**Capstone Project-** Testing an Airline Booking Website

DESCRIPTION

This project requires end-to-end development of a comprehensive QA and test environment for an airline booking website. This QA and test environment should be inclusive of the following testing layers:

1. Browser-based end user testing using Selenium WebDriver
2. Unit testing for backend elements of the website using TestNG
3. API testing with Postman on AWS cloud
4. Automating the whole testing process by creating Jenkins job

The end-deliverables will be executable scripts and modules, which can be run on demand for testing the airline booking web app.

**Git Repo - https://github.com/NIRANJAN-hm/A-TE\_CAPSTONE\_PROJECT.git**

package com.flyawayrunner;

import org.junit.runner.RunWith;

import io.cucumber.junit.Cucumber;

import io.cucumber.junit.CucumberOptions;

@RunWith(Cucumber.class)

@CucumberOptions(

features = "src/test/java/Features",

glue = "StepDefn",

tags = "@tagBooking or @tagregister"

)

public class FlightRunner {

}

package StepDefn;

import java.util.List;

import org.junit.Assert;

import org.openqa.selenium.By;

import org.openqa.selenium.TimeoutException;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.Select;

import org.openqa.selenium.support.ui.WebDriverWait;

import io.cucumber.java.en.Given;

import io.cucumber.java.en.Then;

import io.cucumber.java.en.When;

public class FlyawayBookingsteps {

public static WebDriver driver;

@Given("Sam is on Registration page")

public void sam\_is\_on\_registration\_page() {

// Write code here that turns the phrase above into concrete actions

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\hniranja\\install\\Selenium\\chromedriver\\chromedriver.exe");

driver = new ChromeDriver();

driver.get("http://localhost:8080/FlyAway/signup");

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

System.out.println("Sam is on the SignUp page");

}

@When("Sam enters {string},{string}, {string}, {string}, {string} and {string}")

public void sam\_enters\_and(String emailid, String pwd, String pwd2, String name, String address, String city) {

// Write code here that turns the phrase above into concrete actions

WebElement email\_id1 = driver.findElement(By.name("email\_id"));

email\_id1.sendKeys(emailid);

WebElement password = driver.findElement(By.name("pwd"));

password.sendKeys(pwd);

WebElement password2 = driver.findElement(By.name("pwd2"));

password2.sendKeys(pwd2);

WebElement name1 = driver.findElement(By.name("name"));

name1.sendKeys(name);

WebElement address1 = driver.findElement(By.name("address"));

address1.sendKeys(address);

WebElement city1 = driver.findElement(By.name("city"));

city1.sendKeys(city);

System.out.println("Sam Enters Registrations Details");

}

@When("Sam clicks on Signup")

public void sam\_clicks\_on\_signup() {

// Write code here that turns the phrase above into concrete actions

System.out.println("Sam clicks on Signup");

WebElement signup = driver.findElement(By.tagName("button"));

signup.click();

}

@When("Sam should be able to check {string}")

public void sam\_should\_be\_able\_to\_check(String string) {

// Write code here that turns the phrase above into concrete actions

System.out.println("Sam is able to Register successfully");

List<WebElement> listLogin = driver.findElements(By.linkText("Login/Signup"));

Assert.assertTrue(listLogin.size() > 0);

}

@Then("Sam should be navigated to booking")

public void sam\_should\_be\_navigated\_to\_booking() {

// Write code here that turns the phrase above into concrete actions

System.out.println("Sam is navigated to home page");

String strTitle = driver.getTitle();

System.out.println(strTitle);

}

@Given("Sam is on the login page")

public void sam\_is\_on\_the\_login\_page() {

// Write code here that turns the phrase above into concrete actions

System.setProperty("webdriver.chrome.driver","C:\\Users\\jsahani\\install\\Selenium\\chromedriver\\chromedriver.exe");

driver = new ChromeDriver();

driver.get("http://localhost:8080/FlyAway/login");

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

System.out.println("Sam is on the login page");

}

@When("Sam enters {string} and {string}")

public void sam\_enters\_and(String un, String pwd) {

// Write code here that turns the phrase above into concrete actions

System.out.println("Sam Enters UN and PWD");

WebElement username = driver.findElement(By.name("email\_id"));

username.sendKeys(un);

WebElement password = driver.findElement(By.name("pwd"));

password.sendKeys(pwd);

}

@When("Sam clicks on Login")

public void sam\_clicks\_on\_login() {

// Write code here that turns the phrase above into concrete actions

System.out.println("Sam clicks on Login");

WebElement login = driver.findElement(By.tagName("button"));

login.click();

}

@When("Sam should be able to validate {string}")

public void sam\_should\_be\_able\_to\_validate(String string) {

// Write code here that turns the phrase above into concrete actions

System.out.println("Sam should be able to login successfully");

List<WebElement> listLogout = driver.findElements(By.linkText("Logout"));

Assert.assertTrue(listLogout.size() > 0);

}

@Then("Sam should be navigated to homepage")

public void sam\_should\_be\_navigated\_to\_homepage() {

// Write code here that turns the phrase above into concrete actions

System.out.println("Sam should navigate to home page");

String strTitle = driver.getTitle();

System.out.println(strTitle);

}

@Given("Sam has login with {string} and {string}")

public void sam\_has\_login\_with\_and(String string, String string2) {

// Write code here that turns the phrase above into concrete actions

System.setProperty("webdriver.chrome.driver","C:\\Users\\jsahani\\install\\Selenium\\chromedriver\\chromedriver.exe");

driver = new ChromeDriver();

driver.get("http://localhost:8080/FlyAway/login");

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

driver.findElement(By.name("email\_id")).sendKeys(string);

driver.findElement(By.name("pwd")).sendKeys(string2);

driver.findElement(By.tagName("button")).click();

WebDriverWait wait = new WebDriverWait(driver, 10);

try {

wait.until(ExpectedConditions.invisibilityOfElementLocated(By.linkText("Logout")));

} catch (TimeoutException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

int listLogout = driver.findElements(By.linkText("Logout")).size();

Assert.assertTrue(listLogout > 0);

}

@Given("Same has open home page")

public void same\_has\_open\_home\_page() {

// Write code here that turns the phrase above into concrete actions

driver.findElement(By.linkText("Home")).click();

}

@When("Sam selects {string} as source and {string} as destination")

public void sam\_selects\_as\_source\_and\_as\_destination(String Source, String Destination) {

// Write code here that turns the phrase above into concrete actions

Select selectsrc = new Select(driver.findElement(By.name("source")));

Select selectDest = new Select(driver.findElement(By.name("destination")));

selectsrc.selectByVisibleText(Source);

selectDest.selectByVisibleText(Destination);

}

@When("Sam clicks on submit")

public void sam\_clicks\_on\_submit() {

// Write code here that turns the phrase above into concrete actions

driver.findElement(By.tagName("button")).click();

}

@Then("Sam is able to validate for flight id {string} with msg {string}")

public void sam\_is\_able\_to\_validate\_for\_flight\_id\_with\_msg(String flightNo, String expMessage) {

// Write code here that turns the phrase above into concrete actions

List<WebElement> lstFlights = driver.findElements(By.linkText("Book Flight"));

String actMsg = "Flight Not Present";

for (WebElement flight : lstFlights) {

System.out.println(flight.getAttribute("href"));

System.out.println("Sam is able to validate for flight id");

if (flight.getAttribute("href").contains("bookflight?id=" + flightNo)) {

actMsg = "Flight Present";

break;

}

}

}

}

@tagregister

Feature: Register

I want to use this template for my feature file

@tagr

Scenario: Sam is able to register successfully

Given Sam is on Registration page

When Sam enters "<email\_id>","<password>", "<password>", "<Name>", "<Address>" and "<City>"

And Sam clicks on Signup

And Sam should be able to check "<signupstatus>"

Then Sam should be navigated to booking

Examples:

| email\_id |password | password | Name | Address | City | signupstatus |

| Niranjan2@user.com | user | user | Niranjan | Maharashtra | Mumbai | RegisterPass |

@tagl

Feature: Login

I want to use this template for test Login functionality

@tagLogin

Scenario Outline: Sam is able to login successfully

Given Sam is on the login page

When Sam enters "<username>" and "<password>"

And Sam clicks on Login

And Sam should be able to validate "<loginstatus>"

Then Sam should be navigated to homepage

Examples:

| username | password | loginstatus |

| user@user.com | user | LoginPass |

@tagBooking

Feature: Booking

I want to use this template for test Flight Booking functionality

@tagSelectFlight

Scenario: Sam is able to select a flight

Given Sam has login with "<username>" and "<password>"

And Same has open home page

When Sam selects "<Source>" as source and "<Destination>" as destination

And Sam clicks on submit

Then Sam is able to validate for flight id "<FlightNo>" with msg "<ExpResult>"

Examples:

| username | password | Source | Destination | FlightNo | ExpResult |

| user@user.com | user | Bangalore | Mumbai | 5 | Flight Present |

| user@user.com | user | Bangalore | Ahmedabad | 10 | Flight NOT Present |

**package** com.listeners;

**import** org.openqa.selenium.By;

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

**import** org.openqa.selenium.WebElement;

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

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

**public** **class** FlightBooking {

WebDriver driver;

@Test

**public** **void** Registrationpage() {

System.*setProperty*("webdriver.chrome.driver", "C:\\Users\\hniranja\\install\\Selenium\\chromedriver\\chromedriver.exe");

driver = **new** ChromeDriver();

driver.get("http://localhost:8080/FlyAway/signup");

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

}

@Test(dependsOnMethods = "Registrationpage")

**public** **void** Clicks\_signup() {

WebElement email\_id1 = driver.findElement(By.*name*("email\_id"));

email\_id1.sendKeys("Niranjan@user.com");

WebElement password = driver.findElement(By.*name*("pwd"));

password.sendKeys("user");

WebElement password2 = driver.findElement(By.*name*("pwd2"));

password2.sendKeys("user");

WebElement name1 = driver.findElement(By.*name*("name"));

name1.sendKeys("Niranjan");

WebElement address1 = driver.findElement(By.*name*("address"));

address1.sendKeys("Nerul4");

WebElement city1 = driver.findElement(By.*name*("city"));

city1.sendKeys("Navi-Mumbai");

driver.findElement(By.*xpath*("//button[text()='Signup']")).click();

}

@Test(dependsOnMethods = "Clicks\_signup")

**public** **void** homepage() {

driver.findElement(By.*xpath*("//a[contains(text(), 'Login to continue checking flights')]")).click();

}

@Test(dependsOnMethods = "homepage")

**public** **void** login\_successfully() {

driver.findElement(By.*name*("email\_id")).sendKeys("user@user.com");

driver.findElement(By.*name*("pwd")).sendKeys("user");

driver.findElement(By.*xpath*("//button[text()='Login']")).click();

}

@Test(dependsOnMethods = "login\_successfully")

**public** **void** Flight\_Booking\_Details() {

driver.findElement(By.*xpath*("//a[text()='Home']")).click();

driver.findElement(By.*xpath*("//html/body/table/tbody/tr[3]/td[6]/a")).click();

driver.findElement(By.*xpath*("//a[text()='Click to complete booking']")).click();

driver.findElement(By.*xpath*("//a[text()='See Your Bookings']")).click();

}

}

**Testng**

PASSED: Clicks\_signup

PASSED: Flight\_Booking\_Details

PASSED: login\_successfully

PASSED: Registrationpage

PASSED: homepage

===============================================

Default test

Tests run: 5, Failures: 0, Skips: 0

===============================================

===============================================

Default suite

Total tests run: 5, Passes: 5, Failures: 0, Skips: 0

===============================================

**Testing.xml**

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

<suite parallel="false" name="Suite">

<test name="Test">

<classes>

<class name="com.listeners.FlightBooking"/>

</classes>

</test> <!-- Test -->

</suite> <!-- Suite -->

**Pom.xml**

<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 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>FlyAwayCapstoneProj</groupId>

<artifactId>FlyAwayCapstoneProj</artifactId>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-java</artifactId>

<version>3.141.59</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.testng/testng -->

<dependency>

<groupId>org.testng</groupId>

<artifactId>testng</artifactId>

<version>7.5</version>

<scope>test</scope>

</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-junit -->

<dependency>

<groupId>io.cucumber</groupId>

<artifactId>cucumber-junit</artifactId>

<version>7.2.3</version>

<scope>test</scope>

</dependency>

<!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-core -->

<dependency>

<groupId>io.cucumber</groupId>

<artifactId>cucumber-core</artifactId>

<version>7.2.3</version>

</dependency>

</dependencies>

</project>

**Jenkins**

Started by user admin

Running as SYSTEM

Building in workspace C:\Users\jsahani\.jenkins\workspace\RunPostman

[RunPostman] $ cmd /c call C:\Users\hniranja\AppData\Local\Temp\jenkins7378680367986986415.bat

C:\Users\hniranja\.jenkins\workspace\RunPostman>cd C:\Users\jsahani\Pracs Automation\Postman\Collections

C:\Users\hniranja\Pracs Automation\Postman\Collections>newman run Flyaway.postman\_collection.json

newman

Flyaway

→ Signup

POST http://localhost:8080/FlyAway/signupaction?email\_id=user4@user.com&pwd=user&pwd2=user&name=user1&address=user&city=user [200 OK, 1.4kB, 59ms]

→ Login

POST http://localhost:8080/FlyAway/loginaction?email\_id=user4@user.com&pwd=user [200 OK, 591B, 27ms]

→ SelectFlight

GET http://localhost:8080/FlyAway/home?source=1&destination=2

[200 OK, 2.43kB, 28ms]

→ BookFlight

GET http://localhost:8080/FlyAway/bookflight?id=5 [200 OK, 984B, 33ms]

→ ConfirmFlight

GET http://localhost:8080/FlyAway/completepurchase [200 OK, 805B, 28ms]

┌─────────────────────────┬───────────────────┬──────────────────┐

│ │ executed │ failed │

├─────────────────────────┼───────────────────┼──────────────────┤

│ iterations │ 1 │ 0 │

├─────────────────────────┼───────────────────┼──────────────────┤

│ requests │ 5 │ 0 │

├─────────────────────────┼───────────────────┼──────────────────┤

│ test-scripts │ 5 │ 0 │

├─────────────────────────┼───────────────────┼──────────────────┤

│ prerequest-scripts │ 5 │ 0 │

├─────────────────────────┼───────────────────┼──────────────────┤

│ assertions │ 0 │ 0 │

├─────────────────────────┴───────────────────┴──────────────────┤

│ total run duration: 573ms │

├────────────────────────────────────────────────────────────────┤

│ total data received: 5.18kB (approx) │

├────────────────────────────────────────────────────────────────┤

│ average response time: 35ms [min: 27ms, max: 59ms, s.d.: 12ms] │

└────────────────────────────────────────────────────────────────┘

Finished: SUCCESS

**SCREENSHOTS**

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated