**Identify Courses**

**Identify Courses Package:**

**Homepage.Java**

**package** IdentifyCourses;

**import** java.time.Duration;

**import** java.util.List;

**import** org.openqa.selenium.JavascriptExecutor;

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

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

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

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

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

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

**public** **class** HomePage{

WebDriver driver;

**public** HomePage(WebDriver driver) {

**this**.driver = driver;

PageFactory.*initElements*(driver,**this**);

}

//locators

@FindBy(xpath="//input[@type='text']")

WebElement searchbutton;

@FindBy(xpath="//button[@class='nostyle search-button']//div")

WebElement clicksearch;

@FindBy(xpath="//input[@id=\"cds-react-aria-17\"]")

WebElement lang;

@FindBy(xpath="//span[text()='Beginner']")

WebElement level;

@FindBy(xpath="//h3[@class='cds-CommonCard-title css-6ecy9b']")

List<WebElement> c\_title;

@FindBy(xpath="//p[@class='css-2xargn']")

List<WebElement> c\_ratings;

@FindBy(xpath="//p[contains(text(),'Months')]")

List<WebElement> c\_detail;

@FindBy(xpath="//div[@data-testid='search-filter-group-Language']")//-----scrolling to language

WebElement language;

@FindBy(xpath="//button[@data-track-component='expand\_filter\_items\_button\_language']")//-----languagebuttonselect

WebElement LanguageButton;

@FindBy(xpath="//div[@aria-label='Select Language options']//div[contains(@class,'css-zweepb')]")//-----when language window appears

List<WebElement> languagewindow;

@FindBy(xpath="//button[@class='cds-149 cds-button-disableElevation cds-button-ghost css-1s96oj']")

WebElement langclick;

@FindBy(xpath="//div[@data-testid='search-filter-group-Language']//div[contains(@class,'css-nsxeb5')]")

List<WebElement> langlist;

@FindBy(xpath="//div[@data-testid='search-filter-group-Level']")//----locate level button

WebElement levelbutton;

@FindBy(xpath="//div[@data-testid='search-filter-group-Level']//div[contains(@class,'css-nsxeb5')]")//------list of level

List<WebElement> levellist;

//action methods

**public** **void** searchbutton()//------searching for course

{

searchbutton.click();

searchbutton.sendKeys("Web Development Courses");

}

**public** **void** clicksearch()//-------clicking the button

{

clicksearch.click();

}

**public** **void** languageselect()//------selecting language

{

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("arguments[0].click()",lang);

}

**public** **void** levelselect()//---------level select

{

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("arguments[0].click()",level);

}

**public** List<WebElement> CourseTitle() **throws** InterruptedException//------extracting first two course title

{

Thread.*sleep*(5000);

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("scrollBy(0,-400)");

**int** count = 0;

List<WebElement> courseTitles = c\_title;

System.***out***.println("Title of First 2 Courses:");

**for**(WebElement el : courseTitles) {

**if**(count>=2) {

**break**;

}

System.***out***.println(el.getText());

count++;

}

**return** courseTitles;

}

**public** List<WebElement> CourseRating()//---------first two course ratings

{

**int** c = 0;

List<WebElement> courseRatings = c\_ratings;

System.***out***.println("Ratings of First 2 Courses:");

**for**(WebElement ele : courseRatings) {

**if**(c>=2) {

**break**;

}

System.***out***.println(ele.getText()+ " Stars");

c++;

}

**return** courseRatings;

}

**public** List<WebElement> courseDetail()//-------extracting first two course detail

{

**int** d = 0;

List<WebElement> details = c\_detail;

System.***out***.println("Details of First 2 Courses:");

**for**(WebElement ele : details) {

**if**(d>=2) {

**break**;

}

System.***out***.println(ele.getText());

d++;

}

**return** details;

}

**public** **void** Alllanguages() **throws** InterruptedException//--------printing all available languages

{

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("arguments[0].scrollIntoView()", language);

Thread.*sleep*(5000);

WebDriverWait wait = **new** WebDriverWait(driver, Duration.*ofSeconds*(20));

wait.until(ExpectedConditions.*visibilityOf*(LanguageButton));

LanguageButton.click();

System.***out***.println("Languages Available :");

**try**{

List<WebElement> languagesList = languagewindow;

**for** (WebElement e : languagesList) {

System.***out***.println(e.getText());

}

langclick.click();

}

**catch**(Exception e) {

**try**

{

List<WebElement> languagesList = langlist;

**for** (WebElement el : languagesList) {

System.***out***.println(el.getText());

}

}

**catch**(Exception el) {

}

}

}

**public** List<WebElement> Alllevels() **throws** InterruptedException //------printing all levals

{

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("arguments[0].scrollIntoView()", levelbutton);

Thread.*sleep*(3000);

System.***out***.println("List Available :");

List<WebElement> levelList = levellist;

**for** (WebElement e : levelList) {

System.***out***.println(e.getText());

}

**return** levelList;

}

}

**Institution.Java**

**package** IdentifyCourses;

**import** org.openqa.selenium.JavascriptExecutor;

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

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

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

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

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

**public** **class** Institution {

WebDriver driver;

//constructor

**public** Institution(WebDriver driver)

{

**this**.driver = driver;

PageFactory.*initElements*(driver,**this**);

}

//Locators

@FindBy(linkText="For Enterprise")

WebElement enterprise;

@FindBy(linkText ="For Campus")

WebElement campus;

@FindBy(xpath="//div[@class=\"useFloatingLabels css-esaiwz\"]")

WebElement locateform;

@FindBy(id="FirstName")

WebElement fname;

@FindBy(id="LastName")

WebElement lname;

@FindBy(id="Email")

WebElement mail;

@FindBy(id="Phone")

WebElement ph;

@FindBy(id="Institution\_Type\_\_c")

WebElement insttype;

@FindBy(id="Company")

WebElement comp;

@FindBy(id="Title")

WebElement title;

@FindBy(id="Department")

WebElement Dept;

@FindBy(id="What\_the\_lead\_asked\_for\_on\_the\_website\_\_c")

WebElement Wdescribe;

@FindBy(id="Self\_reported\_employees\_to\_buy\_for\_\_c")

WebElement learn;

@FindBy(id="Country")

WebElement Count;

@FindBy(id="State")

WebElement state;

@FindBy(xpath="//button[@class=\"mktoButton\"]")

WebElement submitform;

@FindBy(id="ValidMsgEmail")

WebElement errormsgs;

//Action methods

**public** **void** ClickEnterprise()

{

JavascriptExecutor js = (JavascriptExecutor)driver;

js.executeScript("window.scrollBy(0,document.body.scrollHeight)");

js.executeScript("arguments[0].click()",enterprise);

}

**public** **void** campusclick()

{

// JavascriptExecutor js = (JavascriptExecutor)driver;

// js.executeScript("arguments[0].click()",campus);

campus.click();

}

**public** **void** locateform()

{

JavascriptExecutor js = (JavascriptExecutor)driver;

js.executeScript("arguments[0].scrollIntoView()",locateform);

}

**public** **void** formdetails(String firname,String lasname,String Email,String Phone,String instutype,String companyname,String Title,String Department,String Whdescribe,String Learn,String Countries,String State)

{

fname.sendKeys(firname);

lname.sendKeys(lasname);

mail.sendKeys(Email);

ph.sendKeys(Phone);

Select ins = **new** Select(insttype);

ins.selectByVisibleText(instutype);

comp.sendKeys(companyname);

Select role = **new** Select(title);

role.selectByVisibleText(Title);

Select dept = **new** Select(Dept);

dept.selectByVisibleText(Department);

Select whichdescribe = **new** Select(Wdescribe);

whichdescribe.selectByVisibleText(Whdescribe);

Select learners = **new** Select(learn);

learners.selectByVisibleText(Learn);

Select Country = **new** Select(Count);

Country.selectByVisibleText(Countries);

Select states = **new** Select(state);

states.selectByVisibleText(State);

}

**public** **void** submitForm()

{

submitform.click();

}

**public** **void** errormsg()

{

JavascriptExecutor js = (JavascriptExecutor)driver;

js.executeScript("scrollBy(0,-500)");

String msg = errormsgs.getText();

System.***out***.println("Error message is:" +msg);

}

}

**Screenshot.Java**

**package** IdentifyCourses;

**import** java.io.File;

**import** java.io.IOException;

**import** org.openqa.selenium.OutputType;

**import** org.openqa.selenium.TakesScreenshot;

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

**import** org.openqa.selenium.io.FileHandler;

**public** **class** ScreenShot {

WebDriver driver;

**public** ScreenShot(WebDriver driver)

{

**this**.driver = driver;

}

**public** **void** takeScreenshot(String screenshotName) **throws** IOException {

TakesScreenshot ts = (TakesScreenshot) driver;

File s = ts.getScreenshotAs(OutputType.***FILE***);

File trg = **new** File("C:\\Users\\2317305\\MinorProject\\HackathonProject\\src\\test\\resources\\Screenshots\\" + screenshotName + ".png");

FileHandler.*copy*(s, trg);

}

}

**Test Package:**

**ExtentReport.Java**

**package** Test;

**import** org.testng.ITestContext;

**import** org.testng.ITestListener;

**import** org.testng.ITestResult;

**import** com.aventstack.extentreports.ExtentReports;

**import** com.aventstack.extentreports.ExtentTest;

**import** com.aventstack.extentreports.Status;

**import** com.aventstack.extentreports.reporter.ExtentSparkReporter;

**import** com.aventstack.extentreports.reporter.configuration.Theme;

**public** **class** ExtentReport **implements** ITestListener

{

**public** ExtentSparkReporter sparkReporter; // UI of the report

**public** ExtentReports extent; //populate common info on the report

**public** ExtentTest test; // creating test case entries in the report and update status of the test methods

**public** **void** onStart(ITestContext context) {

sparkReporter=**new** ExtentSparkReporter("C:\\Users\\2317305\\MinorProject\\HackathonProject\\src\\test\\resources\\"+ "reports/myReport.html");//specify location of the report

sparkReporter.config().setDocumentTitle("Automation Report"); // TiTle of report

sparkReporter.config().setReportName("Functional Testing"); // name of the report

sparkReporter.config().setTheme(Theme.***DARK***);

extent=**new** ExtentReports();

extent.attachReporter(sparkReporter);

extent.setSystemInfo("Tester Name","Agin");

extent.setSystemInfo("Browser name","chrome,edge");

}

**public** **void** onTestSuccess(ITestResult result) {

test = extent.createTest(result.getName()); // create a new entry in the report

test.log(Status.***PASS***, "Test case PASSED is:" + result.getName()); // update status p/f/s

test.addScreenCaptureFromPath("C:\\Users\\2317287\\MinorProject\\HackathonProject\\src\\test\\resources\\Screenshots\\"+result.getName()+".png");// ---for screenshot

}

**public** **void** onTestFailure(ITestResult result) {

test = extent.createTest(result.getName());

test.log(Status.***FAIL***, "Test case FAILED is:" + result.getName());

test.log(Status.***FAIL***, "Test Case FAILED cause is: " + result.getThrowable());

}

**public** **void** onTestSkipped(ITestResult result) {

test = extent.createTest(result.getName());

test.log(Status.***SKIP***, "Test case SKIPPED is:" + result.getName());

}

**public** **void** onFinish(ITestContext context) {

extent.flush();

}

}

**Testng.Java**

**package** Test;

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

**import** com.beust.jcommander.Parameter;

**import** IdentifyCourses.HomePage;

**import** IdentifyCourses.Institution;

**import** IdentifyCourses.ScreenShot;

**import** org.testng.annotations.AfterClass;

**import** org.testng.annotations.BeforeClass;

**import** java.io.IOException;

**import** java.time.Duration;

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

**import** org.openqa.selenium.JavascriptExecutor;

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

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

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

**import** org.openqa.selenium.edge.EdgeDriver;

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

**import** org.testng.annotations.DataProvider;

**import** org.testng.annotations.Optional;

**import** org.testng.annotations.Parameters;

**public** **class** Testng {

WebDriver driver;

HomePage ho;

Institution inst;

ScreenShot ss;

@BeforeClass

@Parameters({"browser"})

**void** URL(String br) {

**if**(br.equals("chrome"))

{

driver = **new** ChromeDriver();

}

**else** **if**(br.equals("edge"))

{

driver = **new** EdgeDriver();

}

driver.manage().timeouts().implicitlyWait(Duration.*ofSeconds*(10));

driver.get("https://www.coursera.org/");

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

ss = **new** ScreenShot(driver);

}

//step1 --------Search for web development courses for Beginners level & English Language

@Test(priority=1)

**void** HomePagesearch() **throws** InterruptedException, IOException

{

ho = **new** HomePage(driver);

ho.searchbutton();//-------clicking on search button and write web development course

ho.clicksearch();//-------clicking and searching above course

ss.takeScreenshot("HomePageSearch");

ho.languageselect();//------ selecting the language

ho.levelselect();//-------selecting the level

}

//Step 1 ----------extract the course names,details & rating for first 2 courses

@Test(priority=2 ,dependsOnMethods ={"HomePagesearch"})

**void** searchResults() **throws** InterruptedException, IOException {

ss.takeScreenshot("SearchResults");

ho.CourseTitle();//---------first two course title

ho.CourseRating();//---------first two course rating

ho.courseDetail();//---------first two course details

}

@Test(priority=3)

**void** deSelect()//--------disselecting language and levels

{

ho.languageselect();

ho.levelselect();

}

//STep 2 --------- Extract all the languages and different levels with its total count & display them

@Test(priority =4 ,dependsOnMethods ={"HomePagesearch" ,"searchResults"} )

**void** languagesandLevels() **throws** InterruptedException, IOException {

ho.Alllanguages();//-----printing all languages

Thread.*sleep*(3000);

ho.Alllevels();//--------printing all levels

driver.navigate().to("https://www.coursera.org/");//-----navigating back to home page

ss.takeScreenshot("LanguageandLevels");

Thread.*sleep*(3000);

}

@Test(priority =5)

**void** Navigate() **throws** InterruptedException, IOException

{

inst = **new** Institution(driver);

inst.ClickEnterprise();

inst.campusclick();

inst.locateform();

ss.takeScreenshot("Navigate");

}

//Step 3 -----Filling form and capturing error msg

@Test(priority =6,dataProvider= "formdetails")

**void** form(String f\_name,String l\_name,String email,String phone,String institutiontype, String institutionname,String Jobrole,String department,String whichd,String learn,String country,String state) **throws** InterruptedException, IOException

{

inst.formdetails(f\_name, l\_name, email, phone, institutiontype, institutionname, Jobrole, department, whichd,learn,country,state);

inst.submitForm();

Thread.*sleep*(3000);

inst.errormsg();

ss.takeScreenshot("form");

Thread.*sleep*(2000);

}

@AfterClass

**void** close()

{

driver.quit();

}

@DataProvider(name ="formdetails")

String [][] formdetails()

{

String data[][] = {

{"Akash","kumar",

"aaa123s@gmail.com","8989788900",

"2 Year College","Ajki",

"Vice-President/Vice-Provost","Career Services",

"Meet with Coursera Sales Team","5-25","India","Gujarat"}

};

**return** data;

}

}

**Testing.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 =*"Test.ExtentReport"*/>

</listeners>

<test thread-count=*"2"* name=*"ChromeTest"*>

<parameter name=*"browser"* value =*"chrome"*/>

<classes>

<class name=*"Test.Testng"*/>

</classes>

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

<test thread-count=*"2"* name=*"EdgeTest"*>

<parameter name=*"browser"* value =*"edge"*/>

<classes>

<class name=*"Test.Testng"*/>

</classes>

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

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

**Output:**

**SourceFolder:**

**A screenshot of a computer program

Description automatically generated**

**CourseSelect:**

**A screenshot of a web page

Description automatically generated**

**HomePage:**

**A person sitting at a desk using a computer

Description automatically generated**

**From:**

A screenshot of a computer

Description automatically generated

**Error:**

**A screenshot of a computer

Description automatically generated**

**Console :**

**A screenshot of a computer

Description automatically generated**