Capstone Project - Testing an Airline Booking Website - Source Code

Git URL :

https://github.com/venkatasaiswabhanu/FlyAwayApplication

Features :-

signupFunctionality.feature :

```
Feature: A user should be able to sign up

Scenario: Sign Up Functionality

Given A user clicks on Login/SignUp link text in Landing page of flyaway application
When he clicks on Not a member? Signup button
And pass Email id as "<email>",Password as "<Password>",Confirm Password as "<Password>", Name as
"<Name>", adress as "<Address>" and city as "<City>"
And click on Sign Up button
Then He should be able to see "Login to continue checking flights" message

Examples:

|email              |Password  |Name                 |Address     |City |
|vburugad@cisco.com|BhanuBK@99|Venkat Sai Swabhanu BK|Ramalayam Street|Lakkavaram    |
```

flightBookingFunctionality.feature :

```
Feature: A user should be able to Book available Flights

Scenario: Flight Booking Functionality

Given A user click on Login/SignUp link text in Landing page of flyaway application
When he pass email as "<email>" and Password as "<Password>"
And click on login button he will be successfully navigated to Dashboard page
And he click on Home
And click on first available flight he will navigate to confirmation page
And if he click on Click to complete booking
Then he should be able to see " FLYAWAY - BOOKING CONFIRMATION " message

Examples:

|email                     |Password                                   |
|vburugad@cisco.com        |BhanuBK@99                                 |
```

POM.xml :-

```
<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.flyaway</groupId>
  <artifactId>FlyAwayApplication</artifactId>
  <version>0.0.1-SNAPSHOT</version>

  <name>FlyAwayApplication</name>
  <!-- FIXME change it to the project's website -->
  <url>http://www.example.com</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.7</maven.compiler.source>
    <maven.compiler.target>1.7</maven.compiler.target>
  </properties>


  <dependencies>

  <!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-core -->
 <dependency>
    <groupId>io.cucumber</groupId>
    <artifactId>cucumber-core</artifactId>
    <version>7.2.3</version>
 </dependency>

 <!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-java -->
 <dependency>
    <groupId>io.cucumber</groupId>
    <artifactId>cucumber-java</artifactId>
    <version>7.2.3</version>
 </dependency>

 <!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-testng -->
 <dependency>
    <groupId>io.cucumber</groupId>
    <artifactId>cucumber-testng</artifactId>
    <version>7.2.3</version>
 </dependency>

 <!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-jvm-deps -->
 <dependency>
    <groupId>io.cucumber</groupId>
    <artifactId>cucumber-jvm-deps</artifactId>
    <version>1.0.6</version>
    <scope>provided</scope>
 </dependency>

 <!-- https://mvnrepository.com/artifact/net.masterthought/cucumber-reporting -->
 <dependency>
    <groupId>net.masterthought</groupId>
    <artifactId>cucumber-reporting</artifactId>
    <version>5.6.1</version>
 </dependency>

 <!-- https://mvnrepository.com/artifact/io.cucumber/gherkin -->
 <dependency>
    <groupId>io.cucumber</groupId>
    <artifactId>gherkin</artifactId>
    <version>22.0.0</version>
 </dependency>
```

```xml
<!-- https://mvnrepository.com/artifact/org.testng/testng -->
<dependency>
    <groupId>org.testng</groupId>
    <artifactId>testng</artifactId>
    <version>7.5</version>
</dependency>

<!-- https://mvnrepository.com/artifact/com.beust/jcommander -->
<dependency>
    <groupId>com.beust</groupId>
    <artifactId>jcommander</artifactId>
    <version>1.82</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
<dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-java</artifactId>
    <version>3.141.59</version>
</dependency>

<dependency>
    <groupId>com.aventstack</groupId>
    <artifactId>extentreports</artifactId>
    <version>3.1.5</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.apache.logging.log4j/log4j-core -->
<dependency>
            <groupId>org.apache.logging.log4j</groupId>
            <artifactId>log4j-core</artifactId>
            <version>2.12.1</version>
</dependency>

<!-- https://mvnrepository.com/artifact/io.rest-assured/json-schema-validator -->
<dependency>
            <groupId>io.rest-assured</groupId>
            <artifactId>json-schema-validator</artifactId>
            <version>4.1.2</version>
</dependency>

<!-- https://mvnrepository.com/artifact/io.rest-assured/json-path -->
<dependency>
            <groupId>io.rest-assured</groupId>
            <artifactId>json-path</artifactId>
            <version>4.1.1</version>
</dependency>

<!-- https://mvnrepository.com/artifact/log4j/log4j -->
<dependency>
            <groupId>log4j</groupId>
            <artifactId>log4j</artifactId>
            <version>1.2.17</version>
</dependency>

<dependency>
            <groupId>org.apache.logging.log4j</groupId>
            <artifactId>log4j-api</artifactId>
            <version>2.11.1</version>
```

```xml
</dependency>

<dependency>
            <groupId>org.apache.logging.log4j</groupId>
            <artifactId>log4j-core</artifactId>
            <version>2.11.1</version>
            </dependency>

<dependency>
            <groupId>org.apache.logging.log4j</groupId>
            <artifactId>log4j-1.2-api</artifactId>
            <version>2.11.1</version>
</dependency>

<!-- https://mvnrepository.com/artifact/io.rest-assured/rest-assured -->
<dependency>
            <groupId>io.rest-assured</groupId>
            <artifactId>rest-assured</artifactId>
            <version>3.3.0</version>
</dependency>

<!-- https://mvnrepository.com/artifact/com.googlecode.json-simple/json-simple -->
<dependency>
            <groupId>com.googlecode.json-simple</groupId>
            <artifactId>json-simple</artifactId>
            <version>1.1.1</version>
</dependency>
<dependency>
             <groupId>org.hamcrest</groupId>
             <artifactId>hamcrest-all</artifactId>
            <version>1.3</version>
             <scope>test</scope>
 </dependency>

 <dependency>
            <groupId>org.mockito</groupId>
            <artifactId>mockito-all</artifactId>
             <version>1.9.5</version>
             <scope>test</scope>
 </dependency>

 <!-- https://mvnrepository.com/artifact/io.rest-assured/json-schema-validator -->
<dependency>
            <groupId>io.rest-assured</groupId>
            <artifactId>json-schema-validator</artifactId>
            <version>4.1.2</version>
</dependency>
<dependency>
            <groupId>javax.activation</groupId>
            <artifactId>activation</artifactId>
            <version>1.1</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.hamcrest/java-hamcrest -->
<dependency>
            <groupId>org.hamcrest</groupId>
            <artifactId>java-hamcrest</artifactId>
            <version>2.0.0.0</version>
            <scope>test</scope>
</dependency>
```

```
<!-- https://mvnrepository.com/artifact/org.hamcrest/hamcrest-core -->
<dependency>
            <groupId>org.hamcrest</groupId>
            <artifactId>hamcrest-core</artifactId>
            <version>2.2-rc1</version>
            <scope>test</scope>
</dependency>

<!-- https://mvnrepository.com/artifact/org.hamcrest/hamcrest-library -->
<dependency>
            <groupId>org.hamcrest</groupId>
            <artifactId>hamcrest-library</artifactId>
            <version>2.2-rc1</version>
            <scope>test</scope>
</dependency>

<!-- https://mvnrepository.com/artifact/io.rest-assured/xml-path -->
<dependency>
            <groupId>io.rest-assured</groupId>
            <artifactId>xml-path</artifactId>
            <version>3.0.0</version>
            </dependency>

<dependency>
            <groupId>com.sun.mail</groupId>
            <artifactId>javax.mail</artifactId>
            <version>1.6.2</version>
            </dependency>

  </dependencies>

  <build>
    <pluginManagement><!-- lock down plugins versions to avoid using Maven defaults (may be moved to
parent pom) -->
      <plugins>
        <!-- clean lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#clean_Lifecycle -
->
        <plugin>
          <artifactId>maven-clean-plugin</artifactId>
          <version>3.1.0</version>
        </plugin>
        <!-- default lifecycle, jar packaging: see https://maven.apache.org/ref/current/maven-core/default-
bindings.html#Plugin_bindings_for_jar_packaging -->
        <plugin>
          <artifactId>maven-resources-plugin</artifactId>
          <version>3.0.2</version>
        </plugin>
        <plugin>
          <artifactId>maven-compiler-plugin</artifactId>
          <version>3.8.0</version>
        </plugin>
        <plugin>
          <artifactId>maven-surefire-plugin</artifactId>
          <version>2.22.1</version>
        </plugin>
        <plugin>
          <artifactId>maven-jar-plugin</artifactId>
          <version>3.0.2</version>
        </plugin>
        <plugin>
```

```
          <artifactId>maven-install-plugin</artifactId>
          <version>2.5.2</version>
        </plugin>
        <plugin>
          <artifactId>maven-deploy-plugin</artifactId>
          <version>2.8.2</version>
        </plugin>
        <!-- site lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#site_Lifecycle -->
        <plugin>
          <artifactId>maven-site-plugin</artifactId>
          <version>3.7.1</version>
        </plugin>
        <plugin>
          <artifactId>maven-project-info-reports-plugin</artifactId>
          <version>3.0.0</version>
        </plugin>
      </plugins>
    </pluginManagement>
  </build>
</project>
```

TESTNG.xml:-

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Suite">

<listeners>
<listener class-name="com.Listeners.TestListner"></listener>
</listeners>

  <test thread-count="5" name="Test">
    <classes>
      <class name="com.flyaway.testscripts.TestRunner"/>

    </classes>
  </test> <!-- Test -->
</suite> <!-- Suite -->
```

/src/main/java :-

com.flyaway.pages :-

LandingPage.java :

```
package com.flyaway.pages;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;

public class LandingPage {

	@FindBy(linkText = "Login/Signup")
	private WebElement loginSignUp;
	@FindBy(linkText = "Not a member? Signup")
	private WebElement signUp;


	public LandingPage(WebDriver driver) {
		PageFactory.initElements(driver, this);
	}

	public void clickLoginSignup() {
		loginSignUp.click();
	}

	public void clickSignUp() {
		signUp.click();
	}
}
```

SignUpPage.java :

```
package com.flyaway.pages;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.interactions.Actions;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;

public class SignUpPage {

	@FindBy(xpath = "//input[contains(@name,'email_id')]")
	private WebElement emailaddr;

	@FindBy(xpath = "//input[contains(@name,'pwd')]")
	private WebElement password;

	@FindBy(xpath = "//input[contains(@name,'pwd2')]")
	private WebElement cnfpassword;

	@FindBy(xpath = "//input[contains(@name,'name')]")
	private WebElement name;

	@FindBy(xpath = "//input[contains(@name,'address')]")
	private WebElement address;

	@FindBy(xpath = "//input[contains(@name,'city')]")
	private WebElement city;

	//body > form > table > tbody > tr:nth-child(7) > td > button
	@FindBy(css = "body > form > table > tbody > tr:nth-child(7) > td > button")
	private WebElement signupBtn;

	@FindBy(css = "body > form > table > tbody > tr:nth-child(3) > td > button")
	private WebElement loginBtn;

	@FindBy(linkText = "Login to continue checking flights")
	private WebElement cnfmsg;
	public SignUpPage(WebDriver driver) {
		PageFactory.initElements(driver, this);
		Actions actions = new Actions(driver);
	}

	public void enteremail(String email) {
		emailaddr.sendKeys(email);
	}

	public void pwd(String Password) {
		password.sendKeys(Password);
	}

	public void pwd2(String Password) {
		cnfpassword.sendKeys(Password);
	}

	public void name(String Name) {
		name.sendKeys(Name);
	}

	public void address(String Address) {
		address.sendKeys(Address);
	}

	public void city(String City) {
		city.sendKeys(City);
	}

	public void clickSignUpbtn() {
		signupBtn.click();
	}
	public String getTxt() {
		String text = cnfmsg.getText();
		return text;
	}
	public void clickLogin() {
		loginBtn.click();
	}

}
```

HomePage.java :

```
package com.flyaway.pages;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.interactions.Actions;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;

public class HomePage {

	@FindBy(linkText = "Home")
	private WebElement home;

	@FindBy(xpath = "(//option[contains(@value,'2')])[1]")
	private WebElement source;
```

```
        @FindBy(xpath = "(//option[contains(@value,'1')])[2]")
        private WebElement destination;

        @FindBy(css = "body > form > table > tbody > tr > td:nth-child(3) > button")
        private WebElement submit;

        @FindBy(linkText = "Book Flight")
        private WebElement bookFlight;

        public HomePage(WebDriver driver) {
                PageFactory.initElements(driver, this);
                Actions actions = new Actions(driver);
        }

        public void clickHome() {
                home.click();
        }

        public void clickSource() {
                source.click();
        }

        public void clickDestination() {
                destination.click();
        }
        public void clickSubmit() {
                submit.click();
        }
        public void clickBookFlight() {
                bookFlight.click();
        }

}
```

PaymentGateWayPage.java :

```
package com.flyaway.pages;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.interactions.Actions;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;

public class PaymentGatewayPage {
        @FindBy(linkText = "Click to complete booking")
        private WebElement cnfBooking;

        @FindBy(css = "body > h3")
        private WebElement cnftxt;

        public PaymentGatewayPage(WebDriver driver) {
                PageFactory.initElements(driver, this);
                Actions actions = new Actions(driver);
        }
        public void clickto() {
                cnfBooking.click();
        }
        public String getcnftxt() {
                String cnftext = cnftxt.getText();
                return cnftext;
        }

}
```

Com.Listners :

ExtentManager.java :

```
package com.Listeners;
import java.io.File;
import com.aventstack.extentreports.ExtentReports;
import com.aventstack.extentreports.reporter.ExtentHtmlReporter;
import com.aventstack.extentreports.reporter.configuration.ChartLocation;
import com.aventstack.extentreports.reporter.configuration.Theme;

public class ExtentManager {
        private static ExtentReports extent;
          private static String reportFileName = "Test-Automation-Report"+".html";
          private static String fileSeperator = System.getProperty("file.separator");
          private static String reportFilepath = System.getProperty("user.dir") +fileSeperator+ "TestReport";
          private static String reportFileLocation =  reportFilepath +fileSeperator+ reportFileName;

          public static ExtentReports getInstance() {
              if (extent == null)
                  createInstance();
              return extent;
          }

          //Create an extent report instance
          public static ExtentReports createInstance() {
              String fileName = getReportPath(reportFilepath);

              ExtentHtmlReporter htmlReporter = new ExtentHtmlReporter(fileName);
              htmlReporter.config().setTestViewChartLocation(ChartLocation.TOP);
              htmlReporter.config().setChartVisibilityOnOpen(true);
              htmlReporter.config().setTheme(Theme.STANDARD);
              htmlReporter.config().setDocumentTitle(reportFileName);
              htmlReporter.config().setEncoding("utf-8");
              htmlReporter.config().setReportName(reportFileName);
              htmlReporter.config().setTimeStampFormat("EEEE, MMMM dd, yyyy, hh:mm a '('zzz')'");
              extent = new ExtentReports();
              extent.attachReporter(htmlReporter);
              //Set environment details
                          extent.setSystemInfo("OS", "Windows");
                          extent.setSystemInfo("AUT", "QA");

              return extent;
          }
          //Create the report path
          private static String getReportPath (String path) {
                  File  testDirectory = new File(path);
              if (!testDirectory.exists()) {
                  if (testDirectory.mkdir()) {
                    System.out.println("Directory: " + path + " is created!" );
                    return reportFileLocation;
                  }
          else
                  {
                    System.out.println("Failed to create directory: " + path);
                    return System.getProperty("user.dir");
                  }
              } else {
                  System.out.println("Directory already exists: " + path);
              }
                          return reportFileLocation;
          }
}
```

ExtentTestManager.java :

```
package com.Listeners;
import java.util.HashMap;
import java.util.Map;
import com.aventstack.extentreports.ExtentReports;
import com.aventstack.extentreports.ExtentTest;

public class ExtentTestManager {
        static Map<Integer, ExtentTest> extentTestMap = new HashMap<Integer, ExtentTest>();
        static ExtentReports extent = ExtentManager.getInstance();

        public static synchronized ExtentTest getTest() {
                return (ExtentTest) extentTestMap.get((int) (long) (Thread.currentThread().getId()));
        }

        public static synchronized void endTest() {
                extent.flush();
        }

        public static synchronized ExtentTest startTest(String testName) {
                ExtentTest test = extent.createTest(testName);
                extentTestMap.put((int) (long) (Thread.currentThread().getId()), test);
                return test;
        }

}
```

TestListner.java :

```
package com.Listeners;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.text.SimpleDateFormat;
import org.apache.log4j.Logger;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.io.FileHandler;
import org.testng.ITestContext;
import org.testng.ITestListener;
import org.testng.ITestResult;
import com.aventstack.extentreports.MediaEntityBuilder;
import com.aventstack.extentreports.Status;
import com.aventstack.extentreports.model.Log;
```

```
public class TestListner implements ITestListener {
    private static Logger log = Logger.getLogger(Log.class.getName());

    public void onStart(ITestContext context) {
        System.out.println("*** Test Suite " + context.getName() + " started ***");
    }

    public void onFinish(ITestContext context) {
        System.out.println(("*** Test Suite " + context.getName() + " ending ***"));
        ExtentTestManager.endTest();
        ExtentManager.getInstance().flush();
    }

    public void onTestStart(ITestResult result) {
        System.out.println(("*** Running test method " + result.getMethod().getMethodName() +
"..."));
        ExtentTestManager.startTest(result.getMethod().getMethodName());
    }

    public void onTestSuccess(ITestResult result) {
        System.out.println("*** Executed " + result.getMethod().getMethodName() + " test
successfully...");
        ExtentTestManager.getTest().log(Status.PASS, "Test passed");
    }

    public void onTestFailure(ITestResult result) {
        log.info("*** Test execution " + result.getMethod().getMethodName() + " failed...");
        log.info((result.getMethod().getMethodName() + " failed!"));

        //ITestContext context = result.getTestContext();
        WebDriver driver = (WebDriver) result.getTestContext().getAttribute("driver");

        String targetLocation = null;

        String testClassName = result.getInstanceName().trim();
        String timeStamp = new SimpleDateFormat("yyyy.MM.dd.HH.mm.ss").format(new
java.util.Date()); // get timestamp
        String testMethodName = result.getName().toString().trim();
        String screenShotName = testMethodName + timeStamp + ".png";
        String fileSeperator = System.getProperty("file.separator");
        String reportsPath = System.getProperty("user.dir") + fileSeperator + "TestReport" +
fileSeperator
                + "screenshots";
        log.info("Screen shots reports path - " + reportsPath);
        try {
            File file = new File(reportsPath + fileSeperator + testClassName); // Set

                                // screenshots

                                // folder
            if (!file.exists()) {
                if (file.mkdirs()) {
                    log.info("Directory: " + file.getAbsolutePath() + " is created!");
                } else {
                    log.info("Failed to create directory: " + file.getAbsolutePath());
                }

            }
```

```
                              File screenshotFile = ((TakesScreenshot)
driver).getScreenshotAs(OutputType.FILE);
                              targetLocation = reportsPath + fileSeperator + testClassName + fileSeperator +
screenShotName;// define


                              // location
                              File targetFile = new File(targetLocation);
                              log.info("Screen shot file location - " + screenshotFile.getAbsolutePath());
                              log.info("Target File location - " + targetFile.getAbsolutePath());
                              FileHandler.copy(screenshotFile, targetFile);

                    } catch (FileNotFoundException e) {
                              log.info("File not found exception occurred while taking screenshot " +
e.getMessage());
                    } catch (Exception e) {
                              log.info("An exception occurred while taking screenshot " + e.getCause());
                    }

                    // attach screenshots to report
                    try {
                              ExtentTestManager.getTest().fail("Screenshot",

          MediaEntityBuilder.createScreenCaptureFromPath(targetLocation).build());
                    } catch (IOException e) {
                              log.info("An exception occured while taking screenshot " + e.getCause());
                    }
                    ExtentTestManager.getTest().log(Status.FAIL, "Test Failed");
          }

          public void onTestSkipped(ITestResult result) {
                    System.out.println("*** Test " + result.getMethod().getMethodName() + " skipped...");
                    ExtentTestManager.getTest().log(Status.SKIP, "Test Skipped");
          }

          public void onTestFailedButWithinSuccessPercentage(ITestResult result) {
                    System.out.println("*** Test failed but within percentage % " +
result.getMethod().getMethodName());
          }
}
```

src/test/java :

com.flyaway.testscripts :

TestRunner.java :

```
package com.flyaway.testscripts;

import io.cucumber.testng.AbstractTestNGCucumberTests;
import io.cucumber.testng.CucumberOptions;

@CucumberOptions(features="Features", glue="com.flyaway.testscripts")

public class TestRunner extends AbstractTestNGCucumberTests{

}
```

Tools.java :-

```
package com.flyaway.testscripts;

import org.openqa.selenium.WebDriver;

public class Tools{

    protected static WebDriver driver;

}
```

Driver.java :

```
package com.flyaway.testscripts;

import java.net.MalformedURLException;

import org.openqa.selenium.chrome.ChromeDriver;

import com.flyaway.pages.HomePage;
import com.flyaway.pages.LandingPage;
import com.flyaway.pages.PaymentGatewayPage;
import com.flyaway.pages.SignUpPage;

public class Driver extends Tools {
    protected static LandingPage landingPage;
    protected static SignUpPage signUpPage;
    protected static HomePage homePage;
    protected static PaymentGatewayPage paymentGatewayPage;

    public static void init() throws MalformedURLException {
        driver = new ChromeDriver();
        landingPage = new LandingPage(driver);
        signUpPage = new SignUpPage(driver);
        homePage = new HomePage(driver);
        paymentGatewayPage = new PaymentGatewayPage(driver);
    }

}
```

BeforeAfterHooks.java :

```
package com.flyaway.testscripts;

import java.net.MalformedURLException;

import io.cucumber.java.After;
import io.cucumber.java.Before;
import io.cucumber.java.Scenario;

public class BeforeAfterHooks extends Tools {
    @Before
    public void setUp(Scenario scenario) throws MalfomedURLException {
        Driver.init();
        driver.manage().window().maximize();
        driver.get("http://localhost:8080/FlyAway/");

    }
```

```
	@After

	public void tearDown() {
		driver.quit();
	}

}
```

SignUpFunctionalitySteps.java :

```
package com.flyaway.testscripts;

import org.testng.Assert;

import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;


public class SignUpFuctionalitySteps extends Driver{
	@Given("A user clicks on Login\\/SignUp link text in Landing page of flyaway application")
	public void a_user_clicks_on_login_sign_up_link_text_in_landing_page_of_flyaway_application() {
		try {
			Thread.sleep(2000);
		} catch (InterruptedException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
		landingPage.clickLoginSignup();
		System.out.println("User Clicked on Login/Signup button");
	}

	@When("he clicks on Not a member? Signup button")
	public void he_clicks_on_not_a_member_signup_button() {
		try {
			Thread.sleep(2000);
		} catch (InterruptedException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
		landingPage.clickSignUp();
		System.out.println("User clicked on Signup button");
	}

	@When("pass Email id as {string},Password as {string},Confirm Password as {string}, Name as
{string}, adress as {string} and city as {string}")
	public void
pass_email_id_as_password_as_confirm_password_as_name_as_adress_as_and_city_as(String string, String
string2, String string3, String string4, String string5, String string6) throws InterruptedException {
		try {
			Thread.sleep(2000);
		} catch (InterruptedException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}

		signUpPage.enteremail(string);
		signUpPage.pwd(string2);
		signUpPage.pwd2(string3);
		signUpPage.name(string4);
		signUpPage.address(string5);
		signUpPage.city(string6);
		System.out.println("User successfully entered mandatory");

	}

	@When("click on Sign Up button")
	public void click_on_sign_up_button() {
		try {
			Thread.sleep(2000);
		} catch (InterruptedException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
		signUpPage.clickSignUpbtn();
		System.out.println("User Clicked on Signup Button");

	}

	@Then("He should be able to see {string} message")
	public void he_should_be_able_to_see_message(String string) {
		try {
			Thread.sleep(2000);
		} catch (InterruptedException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
		String expectedErrMsg = "Login to continue checking flights";
		String actualErrMsg = signUpPage.getTxt();
		Assert.assertEquals(actualErrMsg, expectedErrMsg);
		System.out.println("User Successfully Sign up");
	}


}
```

FlightBookingFunctionalitySteps.java :

```
package com.flyaway.testscripts;

import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;

public class FlightBookingFunctionalitySteps extends Driver {


	@Given("A user click on Login\\/SignUp link text in Landing page of flyaway application")
	public void a_user_click_on_login_sign_up_link_text_in_landing_page_of_flyaway_application() {
		try {
			Thread.sleep(2000);
		} catch (InterruptedException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
		landingPage.clickLoginSignup();
		System.out.println("User Clicked on Login/Signup button");

	}
	@When("he pass email as {string} and Password as {string}")
	public void he_pass_email_as_and_password_as(String string, String string2) {
		signUpPage.enteremail(string);
		signUpPage.pwd(string2);
		System.out.println("User entered mail id and Password");
	}

	@When("click on login button he will be successfully navigated to Dashboard page")
	public void click_on_login_button_he_will_be_successfully_navigated_to_dashboard_page() {

		try {
			Thread.sleep(2000);
		} catch (InterruptedException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}

		signUpPage.clickLogin();
		System.out.println("User clicked on Login button ");
		System.out.println("User successfully Logged into Flyaway application");
	}

	@When("he click on Home")
	public void he_click_on_home() {

		try {
			Thread.sleep(1000);
		} catch (InterruptedException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}

		homePage.clickHome();
		try {
			Thread.sleep(2000);
		} catch (InterruptedException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
		homePage.clickSource();

		try {
			Thread.sleep(1000);
		} catch (InterruptedException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
		homePage.clickDestination();
		try {
			Thread.sleep(1000);
		} catch (InterruptedException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
		homePage.clickSubmit();
		System.out.println("User is on Home page selected source as Hyderabad and destination as 
Bangalore");

	}
	@When("click on first available flight he will navigate to confirmation page")
	public void click_on_first_available_flight_he_will_navigate_to_confirmation_page() {

		try {
			Thread.sleep(2000);
		} catch (InterruptedException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}

		homePage.clickBookFlight();
		System.out.println("User successfully entered Source and destination and booked first
available flight");

	}

	@When("if he click on Click to complete booking")
	public void if_he_click_on_click_to_complete_booking() {
		try {
			Thread.sleep(2000);
		} catch (InterruptedException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
		paymentGatewayPage.clickto();
		System.out.println("User is on Payment Gateway Page");
	}

	@Then("he should be able to see {string} message")
	public void he_should_be_able_to_see_message(String string) {
		try {
			Thread.sleep(2000);
		} catch (InterruptedException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
		String expTxt = string;
		String ActTxt = paymentGatewayPage.getcnftxt();
		System.out.println("User successfully booked flight Ticket");

	}

}
```