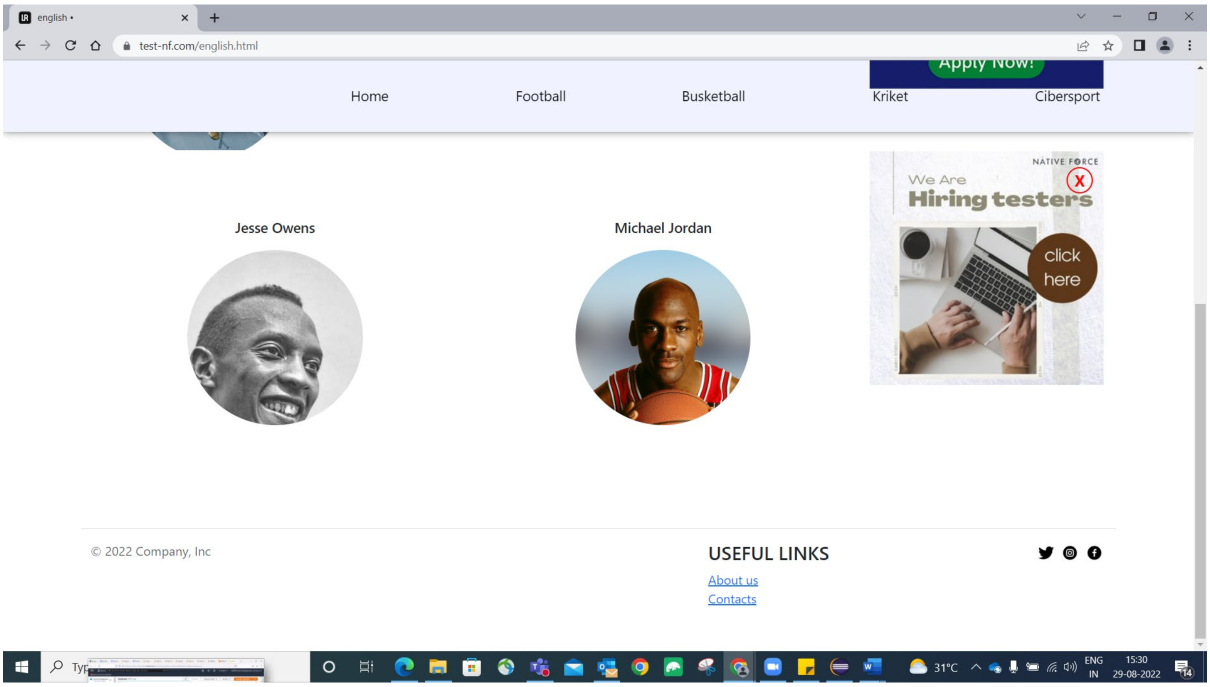
User Story 0001: Add “About Us” link option in footer of the application.

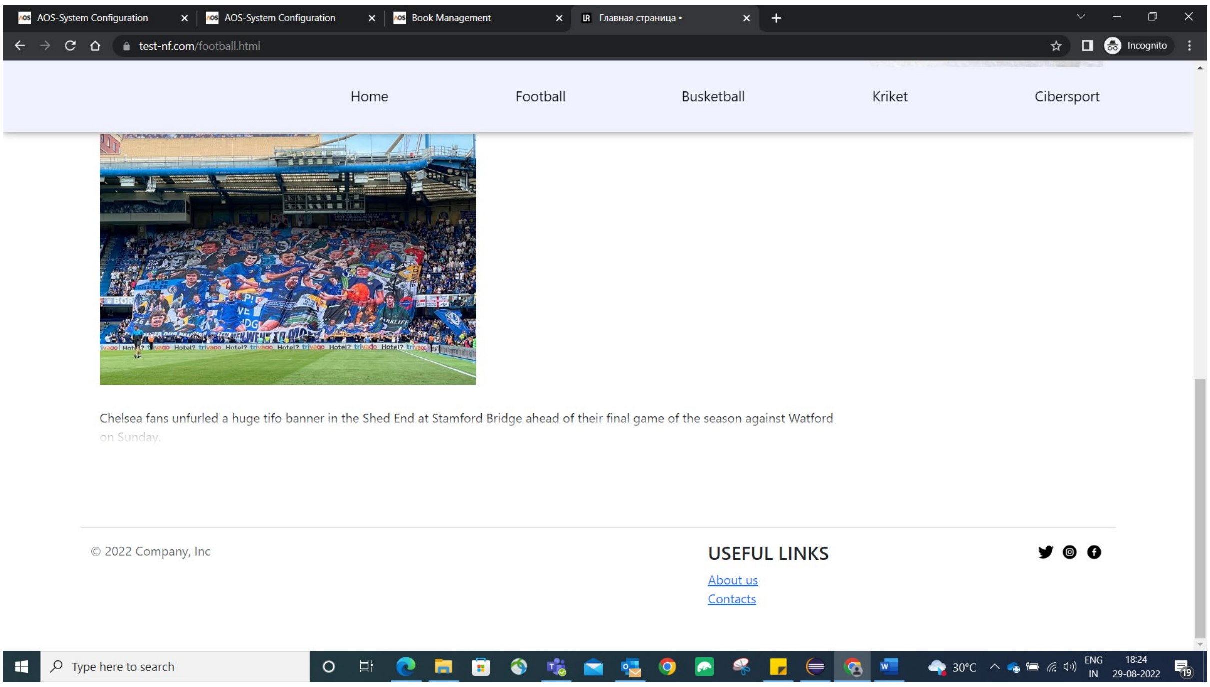
Description: The scope of this US is to provide ‘About us’ link so that users can click on the link and get all the details about the organization vision and mission.

Acceptance Criteria:

1. Users should be able to click on the ‘About us’ link
2. ‘About us’ link should be displayed in footer part (as per the mock-up)
3. When user click on the ‘About us’ link system will navigate to new page which display the Organization Vision and Mission.
4. ‘About us’ link should be available for all the tabs (Home, Cricket, Football, Basketball and Ciber Sports)

Mock-up:





Chelsea fans unfurled a huge tifo banner in the Shed End at Stamford Bridge ahead of their final game of the season against Watford on Sunday.

© 2022 Company, Inc

USEFUL LINKS

About us

Contacts

Scenario 1: Confirm the link "About us" is present and visible in footer on the webpage.

Sr. No.	Test Cases	Expected Results	Test Data	Result
1.	Launch the application and navigate to the homepage	User should be able to launch the application.	https://test-nf.com/english.html	Pass
2.	Scroll down to the bottom of the page and look at the footer section.	“About us” link will be present in footer section as per given mock-up.		Pass

Scenario 2: Verify whether “About us” link is clickable to user or not.

Sr. No.	Test Cases	Expected Results	Test Data	Result
1	Launch the application and navigate to the homepage	User should be able to launch the application.	https://test-nf.com/english.html	Pass
2	Scroll down to the bottom of the page and look at the footer section.	“About us” link will be present in footer section as per given mock-up.		Pass
3	Click on the "About Us"	Users will be able to click the 'About us' option.		Pass

Scenario 3: Validate whether after user click to “About us” option navigate to a new page or not.

Sr. No.	Test Cases	Expected Results	Test Data	Result
1	Launch the application and navigate to the homepage	User should be able to launch the application.	https://test-nf.com/english.html	PASS
2	Scroll down to the bottom of the page and look at the footer section.	“About us” link will be present in footer section as per given mock-up.		PASS
3	Click on the "About Us"	Users will be able to click the 'About us' option.		PASS

4	Check the scenario after clicking on “About us”.	“About us” option will navigate user to a new page after clicking.		PASS

Scenario 4: Confirm that upon clicking “About us” newly opened page display Organization Vision and Mission.

Sr. No.	Test Cases	Expected Results	Test Data	Result
1.	Launch the application and navigate to the homepage	User should be able to launch the application.	https://test-nf.com/english.html	PASS
2.	Scroll down to the bottom of the page and look at the footer section.	“About us” link will be present in footer section as per given mock-up.		PASS
3.	Click on the "About Us"	Users will be able to click the 'About us' option and “About us” option will navigate user to a new page.		PASS
4.	Inspect the newly opened page that it should displays organization vision and mission.	Users should be able to navigate to the newly opened About Us page which display about the Organization Vision and Mission.		FAIL

Scenario 05: Validate after clicking on each tab “About us” option is available for all the tabs.

Sr.No	Test Case	Expected Result	Test Data	Test Status
1.	Launch the application and navigate to the homepage	User should be able to launch the application.	https://test-nf.com/english.html	**Pass**
2.	Click on Home/Football/ Kriket/Busketball/Cibersports tabs	User will be able to click on all the tabs as per given mock-up.		**Pass**
3.	Scroll down to the bottom of the page and look at the footer section.	“About us” link will be present in footer section for all tabs.		**Pass**

Prepare Automation Test Script and perform the test execution.

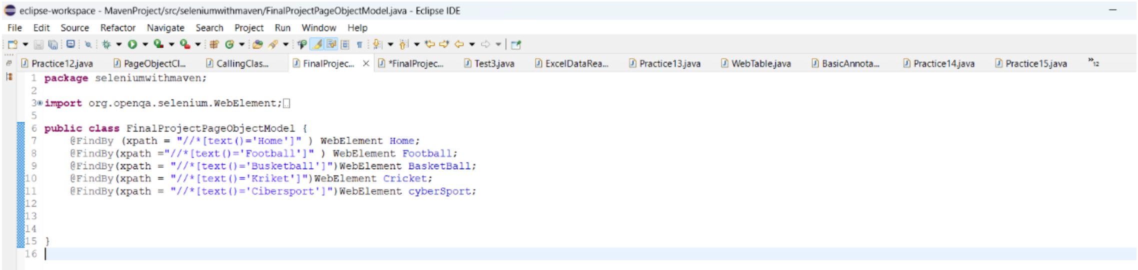
Automation Testing Scenario:

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.

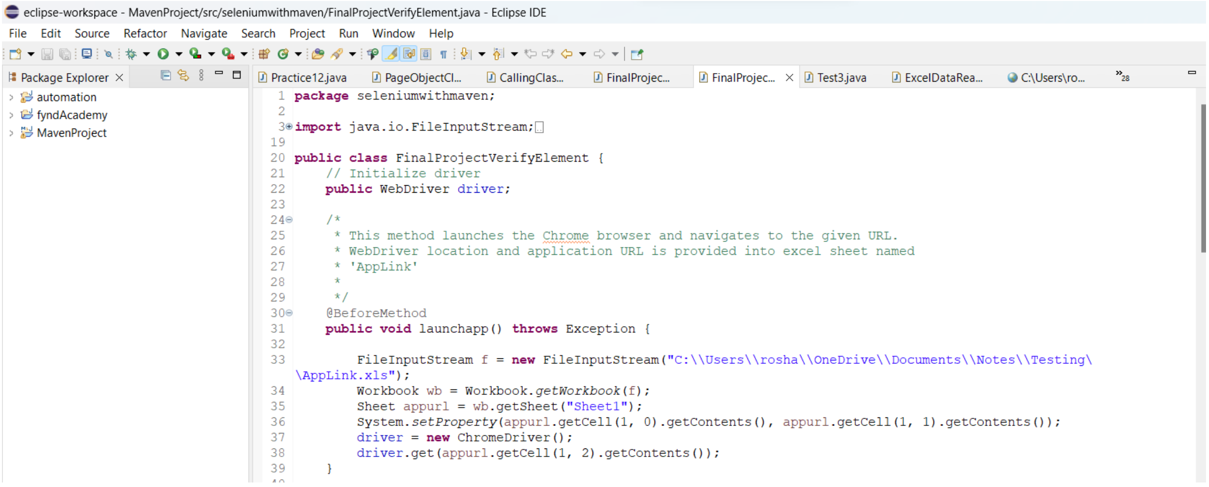
Note: Implement Page Object model, Assertion for Validation, Excel Sheet Integration for passing System properties and application URL.

Scenario 1: Create a Page Object Model to store the location of each tab element (Home, Cricket, Football, Basketball and Ciber Sports) using xpath.



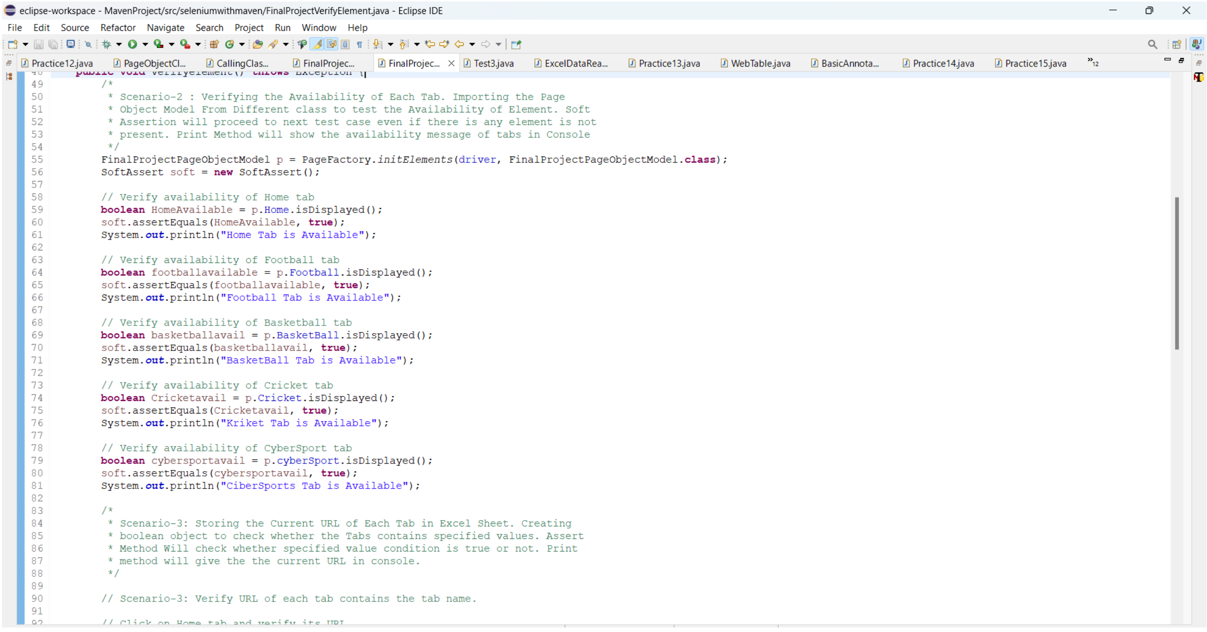
```
package seleniumwithmaven;

import org.openqa.selenium.WebElement;

public class FinalProjectPageObjectModel {
    @FindBy (xpath = "//*[text()='Home']" ) WebElement Home;
    @FindBy(xpath ="//*[text()='Football']" ) WebElement Football;
    @FindBy(xpath = "//*[text()='Busketball']")WebElement BasketBall;
    @FindBy(xpath = "//*[text()='Kriket']")WebElement Cricket;
    @FindBy(xpath = "//*[text()='Cibersport']")WebElement cyberSport;



}

```

Scenario-2: Create a method to launch the URL using excel sheet integration to pass system property for driver path and application url.



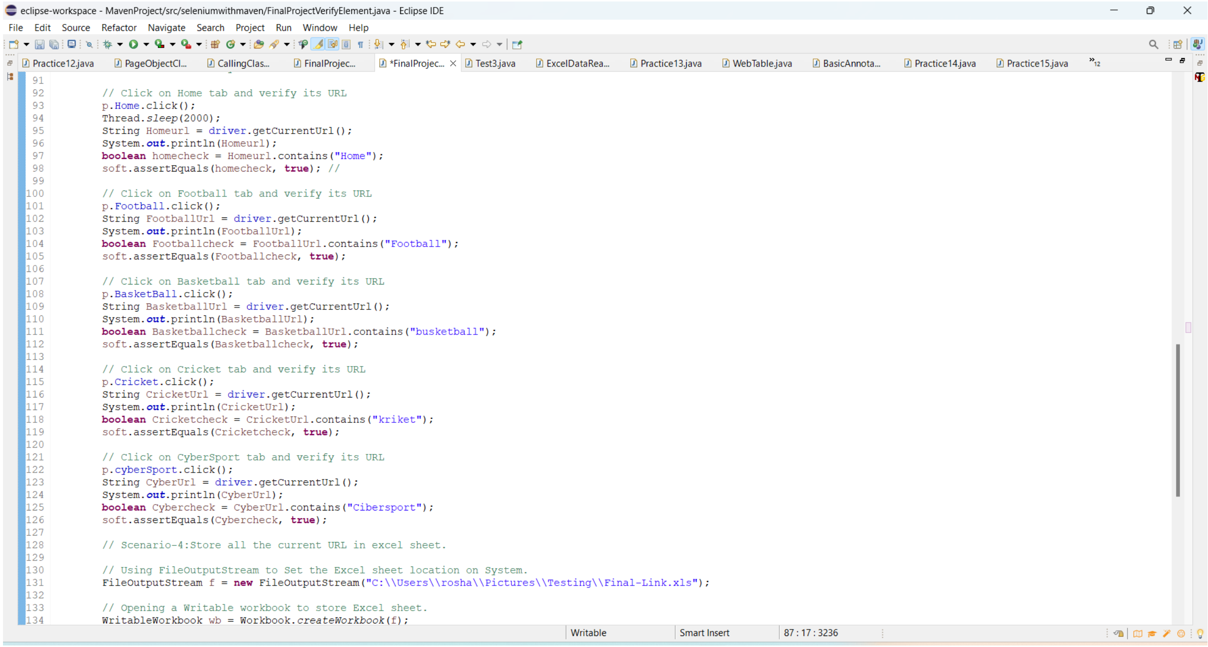
```
package seleniumwithmaven;

import java.io.FileInputStream;

public class FinalProjectVerifyElement {
    // Initialize driver
    public WebDriver driver;

    /*
     * This method launches the Chrome browser and navigates to the given URL.
     * WebDriver location and application URL is provided into excel sheet named
     * 'AppLink'
     *
     */
    @BeforeMethod
    public void launchapp() throws Exception {

        FileInputStream f = new FileInputStream("C:\\Users\\rosha\\OneDrive\\Documents\\Notes\\Testing\
\AppLink.xls");
        Workbook wb = Workbook.getWorkbook(f);
        Sheet appurl = wb.getSheet("Sheet1");
        System.setProperty(appurl.getCell(1, 0).getContents(), appurl.getCell(1, 1).getContents());
        driver = new ChromeDriver();
        driver.get(appurl.getCell(1, 2).getContents());
    }
```

Scenario-3: Create a method to check element availability and import page object model in this method.


eclipse-workspace - MavenProject/src/seleniumwithmaven/FinalProjectVerifyElement.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Practice12.java PageObjectCl... CallingClas... FinalProjec... FinalProjec... Test3.java ExcelDataRea... Practice13.java WebTable.java BasicAnnota... Practice14.java Practice15.java

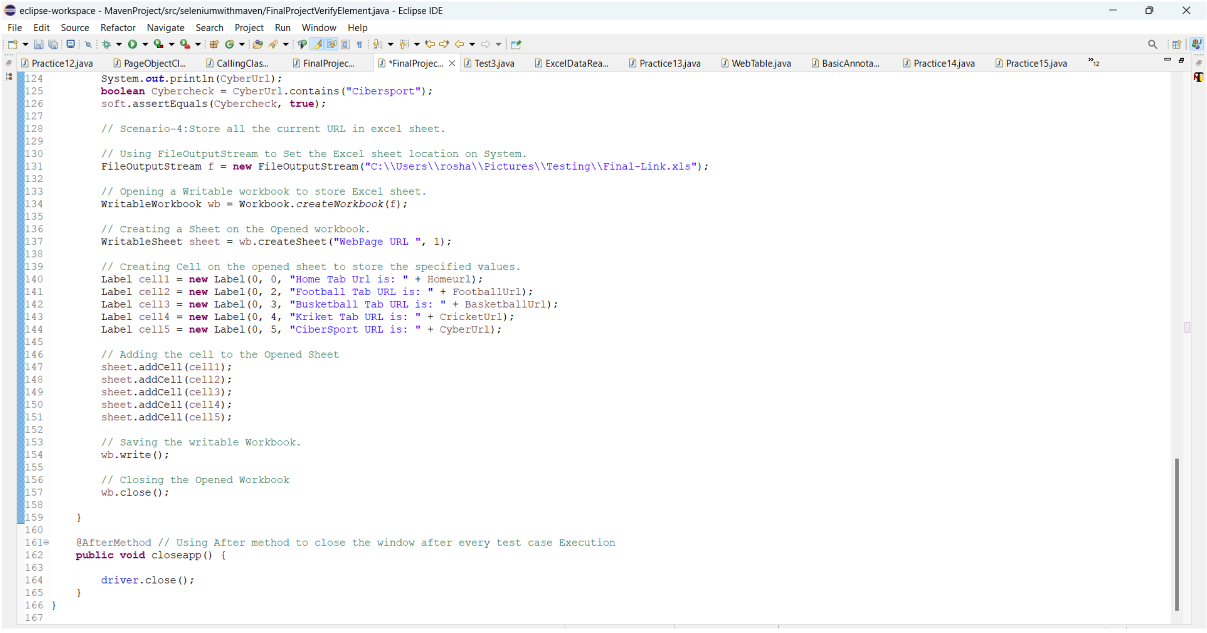
```java
    public void verifyElement() throws Exception {
        /*
         * Scenario-2 : Verifying the Availability of Each Tab. Importing the Page
         * Object Model From Different class to test the Availability of Element. Soft
         * Assertion will proceed to next test case even if there is any element is not
         * present. Print Method will show the availability message of tabs in Console
         */
        FinalProjectPageObjectModel p = PageFactory.initElements(driver, FinalProjectPageObjectModel.class);
        SoftAssert soft = new SoftAssert();

        // Verify availability of Home tab
        boolean HomeAvailable = p.Home.isDisplayed();
        soft.assertEquals(HomeAvailable, true);
        System.out.println("Home Tab is Available");

        // Verify availability of Football tab
        boolean footballavailable = p.Football.isDisplayed();
        soft.assertEquals(footballavailable, true);
        System.out.println("Football Tab is Available");

        // Verify availability of Basketball tab
        boolean basketballavail = p.BasketBall.isDisplayed();
        soft.assertEquals(basketballavail, true);
        System.out.println("BasketBall Tab is Available");

        // Verify availability of Cricket tab
        boolean Cricketavail = p.Cricket.isDisplayed();
        soft.assertEquals(Cricketavail, true);
        System.out.println("Kriket Tab is Available");

        // Verify availability of CyberSport tab
        boolean cybersportavail = p.cyberSport.isDisplayed();
        soft.assertEquals(cybersportavail, true);
        System.out.println("CiberSports Tab is Available");

        /*
         * Scenario-3: Storing the Current URL of Each Tab in Excel Sheet. Creating
         * boolean object to check whether the Tabs contains specified values. Assert
         * Method Will check whether specified value condition is true or not. Print
         * method will give the the current URL in console.
         */

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

        // Click on Home tab and verify its URL
```

Scenario-4: Verify the URL of each tab that contains tab name.


eclipse-workspace - MavenProject/src/seleniumwithmaven/FinalProjectVerifyElement.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Practice12.java PageObjectCl... CallingClas... FinalProjec... *FinalProjec... Test3.java ExcelDataRea... Practice13.java WebTable.java BasicAnnota... Practice14.java Practice15.java
Writable
Smart Insert
87 : 17 : 3236

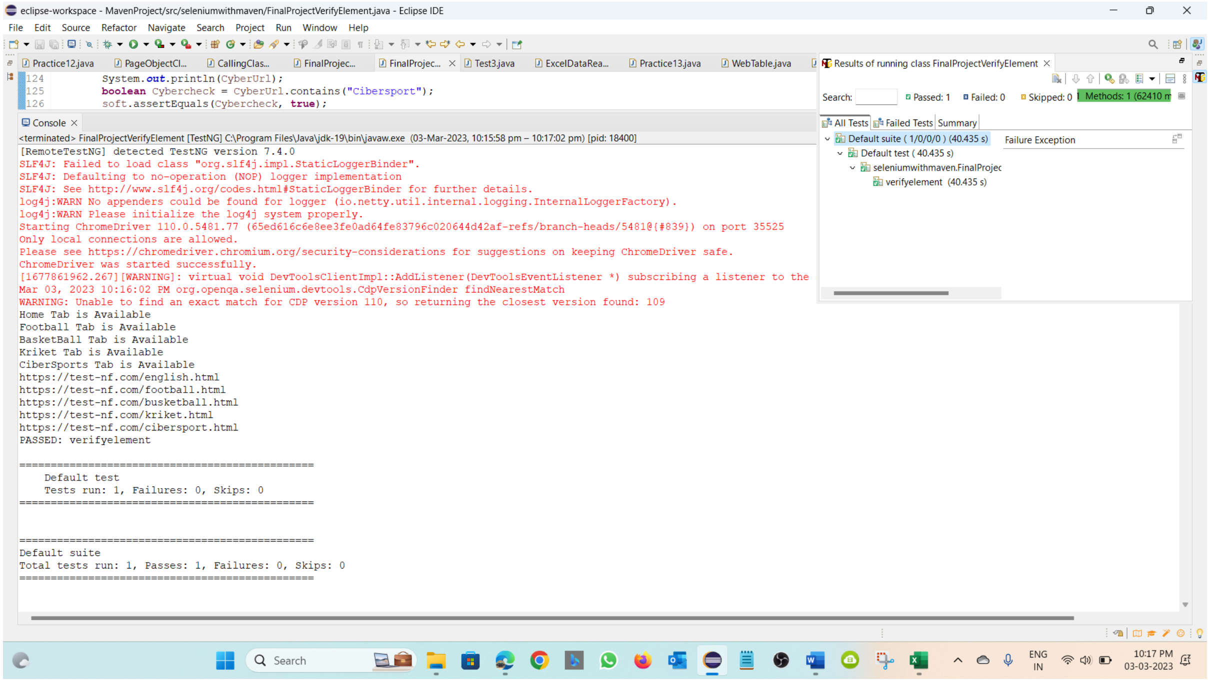
```java

        // Click on Home tab and verify its URL
        p.Home.click();
        Thread.sleep(2000);
        String Homeurl = driver.getCurrentUrl();
        System.out.println(Homeurl);
        boolean homecheck = Homeurl.contains("Home");
        soft.assertEquals(homecheck, true); //

        // Click on Football tab and verify its URL
        p.Football.click();
        String FootballUrl = driver.getCurrentUrl();
        System.out.println(FootballUrl);
        boolean Footballcheck = FootballUrl.contains("Football");
        soft.assertEquals(Footballcheck, true);

        // Click on Basketball tab and verify its URL
        p.BasketBall.click();
        String BasketballUrl = driver.getCurrentUrl();
        System.out.println(BasketballUrl);
        boolean Basketballcheck = BasketballUrl.contains("busketball");
        soft.assertEquals(Basketballcheck, true);

        // Click on Cricket tab and verify its URL
        p.Cricket.click();
        String CricketUrl = driver.getCurrentUrl();
        System.out.println(CricketUrl);
        boolean Cricketcheck = CricketUrl.contains("kriket");
        soft.assertEquals(Cricketcheck, true);

        // Click on CyberSport tab and verify its URL
        p.cyberSport.click();
        String CyberUrl = driver.getCurrentUrl();
        System.out.println(CyberUrl);
        boolean Cybercheck = CyberUrl.contains("Cibersport");
        soft.assertEquals(Cybercheck, true);

        // Scenario-4:Store all the current URL in excel sheet.

        // Using FileOutputStream to Set the Excel sheet location on System.
        FileOutputStream f = new FileOutputStream("C:\\Users\\rosha\\Pictures\\Testing\\Final-Link.xls");

        // Opening a Writable workbook to store Excel sheet.
        WritableWorkbook wb = Workbook.createWorkbook(f);
```

Scenario-5: Create an Excel sheet to store each tab current Link.



```java
        System.out.println(CyberUrl);
        boolean Cybercheck = CyberUrl.contains("Cibersport");
        soft.assertEquals(Cybercheck, true);

        // Scenario-4:Store all the current URL in excel sheet.

        // Using FileOutputStream to Set the Excel sheet location on System.
        FileOutputStream f = new FileOutputStream("C:\\Users\\rosha\\Pictures\\Testing\\Final-Link.xls");

        // Opening a Writable workbook to store Excel sheet.
        WritableWorkbook wb = Workbook.createWorkbook(f);

        // Creating a Sheet on the Opened workbook.
        WritableSheet sheet = wb.createSheet("WebPage URL ", 1);

        // Creating Cell on the opened sheet to store the specified values.
        Label cell1 = new Label(0, 0, "Home Tab Url is: " + Homeurl);
        Label cell2 = new Label(0, 2, "Football Tab URL is: " + FootballUrl);
        Label cell3 = new Label(0, 3, "Busketball Tab URL is: " + BasketballUrl);
        Label cell4 = new Label(0, 4, "Kriket Tab URL is: " + CricketUrl);
        Label cell5 = new Label(0, 5, "CiberSport URL is: " + CyberUrl);

        // Adding the cell to the Opened Sheet
        sheet.addCell(cell1);
        sheet.addCell(cell2);
        sheet.addCell(cell3);
        sheet.addCell(cell4);
        sheet.addCell(cell5);

        // Saving the writable Workbook.
        wb.write();

        // Closing the Opened Workbook
        wb.close();

    }

    @AfterMethod // Using After method to close the window after every test case Execution
    public void closeapp() {

        driver.close();
    }
}
```

Eclipse Console Output:



```java
        System.out.println(CyberUrl);
        boolean Cybercheck = CyberUrl.contains("Cibersport");
        soft.assertEquals(Cybercheck, true);
```

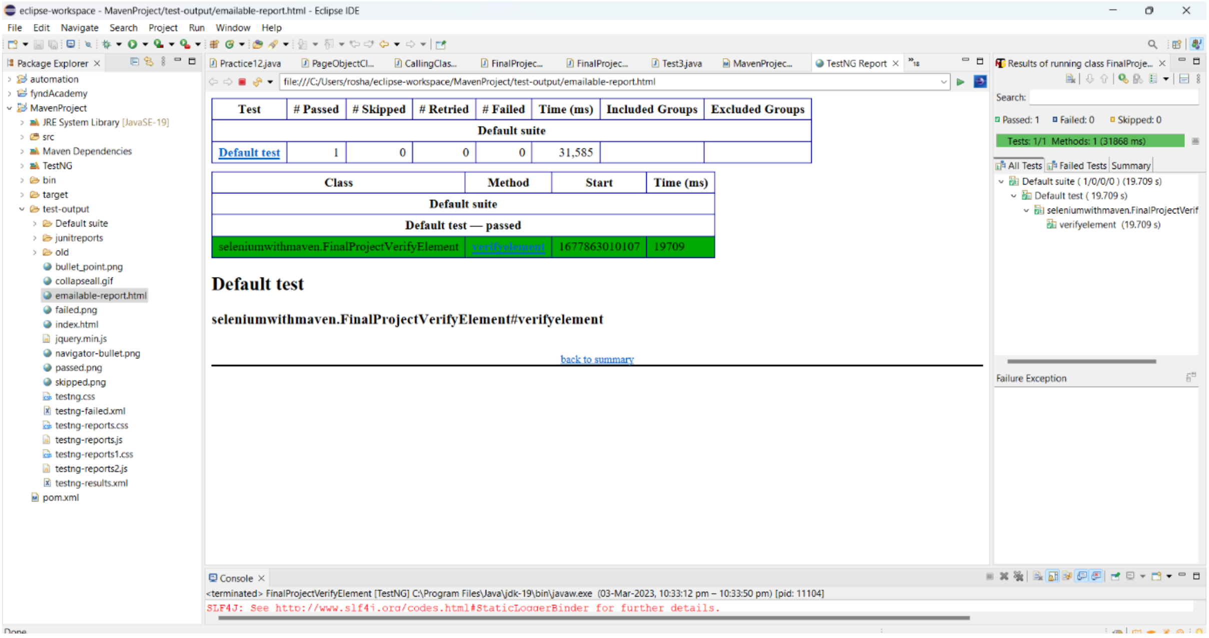
<terminated> FinalProjectVerifyElement [TestNG] C:\Program Files\Java\jdk-19\bin\javaw.exe (03-Mar-2023, 10:15:58 pm – 10:17:02 pm) [pid: 18400]

```
[RemoteTestNG] detected TestNG version 7.4.0
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
log4j:WARN No appenders could be found for logger (io.netty.util.internal.logging.InternalLoggerFactory).
log4j:WARN Please initialize the log4j system properly.
Starting ChromeDriver 110.0.5481.77 (65ed616c6e8ee3fe0ad64fe83796c020644d42af-refs/branch-heads/5481@{#839}) on port 35525
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
[1677861962.267][WARNING]: virtual void DevToolsClientImpl::AddListener(DevToolsEventListener *) subscribing a listener to the
Mar 03, 2023 10:16:02 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 110, so returning the closest version found: 109
Home Tab is Available
Football Tab is Available
BasketBall Tab is Available
Kriket Tab is Available
CiberSports Tab is Available
https://test-nf.com/english.html
https://test-nf.com/football.html
https://test-nf.com/busketball.html
https://test-nf.com/kriket.html
https://test-nf.com/cibersport.html
PASSED: verifyelement

===============================================
    Default test
    Tests run: 1, Failures: 0, Skips: 0
===============================================


===============================================
Default suite
Total tests run: 1, Passes: 1, Failures: 0, Skips: 0
===============================================
```

Search: Passed: 1 Failed: 0 Skipped: 0 1 Methods: 1 (62410 m

- Default suite (1/0/0/0) (40.435 s)
  - Default test (40.435 s)
    - seleniumwithmaven.FinalProjec
      - verifyelement (40.435 s)

TestNG Emailable Report:



Test	# Passed	# Skipped	# Retried	# Failed	Time (ms)	Included Groups	Excluded Groups
Default suite							
Default test	1	0	0	0	31,585		

Class	Method	Start	Time (ms)
Default suite			
Default test — passed			
seleniumwithmaven.FinalProjectVerifyElement	verifyelement	1677863010107	19709

Default test

seleniumwithmaven.FinalProjectVerifyElement#verifyelement

back to summary

```
<terminated> FinalProjectVerifyElement [TestNG] C:\Program Files\Java\jdk-19\bin\javaw.exe  (03-Mar-2023, 10:33:12 pm – 10:33:50 pm) [pid: 11104]
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
```

BUG REPORT:

Project URL: https://test-nf.com/english.html

Sr.No	BUGS	SEVERITY	PRIORITY
	Homepage		
1.	The social media handles are in footer is not clickable.	**S2**	**P2**
2.	The Ads are overlapping with the navigation bar while scrolling the home page.	**S4**	**P4**
3.	On hovering the “Basketball” tab the text of Basketball gets disappear.	**S3**	**P3**
4.	On clicking “TESTERS” and “Apply Now” button, both are navigated to same page.	**S3**	**P2**

5.	On clicking "Click here" button in Ads it only refreshes the page.	**S3**	**P2**
6.	On clicking "About us" link, it will not navigate to any valid page, it navigates to a new page which is holding 404 error.	**S2**	**P2**
7.	On clicking "About us" link, it will navigate to a new page which is holding 404 error.	**S2**	**P2**
8.	Layout of image are not uniform	**S4**	**P4**
9.	In header section, football menu orange background highlight is not covering the full border length.	**S1**	**P1**
10.	In header section, Basketball, Cricket and Cybersport spelling is wrong.	**S4**	**P4**
11.	In header section the titles are not symmetric and not properly aligned	**S1**	**P1**
12.	In body section, learn more information text is overlapped with ads.	**S2**	**P2**
13.	In body section, if we click learn more on Michael Jordon we unable to close information box, the icon is not visible.	**S2**	**P2**
14.	In body section if we click on learn more button of Michael Jordon, we unable to view any information. it's Blank	**S2**	**P2**
15.	Image of Lionel Messi is incorrect.	**S2**	**P2**
	FOOTBALL		
16.	First story article In Football page's the text is not fully visible and there is no option to expand the story or view the full article.	**S2**	**P2**
17.	In the Football page the image given is not related to football.	**S2**	**P2**
	BASKETBALL		
18.	In the Busketball page second story article the image is not loading.	**S2**	**P2**

19.	In the Busketball page's the articles texts are not fully visible and there is no option to expand it or view the full version of it.	**S2**	**S3**
20.	The spelling of Basketball is wrong.	**S3**	**P3**
	CRICKET		
21.	In the Kriket page's both story articles text is not fully visible and there is no option to expand the story or view the full version of it	**S3**	**P4**
22..	In the Kriket page's second story article text is in French Language.	**S3**	**P4**
23.	The spelling of Cricket in the body is wrong	**S3**	**P3**
	CIBERSPORT		
24.	There is no option to expand the story or view the full version of it.	**S3**	**P3**
25.	In the Cibersport page the first story article text is not fully visible and there is no option to expand it.	**S3**	**P3**
26.	The spelling of Cybersport in the body is wrong.	**S3**	**S3**