**SofiaSelenium-@Test-AdvancedMoudeInteractions1**

public class AdvancedMoudeInteractions {  
 WebDriver driver;  
  
 @Before  
 public void setUp() {  
 WebDriverManager.*chromedriver*().setup();  
 driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(5, TimeUnit.*SECONDS*);  
 }  
  
 @After  
 public void tearDown() {  
 driver.close();  
 driver.quit();  
 }  
  
 @Test  
 public void test1() {  
 driver.navigate().to("https://demoqa.com/buttons");  
 //first find the elements  
 WebElement doubleClickButton = driver.findElement(By.*id*("doubleClickBtn"));  
 WebElement rightClickButton = driver.findElement(By.*id*("rightClickBtn"));  
 //create an obj of actions class  
 Actions actions = new Actions(driver);  
 //action to happen on element we always finish with .perform()->make the action happens  
 actions.doubleClick(doubleClickButton).perform();  
  
 Assert.*assertTrue*("double click message isn't displayed", driver.findElement(By.*id*("doubleClickMessage")).isDisplayed());  
  
 //now right click  
 actions.contextClick(rightClickButton).perform();  
  
 Assert.*assertTrue*("right click message isn't displayed", driver.findElement(By.*id*("rightClickMessage")).isDisplayed());  
  
  
 }  
  
 @Test  
 public void test2() throws InterruptedException {  
 driver.navigate().to("https://demoqa.com/tool-tips");  
 WebElement hoverButoon = driver.findElement(By.*id*("toolTipButton"));  
 Actions actions = new Actions(driver);  
  
 actions.moveToElement(hoverButoon).perform();  
 WebDriverWait wait = new WebDriverWait(driver, 3);  
 boolean attributeAppeared = wait.until(ExpectedConditions.*attributeToBe*(hoverButoon, "aria-describedby", "buttonToolTip"));  
 String attribute = hoverButoon.getAttribute("aria-describedby");//"buttonToolTip"  
  
 Assert.*assertEquals*("buttonToolTip", attribute);  
 actions.moveToElement(driver.findElement(By.*xpath*("//div[contains(text(),'Book Store Application')]"))).perform();  
 Thread.*sleep*(3000);  
 }  
  
 @Test  
 public void test3() throws InterruptedException {  
 driver.navigate().to("https://www.etsy.com/");  
 //i need to scroll to subscribe input field type email hit subscribe  
 WebElement subscribeInput = driver.findElement(By.*id*("email-list-signup-email-input"));  
 Actions actions = new Actions(driver);  
 actions.moveToElement(subscribeInput).click().sendKeys("devxschool@gmail.com" + Keys.*ENTER*).perform();  
 Thread.*sleep*(3000);  
 }

hi Baha,

Thanks for your cv, like the highlighted keywords part.

1. Remove "\_" from your summary--> example:  "• \_Experience" should be "•  Experience"

2. the top: name, email etc- change font. name should be bold and larger, than the rest.  
3. I'd cut the professional summary part--> leaving the most important- make it shorter to fit into 1 page

4. summary of tech skills: add more. Sauce Labs- yes add it. Selenium Grid- add it if you know it.

5. Professional experience:  
-companies should be reversed: latest first--> Cnh, then Northern and last CVS  
-functional and automation engineer is not a title, put QA Analyst in there.   
- shorten cv to 3 pages (4 is too much kind of)

-for CVS responsibilities should be as QA analyst, probably Waterfall environment

6. Consistency-->  
-April 16- may 18--> May should be with capital M (look at Capital letters in CV. everything should start with capital)

7. Font --> revisit font for cv, use different (in telegram there is a guidance for fonts)

Overall good, fix those above and send me back to check all statements. Did not check it yet.   
  
Thanks,

Adina.

**SofiaSelenium-@Test-AdvancedMoudeInteractions2**

@Test  
 public void test4() throws InterruptedException {  
 driver.navigate().to("https://demoqa.com/slider");  
 WebElement slider = driver.findElement(By.*xpath*("//input[@type='range']"));  
 Actions actions = new Actions(driver);  
 actions.clickAndHold(slider).moveByOffset(20, 0).release().perform();  
 Thread.*sleep*(3000);  
 WebElement sliderValue = driver.findElement(By.*id*("sliderValue"));  
 Assert.*assertTrue*(Integer.*parseInt*(sliderValue.getAttribute("value")) > 25);  
 }  
  
 @Test  
 public void test5() throws InterruptedException {  
 driver.navigate().to("https://demoqa.com/droppable");  
 WebElement source = driver.findElement(By.*id*("draggable"));  
 WebElement target = driver.findElement(By.*id*("draggable"));  
  
 Actions actions = new Actions(driver);  
 actions.dragAndDrop(source, target).perform();  
 Thread.*sleep*(3000);  
 Assert.*assertTrue*(target.getText().contains("Dropped!"));  
  
 }  
  
 @Test  
 public void test6() throws InterruptedException {  
  
 driver.navigate().to("https://demoqa.com/droppable");  
 driver.findElement(By.*id*("droppableExample-tab-preventPropogation")).click();  
 WebElement source = driver.findElement(By.*xpath*("//div[@id='dragBox']"));  
 WebElement target = driver.findElement(By.*id*("notGreedyInnerDropBox"));  
 Actions actions = new Actions(driver);  
 actions.dragAndDrop(source, target).perform();  
 Thread.*sleep*(4000);  
  
 Assert.*assertTrue*(target.getText().contains("Dropped!"));  
 Assert.*assertTrue*(driver.findElement(By.*xpath*("//div[@id='notGreedyDropBox']")).getText().contains("Dropped!"));  
  
 }  
  
 @Test  
 public void test7() {  
 //1.Go to https://demoqa.com/progress-bar  
 // 2.Click start  
 // 3.Wait until the value reaches 100–Use EXPLICIT WAIT  
 // 4.Verify the button has changed to Reset  
 // 5.Click Reset  
 // 6.Verify the progress bar has reset to 0;  
 driver.navigate().to("https://demoqa.com/progress-bar");  
 driver.findElement(By.*id*("startStopButton")).click();  
 WebElement elemen = driver.findElement(By.*xpath*("//div[@role='progressbar']"));  
 WebDriverWait wait = new WebDriverWait(driver, 14);  
 wait.until(ExpectedConditions.*attributeToBe*(elemen, "aria-valuenow", "100"));  
 // wait.until(ExpectedConditions.presenceOfElementLocated(By.id("resetButton")));  
 driver.findElement(By.*id*("resetButton")).click();  
 // WebElement elemen = driver.findElement(By.xpath("//div[@role='progressbar']"));  
  
 Assert.*assertTrue*(Integer.*parseInt*(elemen.getAttribute("aria-valuenow")) == 0);  
 }

**SofiaSelenium-@Test-AdvancedMoudeInteractions3**

@Test  
 public void test8() {  
 //1. Go to https://demoqa.com/menu#  
 // 2. Hover over Main Item 2  
 // 3. Hover over SUBSUB LIST  
 // 4. Click on Sub Sub item  
 driver.navigate().to("https://demoqa.com/menu#");  
 WebElement mainItem = driver.findElement(By.*xpath*("//a[contains(text(),'Main Item 2')]"));  
 Actions actions = new Actions(driver);  
 actions.moveToElement(mainItem).perform();  
 actions.moveToElement(driver.findElement(By.*xpath*("//a[contains(text(),'SUB SUB LIST »')]"))).perform();  
 driver.findElement(By.*xpath*("//a[contains(text(),'Sub Sub Item 1')]")).click();  
  
 }  
  
 @Test  
 public void test9() throws InterruptedException {  
 //1.Go to https://www.seleniumeasy.com/test/drag-and-drop-demo.html  
 // 2.Drag and drop 1st and 3rd Draggables to Drop Here object  
 // 3.Verify your Dropped item list contains correct values  
 driver.navigate().to("https://www.seleniumeasy.com/test/drag-and-drop-demo.html");  
 WebElement draggable1 = driver.findElement(By.*xpath*("//div[@id='todrag']//span[contains(text(),'Draggable 1')]"));  
 WebElement draggable3 = driver.findElement(By.*xpath*("//div[@id='todrag']//span[contains(text(),'Draggable 3')]"));  
 WebElement target = driver.findElement(By.*xpath*("//div[@id='mydropzone']"));  
 Actions actions = new Actions(driver);  
 actions.moveToElement(target);  
 actions.moveToElement(target,10, 0);  
  
 actions.dragAndDrop(draggable1, target).build().perform();  
 actions.moveToElement(target,10, 0);  
 actions.moveToElement(target);  
  
 Thread.*sleep*(4000);  
 actions.moveToElement(target);  
 actions.moveToElement(target,10, 0);  
  
 actions.dragAndDrop(draggable3, target).build().perform();  
 actions.moveToElement(target,10, 0);  
  
 actions.moveToElement(target);  
  
 Thread.*sleep*(4000);  
 WebElement droppedListBox = driver.findElement(By.*xpath*("//div[@id='droppedlist']"));  
// actions.clickAndHold(draggable1);  
// actions.dragAndDropBy(draggable1, 10,0).release().perform();  
// Thread.sleep(4000);  
// actions.clickAndHold(draggable3);  
// actions.dragAndDropBy(draggable3, 10,0).release().perform();  
// Thread.sleep(4000);  
 Assert.*assertTrue*(droppedListBox.getAttribute("style").equals("visibility: visible;"));  
 //Assert.assertTrue(droppedListBox.getAttribute("style").equals("style"));  
 ////div[@id='droppedlist']//\*[contains(text(),'Draggable 1')]  
 // Assert.assertTrue(driver.findElement(By.xpath("//div[@id='droppedlist']//\*[contains(text(),'Draggable 1')]")).isDisplayed());  
 // Assert.assertTrue(driver.findElement(By.xpath("//div[@id='droppedlist']//\*[contains(text(),'Draggable 3')]")).isDisplayed());  
 //Assert.assertTrue(driver.findElement(By.xpath("//div[@style='visibility: visible;']")).isDisplayed());  
 //Assert.assertEquals();  
 }

**SofiaSelenium-@Test-AdvancedMoudeInteractions4**

@Test  
 public void test10() throws InterruptedException{  
 driver.navigate().to("https://www.globalsqa.com/demo-site/draganddrop/");  
 WebElement source1 = driver.findElement(By.*xpath*("//li[@class='ui-widget-content ui-corner-tr ui-draggable ui-draggable-handle']//img[@alt='The peaks of High Tatras']"));  
 WebElement source2 = driver.findElement(By.*xpath*("//li[@class='ui-widget-content ui-corner-tr ui-draggable ui-draggable-handle']//img[@alt='The chalet at the Green mountain lake']"));  
 WebElement target = driver.findElement(By.*xpath*("//div[@id='trash']"));  
 Actions actions = new Actions(driver);  
 actions.dragAndDrop(source1,target);  
 Thread.*sleep*(4000);  
 actions.dragAndDrop(source2,target);  
 Thread.*sleep*(4000);  
  
 List<WebElement>listWithItems = driver.findElements(By.*xpath*("//div[@id='trash']"));  
 for(WebElement element : listWithItems){  
 Assert.*assertTrue*(element.getText().contains("The peaks of High Tatras") && element.getText().contains("The chalet at the Green mountain lake"));  
 }  
  
  
 }

public class PracticeWaits {  
 WebDriver driver;  
  
 @Before  
 public void setUp(){  
 WebDriverManager.*chromedriver*().setup();  
 driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(3, TimeUnit.*SECONDS*);  
  
}  
  
 @Test  
 public void test1() {  
  
 driver.get("https://www.etsy.com/");  
 driver.findElement(By.*xpath*("//button[@class='wt-btn wt-btn--small wt-btn--transparent wt-mr-xs-1 inline-overlay-trigger signin-header-action select-signin']")).click();  
 WebDriverWait wait = new WebDriverWait(driver,4);  
 wait.until(ExpectedConditions.*presenceOfElementLocated*(By.*xpath*("//input[@id='join\_neu\_email\_field']")));  
 WebElement emailInputField = driver.findElement(By.*xpath*("//input[@id='join\_neu\_email\_field']"));  
 emailInputField.sendKeys("someemail");  
 // Assert.assertTrue("Email input is not displayed", emailInputField.isDisplayed());  
 }  
  
 @Test  
 public void test2() {  
 driver.get("https://demoqa.com/dynamic-properties");  
 WebDriverWait wait = new WebDriverWait(driver,10);  
 wait.until(ExpectedConditions.*presenceOfElementLocated*(By.*xpath*("//button[@id='visibleAfter']")));  
 }  
  
 @Test  
 public void fileUpload(){  
 driver.get("https://demoqa.com/upload-download");  
 driver.findElement(By.*xpath*("//input[@id='uploadFile']")).sendKeys("/Users/mac/Desktop/DevX-School-Logo copy.png");  
  
  
  
 }  
 @Test  
 public void sendToGoogleImages(){  
 driver.get("https://images.google.com/");  
 driver.findElement(By.*xpath*("//div[@aria-label='Search by image']")).click();  
 WebDriverWait wait = new WebDriverWait(driver,7);  
 wait.until(ExpectedConditions.*presenceOfElementLocated*(By.*xpath*("//input[@id='awyMjb']"))).sendKeys("/Users/mac/Desktop/DevX-School-Logo copy.png");  
 }  
 @After  
 public void tearDown(){  
 //driver.close();

public class PageNavigation {  
 WebDriver driver;  
  
 @Before  
 public void setUp() {  
 WebDriverManager.*chromedriver*().setup();  
 driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(5, TimeUnit.*SECONDS*);  
  
 }  
  
 @Test  
 public void test1() throws InterruptedException{  
 //does exactly the same as driver.get() - no difference between them  
 //driver.navigate().to("http://www.practiceselenium.com/");  
 driver.get("http://www.practiceselenium.com/welcome.html");  
 WebElement letsTalkTeaLink = driver.findElement(By.*xpath*("//li[@style='width: '][4]"));  
 letsTalkTeaLink.click();  
 //element.getAttribute("attribute name") -> return the value of that attribute  
 letsTalkTeaLink = driver.findElement(By.*xpath*("//li[@style='width: '][4]"));  
 String attribute = letsTalkTeaLink.getAttribute("class");  
 Assert.*assertTrue*("Let's talk tea must be active but it is not",attribute.equals("active"));  
 driver.navigate().back();  
 WebElement welcomeLink = driver.findElement(By.*xpath*("//li[@style='width: '][1]"));  
 Assert.*assertTrue*("Welcome link must be active but it is not", welcomeLink.getAttribute("class").equals("active"));  
 driver.navigate().forward();  
 letsTalkTeaLink = driver.findElement(By.*xpath*("//li[@style='width: '][4]"));  
 Assert.*assertTrue*("Let's talk tea must be active but it is not",letsTalkTeaLink.getAttribute("class").equals("active"));  
 WebElement nameInputField = driver.findElement(By.*xpath*("//input[@name='name']"));  
 Assert.*assertTrue*(nameInputField.isDisplayed());  
 driver.navigate().refresh();  
 //StaleElementException - element is old, not fresh, you have to find it again in order to interact with it  
 try{  
 nameInputField.sendKeys("My name is here");  
 }catch(StaleElementReferenceException e){  
 nameInputField = driver.findElement(By.*xpath*("//input[@name='name']"));  
 nameInputField.sendKeys("My name is here");  
 }  
 Thread.*sleep*(3000);  
 }  
  
 @After  
 public void tearDown() {  
 driver.close();  
 driver.quit();  
 }

public class DropDowns {  
 WebDriver driver;  
  
 @Test  
 public void test1() {  
  
 driver.navigate().to("https://www.devxschool.com/");  
  
 WebElement enrollNowButton = driver.findElement(By.*linkText*("Enroll Now"));  
 enrollNowButton.click();  
 WebElement dropDown = driver.findElement(By.*xpath*("//select[@id='form-field-ads']"));  
 Select selectElement = new Select(dropDown);  
 List<WebElement> selectedOptions = selectElement.getAllSelectedOptions();  
 for (WebElement option : selectedOptions) {  
 System.*out*.println(option.getText());  
 }  
 Assert.*assertTrue*(selectedOptions.size() == 1 &&  
 selectedOptions.get(0).getText().equals("From a friend"));  
  
 //selectElement.selectByVisibleText("Instagram");  
 //selectElement.selectByIndex(2);  
 selectElement.selectByValue("Instagram");  
  
 selectedOptions = selectElement.getAllSelectedOptions();  
 Assert.*assertTrue*(selectedOptions.size() == 1 &&  
 selectedOptions.get(0).getText().equals("Instagram"));  
 }  
  
  
 @Test  
 public void test2() {  
 driver.navigate().to("https://www.expedia.com/");  
 //find cruises button  
 WebElement cruisesButton = driver.findElement(By.*id*("tab-cruise-tab-hp"));  
 //click on it  
 cruisesButton.click();  
 //find select element  
 WebElement selectElement = driver.findElement(By.*id*("cruise-destination-hp-cruise"));  
 //create object of Select class  
 Select selectDestination = new Select(selectElement);  
 //select Alaska by value -->> value="alaska"  
 selectDestination.selectByValue("alaska");  
 //verify Alaska is selected  
 Assert.*assertTrue*(selectDestination.getAllSelectedOptions().get(0).getText().equals("Alaska"));  
 //select Africa by visible text -->> visible text is Africa  
 selectDestination.selectByVisibleText("Africa");  
 Assert.*assertTrue*(selectDestination.getAllSelectedOptions().get(0).getText().equals("Africa"));  
 //select Mexico by index -->> index of Mexico is 3  
 selectDestination.selectByIndex(3);  
 Assert.*assertTrue*(selectDestination.getAllSelectedOptions().get(0).getText().equals("Mexico"));  
 //print out all possible options  
 for (WebElement option : selectDestination.getOptions())  
 System.*out*.println(option.getText());  
 }

**SofiaSelenium-@Test- DropDowns**

@Test  
 public void test3() {  
 driver.navigate().to("https://www.jquery-az.com/boots/demo.php?ex=63.0\_2");  
 //find select element  
 WebElement selectElement = driver.findElement(By.*id*("option-droup-demo"));  
 //create an object of Select class  
 Select dropdown = new Select(selectElement);  
 //dropdown.deselectAll();  
 //select Java  
 dropdown.selectByVisibleText("Java");  
 dropdown.selectByVisibleText("Oracle");  
 //deselect default values  
 dropdown.deselectByVisibleText("HTML");  
 dropdown.deselectByVisibleText("CSS");  
 List<WebElement> mySelectedOptions = dropdown.getAllSelectedOptions();  
 for (WebElement selectedOption : mySelectedOptions) {  
 Assert.*assertTrue*(selectedOption.getText().equals("Java") ||  
 selectedOption.getText().equals("Oracle"));  
 }  
 }  
  
 @Test  
 public void test4() {  
 driver.navigate().to("https://www.facebook.com/");  
 WebElement dropdownMonth = driver.findElement(By.*xpath*("//select[@id='month']"));  
 Select select = new Select(dropdownMonth);  
 //List<WebElement> list = select.getAllSelectedOptions();  
 Assert.*assertTrue*(  
 select.getAllSelectedOptions().get(0).getText().equals("Jul"));  
 select.selectByIndex(1);  
 Assert.*assertTrue*(select.getAllSelectedOptions().get(0).getText().equals("Jan"));  
 for (WebElement li : select.getOptions()) {  
 System.*out*.println(li.getText());  
 }  
 }  
  
 @Test  
 public void test5() {  
 driver.navigate().to("https://www.jquery-az.com/boots/demo.php?ex=63.0\_2");  
 WebElement outputCss = driver.findElement(By.*id*("option-droup-demo"));  
 Select select = new Select(outputCss);  
 select.selectByValue("Java");  
 select.selectByValue("Python");  
 select.deselectByValue("HTML");  
 select.deselectByValue("CSS");  
 List<WebElement>listWithDropDowns = select.getAllSelectedOptions();  
 for(WebElement option : listWithDropDowns){  
 Assert.*assertTrue*(option.getText().equals("Java") || option.getText().equals("Python"));  
 }  
  
 }

public class JavaScriptPractice {  
 WebDriver driver;  
  
 @Before  
 public void setUp() {  
 WebDriverManager.*chromedriver*().setup();  
 driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(5, TimeUnit.*SECONDS*);  
 }  
  
 @Test  
 public void testSofia() {  
 driver.navigate().to("https://opensource-demo.orangehrmlive.com/");  
 JavascriptExecutor js = (JavascriptExecutor) driver;  
 WebDriverWait wait = new WebDriverWait(driver, 3);  
 WebElement usenameInput = driver.findElement(By.*id*("txtUsername"));  
 WebElement passwordInput = driver.findElement(By.*id*("txtPassword"));  
 WebElement logInButton = driver.findElement(By.*id*("btnLogin"));  
 js.executeScript("arguments[0].setAttribute('style', arguments[1]);", usenameInput, "border: 2px solid red");  
 usenameInput.sendKeys("Admin");  
 js.executeScript("arguments[0].setAttribute('style', arguments[1]);", passwordInput, "border: 2px solid red");  
 passwordInput.sendKeys("admin123");  
 js.executeScript("arguments[0].setAttribute('style', arguments[1]);", logInButton, "border: 2px solid red");  
 logInButton.click();  
 WebElement assignLeave = driver.findElement(By.*linkText*("Assign Leave"));  
 js.executeScript("arguments[0].setAttribute('style', arguments[1]);", assignLeave, "border: 2px solid red");  
 assignLeave.click();  
 WebElement employeeName = driver.findElement(By.*id*("assignleave\_txtEmployee\_empName"));  
 js.executeScript("arguments[0].setAttribute('style', arguments[1]);", employeeName, "border: 2px solid red");  
 employeeName.sendKeys("Fiona Grace");  
 //WebElement employeeNameFromList = driver.findElement(By.xpath("//strong[text()='Fiona Grac']"));  
 //js.executeScript("arguments[0].setAttribute('style', arguments[1]);", employeeNameFromList, "border: 2px solid red");  
 //employeeNameFromList.click();  
 WebElement leaveType = driver.findElement(By.*id*("assignleave\_txtLeaveType"));  
 js.executeScript("arguments[0].setAttribute('style', arguments[1]);", leaveType, "border: 2px solid red");  
 Select dropdown = new Select(leaveType);  
 dropdown.selectByValue("1");  
 WebElement fromDate = driver.findElement(By.*id*("assignleave\_txtFromDate"));  
 js.executeScript("arguments[0].setAttribute('style', arguments[1]);", fromDate, "border: 2px solid red");  
 fromDate.click();  
 //  
// List<WebElement> allDates = driver.findElements(By.xpath("//td[@data-handler='selectDay']"));  
// //create local date time  
// int date = LocalDateTime.now().getDayOfMonth();  
// for (WebElement day : allDates) {  
// if (Integer.parseInt(day.getText()) == date)  
// driver.findElement(By.xpath("//a[text()=" + date + "]"));  
// }  
 //  
 // driver.findElement(By.xpath("//a[text()=29]")).click();  
 WebElement toDate = driver.findElement(By.*id*("assignleave\_txtToDate"));  
 js.executeScript("arguments[0].setAttribute('style', arguments[1]);", toDate, "border: 2px solid red");  
 toDate.click();  
 driver.findElement(By.*xpath*("//a[text()=29]")).click();  
 WebElement assignButton = driver.findElement(By.*id*("assignBtn"));  
 js.executeScript("arguments[0].setAttribute('style', arguments[1]);", assignButton, "border: 2px solid red");  
 assignButton.click();

**SofiaSelenium-@Test- JavaScriptPractice**

**Continue page1**

wait.until(ExpectedConditions.*presenceOfElementLocated*(By.*xpath*("//p[text()='Employee does not have sufficient leave balance for leave request.']")));  
 WebElement errorMessage = driver.findElement(By.*xpath*("//p[text()='Employee does not have sufficient leave balance for leave request.']"));  
 Assert.*assertTrue*(errorMessage.isDisplayed());  
 }  
 //if .clear() doesn't work on the input field u can use:  
 // element.sendKeys(Keys.CONTROL+"a"+Keys.DELETE);  
  
  
 @After  
 public void tearDown() {  
 driver.close();  
 driver.quit();  
 }  
  
 @Test  
 public void test1() throws InterruptedException {  
 JavascriptExecutor js = (JavascriptExecutor) driver;  
 js.executeScript("window.location='https://www.etsy.com/'");  
 WebElement signInButton = driver.findElement(By.*xpath*("//nav[@aria-label='Main navigation']//button[contains(text(),'Sign in')]"));  
 js.executeScript("arguments[0].setAttribute('style',arguments[1]);", signInButton, "border: 2px solid red");  
 Thread.*sleep*(2000);  
 js.executeScript("arguments[0].click();", signInButton);  
 WebElement emailInput = driver.findElement(By.*id*("join\_neu\_email\_field"));  
 js.executeScript("arguments[0].setAttribute('style',arguments[1]);", signInButton, "border: 2px solid red");  
 Faker faker = new Faker();  
 String fakeEmail = faker.bothify("?????##@gmail.com");  
  
 js.executeScript("arguments[0].setAttribute('value', arguments[1]);", emailInput, fakeEmail);  
 WebElement passwordInput = driver.findElement(By.*id*("join\_neu\_password\_field"));  
 js.executeScript("arguments[0].setAttribute('style', arguments[1]);", passwordInput, "border: 2px solid red");  
 js.executeScript("arguments[0].setAttribute('value', arguments[1]);", passwordInput, "abs123");  
  
 WebElement sighinButton2 = driver.findElement(By.*xpath*("//button[@value='sign-in']"));  
 js.executeScript("arguments[0].setAttribute('style', arguments[1]);", sighinButton2, "border: 2px solid red");  
 Thread.*sleep*(2000);  
 js.executeScript("arguments[0].click();", sighinButton2);  
 }  
  
 @Test  
 public void test2() throws InterruptedException{  
 WebDriverWait wait = new WebDriverWait(driver,7);  
 JavascriptExecutor js = (JavascriptExecutor) driver;  
 js.executeScript("window.location='https://opensource-demo.orangehrmlive.com'");  
 WebElement inputUserName = driver.findElement(By.*xpath*("//input[@id='txtUsername']"));  
 js.executeScript("arguments[0].setAttribute('style',arguments[1]);", inputUserName, "border: 2px solid red");  
 inputUserName.sendKeys("Admin");  
 WebElement inputPassword = driver.findElement(By.*id*("txtPassword"));  
 js.executeScript("arguments[0].setAttribute('style',arguments[1]);", inputPassword, "border: 2px solid red");  
 inputPassword.sendKeys("admin123");  
 driver.findElement(By.*id*("btnLogin")).click();  
  
 js.executeScript("arguments[0].click();", driver.findElement(By.*xpath*("//img[@src='/webres\_5ebd1457b45137.49759927/orangehrmLeavePlugin/images/ApplyLeave.png']")));  
 WebElement employeeEnter = driver.findElement(By.*xpath*("//input[@id='assignleave\_txtEmployee\_empName']"));

**SofiaSelenium-@Test- JavaScriptPractice**

**Continue page3**

js.executeScript("arguments[0].setAttribute('style', arguments[1]);", employeeEnter, "border: 2px solid red");  
 employeeEnter.sendKeys("Fiona Grace");  
 employeeEnter.click();  
 // WebElement fionaHoverOver = driver.findElement(By.xpath("//li[@class='ac\_even ac\_over']//strong[contains(text(),'Fiona Grac')]"));  
 // wait.until(ExpectedConditions.visibilityOf(fionaHoverOver));  
 // Thread.sleep(3000);  
 // fionaHoverOver.click();  
//  
 WebElement selectElement = driver.findElement(By.*xpath*("//select[@id='assignleave\_txtLeaveType']"));  
 js.executeScript("arguments[0].setAttribute('style', arguments[1]);",selectElement,"border: 2px solid red");  
 Select select = new Select(selectElement);  
 select.selectByVisibleText("Vacation US");  
 WebElement enterCalendar = driver.findElement(By.*xpath*("//input[@id='assignleave\_txtFromDate']"));  
 js.executeScript("arguments[0].setAttribute('style', arguments[1]);", enterCalendar, "border: 2px solid red");  
 WebElement enterCalendarCheckOut = driver.findElement(By.*xpath*("//input[@id='assignleave\_txtToDate']"));  
 enterCalendar.click();  
 int date = LocalDateTime.*now*().getDayOfMonth();  
 driver.findElement(By.*xpath*("//a[contains(text(),'29')]")).click();  
 js.executeScript("arguments[0].setAttribute('style', arguments[1]);", enterCalendarCheckOut, "border: 2px solid red");  
 enterCalendarCheckOut.click();  
 driver.findElement(By.*xpath*("//a[contains(text(),'29')]")).click();  
 WebElement area = driver.findElement(By.*xpath*("//textarea[@id='assignleave\_txtComment']"));  
 Thread.*sleep*(3000);  
 js.executeScript("arguments[0].setAttribute('style', arguments[1]);", area, "border: 2px solid red");  
 area.sendKeys("My comment!!!!");  
 WebElement assign = driver.findElement(By.*xpath*("//input[@id='assignBtn']"));  
 Thread.*sleep*(3000);  
 js.executeScript("arguments[0].click();", assign);  
 Thread.*sleep*(3000);  
   
 WebElement el = wait.until(ExpectedConditions.*presenceOfElementLocated*(By.*xpath*("//div[contains(text(),‘Failed to Assign’)]")));  
 Assert.*assertTrue*(el.getText().contains("Failed to Assign"));  
  
 }

public class Iframes {  
 WebDriver driver;  
  
 @Before  
 public void setUp() {  
 WebDriverManager.*chromedriver*().setup();  
 driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(5, TimeUnit.*SECONDS*);  
 }  
  
 @After  
 public void tearDown() {  
 driver.close();  
 driver.quit();  
 }  
 @Test  
 public void test1(){  
  
 driver.navigate().to("https://demoqa.com/frames");  
  
 driver.switchTo().frame("frame1");  
 WebElement textOfFrame1 = driver.findElement(By.*tagName*("h1"));  
 Assert.*assertTrue*(textOfFrame1.isDisplayed());  
  
 //driver.switchTo().defaultContent();  
  
 driver.switchTo().frame(1);  
 WebElement textOfFrame2 = driver.findElement(By.*id*("sampleHeading"));  
  
 Assert.*assertEquals*("This is a sample page",textOfFrame2.getText());  
  
 }  
 @Test  
 public void test2(){  
 driver.navigate().to("https://allwebco-templates.com/support/S\_script\_IFrame.htm");  
  
 driver.switchTo().frame(driver.findElement(By.*xpath*("//iframe[@class='framesample']")));  
  
 System.*out*.println(driver.findElement(By.*tagName*("div")).getText());  
  
 driver.switchTo().defaultContent();  
  
 driver.findElement(By.*xpath*("//a[text()='Free website scripts']")).click();  
  
 Assert.*assertTrue*(driver.findElement(By.*xpath*("//h1[@class='title titlespad hidemobile610']")).isDisplayed());  
 }  
 @Test  
 public void test3(){  
 driver.navigate().to("https://demoqa.com/nestedframes");  
  
 driver.switchTo().frame("frame1");  
 driver.switchTo().frame(0);  
 Assert.*assertEquals*("Child Iframe",driver.findElement(By.*tagName*("p")).getText());  
  
 driver.switchTo().parentFrame();  
 Assert.*assertEquals*("Parent frame",driver.findElement(By.*tagName*("body")).getText());  
  
 driver.switchTo().defaultContent();  
 }

public class AlertsPractice {  
 WebDriver driver;

@Before  
 public void setUp() {  
 WebDriverManager.*chromedriver*().setup();  
 driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(5, TimeUnit.*SECONDS*);  
 }

@After  
 public void tearDown() {  
 driver.close();  
 driver.quit();  
 }

@Test  
 public void test1() throws InterruptedException {  
 driver.navigate().to("https://demoqa.com/alerts");  
 driver.findElement(By.*id*("alertButton")).click();  
 Alert alert = driver.switchTo().alert();  
 alert.accept();  
 driver.findElement(By.*id*("timerAlertButton")).click();  
 new WebDriverWait(driver, 7).until(ExpectedConditions.*alertIsPresent*());  
 alert.accept();  
 driver.findElement(By.*id*("confirmButton")).click();  
 alert.dismiss();  
  
 String resultText = driver.findElement(By.*id*("confirmResult")).getText();  
  
 Assert.*assertEquals*("You selected Cancel", resultText);  
  
 driver.findElement(By.*id*("promtButton")).click();  
 System.*out*.println("The text of alert is: " + alert.getText());  
  
 alert.sendKeys("Baha");  
 alert.accept();  
 String promptTextResult = driver.findElement(By.*id*("promtButton")).getText();  
 Assert.*assertTrue*(promptTextResult.contains("Baha"));  
 }  
  
 @Test  
 public void test2() {  
 //Task #11. Navigate to https://chercher.tech/practice/practice-pop-ups-selenium-webdriver  
 // 2. Double click on the button “Double Click Me”  
 // 3. Verify the text of the alert contains “You double clicked me!!!”  
 // 4. Click ok on the alert  
 // 5. Click on the Promptbutton  
 // 6. Verify the text of the alert equals to “I am prompt”  
 // 7. Pass your name to the prompt  
 // 8.Clickon Confirmation box  
 // 9. Verify the text of the alert equals to “I am confirm”  
 // 10. Cancel the aler  
 driver.navigate().to("https://chercher.tech/practice/practice-pop-ups-selenium-webdriver");  
 WebElement doubleClickElem = driver.findElement(By.*xpath*("//input[@value='Double Click Me']"));  
 Actions actions = new Actions(driver);  
 actions.doubleClick(doubleClickElem).perform();  
 Alert alert = driver.switchTo().alert();  
 String alertText = alert.getText();  
 Assert.*assertTrue*(alert.getText().contains("You double clicked me!"));  
 alert.accept();  
 driver.findElement(By.*name*("prompt")).click();  
 Assert.*assertEquals*("I am prompt", alert.getText());  
 alert.sendKeys("DevX");  
 alert.accept();  
  
 driver.findElement(By.*xpath*("//\*[@name='confirmation']"));  
 Assert.*assertEquals*("I am confirm",alert.getText());  
 alert.dismiss();  
  
 }

public class TaskSmallFromSofia {  
 @Test  
 public void test1() throws InterruptedException {  
 WebDriverManager.*chromedriver*().setup();  
  
 WebDriver driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(10, TimeUnit.*SECONDS*);  
  
 driver.get("https://www.practiceselenium.com/");  
 driver.findElement(By.*partialLinkText*("let's Talk")).click();  
 Thread.*sleep*(3000);  
 WebElement nameInput = driver.findElement(By.*name*("name"));  
  
 Assert.*assertTrue*(nameInput.isDisplayed());  
 //driver.close();  
  
 }  
  
 @Test  
 public void test2() throws InterruptedException {  
 WebDriverManager.*chromedriver*().setup();  
  
 WebDriver driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(10, TimeUnit.*SECONDS*);  
  
 driver.get("https://www.amazon.com/");  
 driver.findElement(By.*linkText*("Sell")).click();  
 Thread.*sleep*(4000);  
 WebElement title = driver.findElement(By.*tagName*("h1"));  
 Assert.*assertTrue*(title.isDisplayed());  
 driver.findElement(By.*linkText*("Sign up")).click();  
 Thread.*sleep*(4000);  
 WebElement emailInput = driver.findElement(By.*id*("ap\_email"));  
 WebElement passwordInputField = driver.findElement(By.*id*("ap\_password"));  
 WebElement nextButton = driver.findElement(By.*id*("signInSubmit"));  
  
 emailInput.sendKeys("invalidEmail");  
 passwordInputField.sendKeys("pass123");  
 nextButton.click();  
 Thread.*sleep*(2000);  
 WebElement errorMessage = driver.findElement(By.*tagName*("h4"));  
 Assert.*assertTrue*(errorMessage.isDisplayed());  
 driver.close();  
  
  
 }

**SofiaSelenium-@Test- SmallTaskFromSofia**

@Test  
 public void test3() throws InterruptedException {  
 WebDriverManager.*chromedriver*().setup();  
  
 WebDriver driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(10, TimeUnit.*SECONDS*);  
  
 driver.get("https://www.practiceselenium.com/");  
 Assert.*assertTrue*(driver.findElement(By.*tagName*("h1")).isDisplayed());  
  
 }  
  
 @Test  
 public void test4() throws InterruptedException {  
 WebDriverManager.*chromedriver*().setup();  
  
 WebDriver driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(10, TimeUnit.*SECONDS*);  
  
 driver.get("https://www.practiceselenium.com/");  
 driver.findElement(By.*linkText*("Check Out")).click();  
 Thread.*sleep*(3000);  
 driver.findElement(By.*linkText*("Cancel")).click();  
  
  
 driver.close();  
 }

}

package basicLocatorsPractice;

public class Locators {  
 @Test  
 public void test1() throws InterruptedException {  
 //to set up the driver instead system.setProperty we do this:  
 WebDriverManager.*chromedriver*().setup();  
  
 WebDriver driver = new ChromeDriver();  
 //gi driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(10, TimeUnit.*SECONDS*);  
 driver.get("https://www.google.ru/");  
  
 WebElement searchInputField = driver.findElement(By.*name*("q"));  
 String searchCriteria = "apple";  
 searchInputField.sendKeys(searchCriteria);  
 driver.findElement(By.*name*("btnK")).click();  
 Thread.*sleep*(5000);  
 Assert.*assertTrue*("The title doesn't contain the search criteria. Expected: " + searchCriteria + ". Actual: " + driver.getTitle(), driver.getTitle().contains(searchCriteria));  
  
 driver.close();  
 }  
  
 @Test  
 public void test2() throws InterruptedException{  
 WebDriverManager.*chromedriver*().setup();  
  
 WebDriver driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(10, TimeUnit.*SECONDS*);  
  
 driver.get("https://www.google.com");  
 driver.findElement(By.*linkText*("Gmail")).click();  
 Thread.*sleep*(3000);  
 WebElement createAnAccountButton = driver.findElement(By.*partialLinkText*("Create an"));  
  
 Assert.*assertTrue*(createAnAccountButton.isDisplayed());  
 driver.close();  
  
 }  
 @Test  
 public void test3()throws InterruptedException{  
 WebDriverManager.*chromedriver*().setup();  
  
 WebDriver driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(10, TimeUnit.*SECONDS*);  
  
 driver.get("https://demostore.x-cart.com/");  
 driver.findElement(By.*linkText*("Contact us")).click();  
 Thread.*sleep*(3000);  
 WebElement headind = driver.findElement(By.*tagName*("h1"));  
  
 Assert.*assertTrue*(headind.isDisplayed());  
 driver.close();  
 }

**SofiaSelenium-@Test- Locators**

@Test  
 public void test4() throws InterruptedException {  
 WebDriverManager.*chromedriver*().setup();  
  
 WebDriver driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(10, TimeUnit.*SECONDS*);  
  
 driver.get("https://www.salesforce.com/");  
 driver.findElement(By.*xpath*("//\*[@id='globalnavbar-header-container']/div[2]/div/div[5]/div/div/a")).click();  
 driver.findElement(By.*id*("username")).sendKeys("Hello");  
 Thread.*sleep*(3000);  
 driver.findElement(By.*id*("password")).sendKeys("pa$$1234");  
 Thread.*sleep*(3000);  
 driver.findElement(By.*cssSelector*("#Login")).click();  
 WebElement errorMessageToLogin = driver.findElement(By.*cssSelector*("#error"));  
 System.*out*.println(errorMessageToLogin.getText());  
 Assert.*assertTrue*(errorMessageToLogin.isDisplayed());  
  
 driver.close();  
  
 }  
  
 @Test  
 public void test5()throws InterruptedException{  
  
 WebDriverManager.*chromedriver*().setup();  
 WebDriver driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(10,TimeUnit.*SECONDS*);  
  
 driver.get("https://www.facebook.com/");  
 driver.findElement(By.*xpath*("//\*[@id='email']")).sendKeys("Hello");  
 driver.findElement(By.*cssSelector*("input[type='submit']")).click();  
  
 Thread.*sleep*(3000);  
 driver.findElement(By.*cssSelector*("input[type='text']")).sendKeys("phone number");  
 Thread.*sleep*(3000);  
 driver.findElement(By.*xpath*("//\*[@id='month']")).click();  
 driver.close();  
  
  
 }

FFF

public class AdvancedLocators {  
 @Test  
 public void test1() throws InterruptedException {  
 WebDriverManager.*chromedriver*().setup();  
 WebDriver driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(7, TimeUnit.*SECONDS*);  
  
 driver.get("https://www.google.com");  
 WebElement searchInputField = driver.findElement(By.*name*("q"));  
 searchInputField.sendKeys("devxschool");  
 driver.findElement(By.*name*("btnK")).click();  
 Thread.*sleep*(4000);  
 WebElement firstLink = driver.findElement(By.*tagName*("a"));  
 //element.getText() - returns a String = the text of the webelement  
 System.*out*.println(firstLink.getText());  
 //find aaLL THE LINKS ON THE PAGE  
 System.*out*.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");  
  
 List<WebElement> allLinksOnThePage = driver.findElements(By.*tagName*("a"));  
 for (WebElement link : allLinksOnThePage) {  
 System.*out*.println(link.getText());  
// if(link.getText().contains("DevX School")){  
// link.click();  
// break;  
// }  
 }  
 driver.close();  
  
 }  
  
 @Test  
 public void test2() throws InterruptedException {  
 WebDriverManager.*chromedriver*().setup();  
 WebDriver driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(7, TimeUnit.*SECONDS*);  
  
 driver.get("https://www.amazon.com/");  
  
 driver.findElement(By.*cssSelector*("#twotabsearchtextbox")).sendKeys("iPhone" + Keys.*ENTER*);  
 Thread.*sleep*(3000);  
 driver.findElement(By.*xpath*("//span[text()=‘See more’][1]")).click();  
 Thread.*sleep*(3000);  
 List<WebElement> brandsOfPhones = driver.findElements(By.*cssSelector*("pi[id^='p\_89']"));  
 for (WebElement element : brandsOfPhones) {  
 System.*out*.println(element.getText());  
 }  
 Assert.*assertTrue*("The brands displayed must be 10", brandsOfPhones.size() < 10);  
  
 driver.close();  
  
  
 }

**Sofia-@Test-Advanced Locators**

**Meetup\_config.properties**

@Test  
 public void test3() throws InterruptedException {  
 WebDriverManager.*chromedriver*().setup();  
 WebDriver driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(7, TimeUnit.*SECONDS*);  
 driver.get("https://demoqa.com/checkbox");  
 driver.findElement(By.*xpath*("//button[@title='Toggle']")).click();  
 Thread.*sleep*(2000);  
 List<WebElement> subfolders = driver.findElements(By.*xpath*("//\*[@class='rct-icon rct-icon-expand-close']/.."));  
 for (WebElement subfolder : subfolders) {  
 System.*out*.println(subfolder.getText());  
 }  
 subfolders.get(1).click();  
 Assert.*assertTrue*(subfolders.size() == 3 && subfolders.get(1).isSelected());  
 }  
  
  
 @Test  
 public void test4() throws InterruptedException {  
 WebDriverManager.*chromedriver*().setup();  
 WebDriver driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(7, TimeUnit.*SECONDS*);  
 driver.get("https://www.rediff.com/");  
 driver.findElement(By.*xpath*("//\*[contains(@class,'signin')]")).click();  
 driver.findElement(By.*xpath*("//\*[@id='login1']")).sendKeys("Hello");  
 driver.findElement(By.*xpath*("//input[@id='password']")).sendKeys("pa$$123");  
 Thread.*sleep*(3000);  
 driver.findElement(By.*xpath*("//\*[@type='submit']")).click();  
 Thread.*sleep*(3000);  
 driver.close();  
 //rct-icon rct-icon-expand-close  
  
 }  
  
 @Test  
 public void test5() throws InterruptedException {  
 WebDriverManager.*chromedriver*().setup();  
 WebDriver driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(7, TimeUnit.*SECONDS*);  
 driver.get("https://demoqa.com/radio-button");  
 WebElement getTextFromTheSite = driver.findElement(By.*xpath*("//\*[@class='mb-3']"));  
 System.*out*.println(getTextFromTheSite.getText());  
 driver.findElement(By.*xpath*("//\*[@class='custom-control-label']")).click();  
 WebElement showMessage = driver.findElement(By.*xpath*("//p[contains(text(),'You have selected ')]"));  
 Thread.*sleep*(4000);  
 List<WebElement> answersToRadioButtons = driver.findElements(By.*xpath*("//div[starts-with(@class,'custom-control')]"));  
 for (WebElement answer : answersToRadioButtons) {  
 if (!answer.isSelected()) {  
 System.*out*.println("answer is selected : " + answer.getText());  
 } else {  
 System.*out*.println("answer is not selected : " + answer.getText());  
 }  
 }  
 Assert.*assertTrue*(showMessage.isDisplayed());  
 driver.close();  
 }

**Sofia-@Test-Advanced Locators**

**Meetup\_config.properties**

@Test  
 public void test6() throws InterruptedException {  
 WebDriverManager.*chromedriver*().setup();  
 WebDriver driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(7, TimeUnit.*SECONDS*);  
 driver.get("http://www.qaclickacademy.com/interview.php");  
 driver.findElement(By.*xpath*("//li[text()=' Selenium ']")).click();  
  
 driver.findElement(By.*xpath*("//ul[@class='responsive-tabs\_\_list']/li[1]/following-sibling::li[2]")).click();  
  
 System.*out*.println(driver.findElement(By.*xpath*(".//\*[@id='tablist1-tab2']/parent::ul")).getAttribute("role"));  
 }  
  
 @Test  
 public void test7() throws InterruptedException {  
 WebDriverManager.*chromedriver*().setup();  
 WebDriver driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(7, TimeUnit.*SECONDS*);  
 driver.get("https://www.spicejet.com/");  
 Select passengers = new Select(driver.findElement(By.*xpath*("//div[@id='divpaxinfo']")));  
 // s.selectByValue();  
 Select adults = new Select(driver.findElement(By.*xpath*("//\*[@id='ctl00\_mainContent\_ddl\_Adult']")));  
 adults.selectByValue("2");  
 Thread.*sleep*(3000);  
 driver.close();  
 }  
  
 @Test  
 public void test8() throws InterruptedException {  
 WebDriverManager.*chromedriver*().setup();  
 WebDriver driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(7, TimeUnit.*SECONDS*);  
 driver.get("https://opensource-demo.orangehrmlive.com/");  
 String userName = "Admin";  
 String password = "admin123";  
 WebElement userNameInputFiled = driver.findElement(By.*xpath*("//input[@id='txtUsername']"));  
 WebElement passwordInputFiled = driver.findElement(By.*xpath*("//input[@id='txtPassword']"));  
 WebElement loginButton = driver.findElement(By.*xpath*("//input[@type='submit']"));  
 userNameInputFiled.sendKeys(userName);  
 passwordInputFiled.sendKeys(password);  
 loginButton.click();  
 Thread.*sleep*(3000);  
 WebElement welcomeText = driver.findElement(By.*xpath*("//a[contains(text(),'Welcome')]"));  
 Assert.*assertTrue*(welcomeText.getText().contains(userName));  
 ////a[@class='panelTrigger activated-welcome']

baseUrl = https://www.meetup.com/  
username = JohnDoe  
password = pass123  
  
browser = google

headless = false

**hedless =**

package utilities;  
  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.safari.SafariDriver;  
  
  
import java.util.concurrent.TimeUnit;  
  
public class Driver {  
 private Driver() {  
 }  
  
 ;  
 private static WebDriver *driver*;  
  
 //will not allow anyone to create a driver object directly  
 //1st method for creating a driver depending on browser specified in config.properties  
 public static WebDriver getDriver() {  
  
 if (*driver* == null) {  
 //we will create a new driver instance here and assign it to our driver variable  
 switch (ConfigsReader.*getProperty*("browser").toLowerCase()) {  
 case "chrome":  
 *driver* =ChromeWebDriver.*loadChromeDriver*(Boolean.*parseBoolean*(ConfigsReader.*getProperty*("headless")));  
 break;  
 case "firefox":  
 *driver* = FireFoxWebDriver.*loadFireFoxDriver*(Boolean.*parseBoolean*(ConfigsReader.*getProperty*("headless")));  
 break;  
// case "safari":  
// driver = new SafariDriver();  
// driver.manage().window().maximize();  
// driver.manage().timeouts().implicitlyWait(7, TimeUnit.SECONDS);  
// break;  
 default:  
 *driver* =ChromeWebDriver.*loadChromeDriver*(Boolean.*parseBoolean*(ConfigsReader.*getProperty*("headless")));  
 break;  
  
 }  
 }  
 return *driver*;  
  
 }  
  
 public static void closeDriver() {  
 if (*driver* == null) return;  
  
 try {  
 *driver*.close();  
 *driver*.quit();  
 *driver* = null;  
 } catch (Exception e) {  
 e.printStackTrace();}  
 }

baseUrl = https://www.meetup.com/  
username = JohnDoe  
password = pass123  
  
browser = google

baseUrl = https://www.meetup.com/  
username = JohnDoe  
password = pass123  
  
browser = google  
headless = false

package utilities;  
  
public class ConfigsReader {  
  
 private static Properties *properties*;  
  
 static {  
 //here we will load out config file so that class can read from it  
 try {  
 String path = "src/test/resources/meetup\_configs.properties";  
 FileInputStream input = new FileInputStream(path);  
  
 *properties* = new Properties();  
 *properties*.load(input);  
 input.close();  
 }catch (IOException e){  
 e.printStackTrace();  
 }  
 }  
  
 //only one method to read a value from //configs.properties file  
 public static String getProperty(String key){  
 return *properties*.getProperty(key);  
 }

package utilities;  
  
import io.github.bonigarcia.wdm.WebDriverManager;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.firefox.FirefoxDriver;  
import org.openqa.selenium.firefox.FirefoxOptions;  
  
import java.util.concurrent.TimeUnit;  
  
public class FireFoxWebDriver {  
 public static WebDriver loadFireFoxDriver(boolean headless){  
 WebDriverManager.*firefoxdriver*().setup();  
  
 FirefoxOptions options = new FirefoxOptions();  
 options.setHeadless(headless);  
  
 WebDriver driver = new FirefoxDriver(options);  
  
 driver.manage().timeouts().implicitlyWait(8, TimeUnit.*SECONDS*);  
 driver.manage().timeouts().pageLoadTimeout(20,TimeUnit.*SECONDS*);  
 driver.manage().window().maximize();  
  
 return driver;  
 }

package utilities;  
  
import io.github.bonigarcia.wdm.WebDriverManager;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.chrome.ChromeDriver;  
import org.openqa.selenium.chrome.ChromeOptions;  
  
import java.util.concurrent.TimeUnit;  
  
public class ChromeWebDriver {  
 public static WebDriver loadChromeDriver(boolean headless){  
 WebDriverManager.*chromedriver*().setup();  
  
 ChromeOptions options = new ChromeOptions();  
 options.addArguments("--disable-extensions");  
 options.addArguments("--start-maximized");  
 options.addArguments("--window-size=1920,1080");  
  
 if(headless){  
 options.setHeadless(true);  
 //options.addArguments("--headless");  
 }  
  
 WebDriver driver = new ChromeDriver(options);  
  
 driver.manage().timeouts().implicitlyWait(7, TimeUnit.*SECONDS*);  
 // driver.manage().timeouts().pageLoadTimeout(15, TimeUnit.SECONDS);  
 return driver;  
  
 }

package Steps;  
  
public class ChicagoOnMeetupVerificationSteps {  
 CHicagoOnmeetupVerificationPage chicagoPage = new CHicagoOnmeetupVerificationPage();  
   
  
 @When("^the user types \"([^\"]\*)\" in the \"([^\"]\*)\" field$")  
 public void the\_user\_types\_in\_the\_field(String software, String FindNextEvent) throws Throwable {  
 chicagoPage.inputFieldForNextEvent.sendKeys(software);  
 }  
  
 @When("^the user types \"([^\"]\*)\"$")  
 public void the\_user\_types(String ChicagoIL) throws Throwable {  
 //chicagoPage.inputFindByLocationCity.sendKeys("Chicago");  
 chicagoPage.inputFindByLocationCity.click();  
 chicagoPage.inputFindByLocationCity.sendKeys("Chicago");  
 chicagoPage.inputFindByLocationCity.click();  
 }  
  
 @Then("^the user should select \"([^\"]\*)\" from the list of autosuggested values$")  
 public void the\_user\_should\_select\_from\_the\_list\_of\_autosuggested\_values(String ChicagoIL) throws Throwable {  
 WebDriverWait wait = new WebDriverWait(Driver.*getDriver*(),5);  
 // wait.until(ExpectedConditions.presenceOfElementLocated(By.xpath("//span[contains(text(),'ago, Illinois, USA') and @data-suggested-value='true']")));  
 // chicagoPage.autoSuggestionChicagoIlFromList.click();  
 }  
  
 @Then("^user clicks Search button$")  
 public void user\_clicks\_Search\_button() throws Throwable {  
 chicagoPage.searchButtonOnEvent.click();  
  
 }  
  
 @When("^results are loaded$")  
 public void results\_are\_loaded() throws Throwable {  
  
 }  
  
 @Then("^the user should change \"([^\"]\*)\" field to \"([^\"]\*)\"$")  
 public void the\_user\_should\_change\_field\_to(String anyDay, String tomorrow) throws Throwable {  
 chicagoPage.buttonWhichShowsAnyDay.click();  
 chicagoPage.autoSuggestionOnButtonToday.click();  
  
 }  
  
 @Then("^the user should verify all search results contain tomorrows$")  
 public void the\_user\_should\_verify\_all\_search\_results\_contain\_tomorrows() throws Throwable {  
 Assert.*assertTrue*(chicagoPage.eventFullStackNYCCampusTour.isDisplayed() &&  
 chicagoPage.eventUpgradindToOracleDB.isDisplayed() &&  
 chicagoPage.eventProductAmazon0ToOne.isDisplayed() &&  
 chicagoPage.eventOlympusDemoDay.isDisplayed() &&  
 chicagoPage.eventFridayNightMixer.isDisplayed());  
}

package runners;  
  
@RunWith(Cucumber.class)  
//cucmber options are annotations which take config for the runner class  
//features will take a relate path to the folder where feature are stored  
//cucumber will scan for all features files under that folder  
//glue is a property()  
@CucumberOptions(  
 plugin = {"pretty","html:target/default-cucumber-reports",  
 "json:target/cucumber.json"},  
 features = {"classpath:features"},  
 glue = {"Steps"},  
 // tags = {"~@Ignore"},  
 tags = {" @ScrollButtonTest"},  
 dryRun = false

)  
public class Runner {  
 @BeforeClass  
 public static void beforeClassHook() {  
 System.out.println("Go to website"); System.out.println("Login to the app");  
 }  
  
 @AfterClass  
 public static void afterClass() {  
 System.out.println("Close browser");

public class CHicagoOnmeetupVerificationPage {  
 WebDriver driver;  
  
 public CHicagoOnmeetupVerificationPage() {  
 driver = Driver.*getDriver*();  
 PageFactory.*initElements*(driver, this);  
 }  
  
 @FindBy(id = "tracking--searchComponentInput")  
 public WebElement inputFieldForNextEvent;  
  
 @FindBy(xpath = "//input[@aria-label='Search for location by city or zip code']")  
 public WebElement inputFindByLocationCity;

F

package pages;  
  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.WebElement;  
import org.openqa.selenium.support.FindBy;  
import org.openqa.selenium.support.PageFactory;  
import utilities.Driver;  
  
public class CHicagoOnmeetupVerificationPage {  
 WebDriver driver;  
  
 public CHicagoOnmeetupVerificationPage() {  
 driver = Driver.*getDriver*();  
 PageFactory.*initElements*(driver, this);  
 }  
  
 @FindBy(id = "tracking--searchComponentInput")  
 public WebElement inputFieldForNextEvent;  
  
 @FindBy(xpath = "//input[@aria-label='Search for location by city or zip code']")  
 public WebElement inputFindByLocationCity;  
  
 @FindBy(xpath = "//span[contains(text(),'ago, Illinois, USA') and @data-suggested-value='true']")  
 public WebElement autoSuggestionChicagoIlFromList;

d

package runners;  
  
import cucumber.api.CucumberOptions;  
import cucumber.api.java.Before;  
import cucumber.api.junit.Cucumber;  
import org.junit.AfterClass;  
import org.junit.BeforeClass;  
import org.junit.runner.RunWith;  
  
  
@RunWith(Cucumber.class)  
//cucmber options are annotations which take config for the runner class  
//features will take a relate path to the folder where feature are stored  
//cucumber will scan for all features files under that folder  
//glue is a property()  
@CucumberOptions(  
 plugin = {"pretty","html:target/default-cucumber-reports",  
 "json:target/cucumber.json"},  
 features = {"classpath:features"},  
 glue = {"Steps"},  
 // tags = {"~@Ignore"},  
 tags = {" @ScrollButtonTest"},  
 dryRun = false  
)  
public class Runner {  
// @BeforeClass  
// public static void beforeClassHook() {  
// System.out.println("Go to website");  
// System.out.println("Login to the app");  
// }  
//  
// @AfterClass  
// public static void afterClass() {  
// System.out.println("Close browser");  
// }  
  
}

dd

package Steps;  
  
import cucumber.api.java.en.\*;  
import org.junit.Assert;  
import org.openqa.selenium.By;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.support.ui.ExpectedConditions;  
import org.openqa.selenium.support.ui.WebDriverWait;  
import pages.CHicagoOnmeetupVerificationPage;  
import utilities.Driver;  
  
public class ChicagoOnMeetupVerificationSteps {  
 CHicagoOnmeetupVerificationPage chicagoPage = new CHicagoOnmeetupVerificationPage();  
   
  
 @When("^the user types \"([^\"]\*)\" in the \"([^\"]\*)\" field$")  
 public void the\_user\_types\_in\_the\_field(String software, String FindNextEvent) throws Throwable {  
 chicagoPage.inputFieldForNextEvent.sendKeys(software);  
 }  
  
 @When("^the user types \"([^\"]\*)\"$")  
 public void the\_user\_types(String ChicagoIL) throws Throwable {  
 //chicagoPage.inputFindByLocationCity.sendKeys("Chicago");  
 chicagoPage.inputFindByLocationCity.click();  
 chicagoPage.inputFindByLocationCity.sendKeys("Chicago");  
 chicagoPage.inputFindByLocationCity.click();  
  
  
 }  
  
 @Then("^the user should select \"([^\"]\*)\" from the list of autosuggested values$")  
 public void the\_user\_should\_select\_from\_the\_list\_of\_autosuggested\_values(String ChicagoIL) throws Throwable {  
 WebDriverWait wait = new WebDriverWait(Driver.*getDriver*(),5);  
 // wait.until(ExpectedConditions.presenceOfElementLocated(By.xpath("//span[contains(text(),'ago, Illinois, USA') and @data-suggested-value='true']")));  
 // chicagoPage.autoSuggestionChicagoIlFromList.click();  
 }  
  
 @Then("^user clicks Search button$")  
 public void user\_clicks\_Search\_button() throws Throwable {  
 chicagoPage.searchButtonOnEvent.click();  
  
 }  
  
 @When("^results are loaded$")  
 public void results\_are\_loaded() throws Throwable {  
  
 }  
  
 @Then("^the user should change \"([^\"]\*)\" field to \"([^\"]\*)\"$")  
 public void the\_user\_should\_change\_field\_to(String anyDay, String tomorrow) throws Throwable {  
 chicagoPage.buttonWhichShowsAnyDay.click();  
 chicagoPage.autoSuggestionOnButtonToday.click();  
  
 }  
  
 @Then("^the user should verify all search results contain tomorrows$")  
 public void the\_user\_should\_verify\_all\_search\_results\_contain\_tomorrows() throws Throwable {  
 Assert.*assertTrue*(chicagoPage.eventFullStackNYCCampusTour.isDisplayed() &&  
 chicagoPage.eventUpgradindToOracleDB.isDisplayed() &&  
 chicagoPage.eventProductAmazon0ToOne.isDisplayed() &&  
 chicagoPage.eventOlympusDemoDay.isDisplayed() &&  
 chicagoPage.eventFridayNightMixer.isDisplayed());  
  
 }  
  
  
}

f

package utilities;  
  
import io.github.bonigarcia.wdm.WebDriverManager;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.chrome.ChromeDriver;  
import org.openqa.selenium.chrome.ChromeOptions;  
  
import java.util.concurrent.TimeUnit;  
  
public class ChromeWebDriver {  
 public static WebDriver loadChromeDriver(boolean headless){  
 WebDriverManager.*chromedriver*().setup();  
  
 ChromeOptions options = new ChromeOptions();  
 options.addArguments("--disable-extensions");  
 options.addArguments("--start-maximized");  
 options.addArguments("--window-size=1920,1080");  
  
 if(headless){  
 options.setHeadless(true);  
 //options.addArguments("--headless");  
 }  
  
 WebDriver driver = new ChromeDriver(options);  
  
 driver.manage().timeouts().implicitlyWait(7, TimeUnit.*SECONDS*);  
 // driver.manage().timeouts().pageLoadTimeout(15, TimeUnit.SECONDS);  
 return driver;  
  
 }  
}

d

package utilities;  
  
import io.github.bonigarcia.wdm.WebDriverManager;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.firefox.FirefoxDriver;  
import org.openqa.selenium.firefox.FirefoxOptions;  
  
import java.util.concurrent.TimeUnit;  
  
public class FireFoxWebDriver {  
 public static WebDriver loadFireFoxDriver(boolean headless){  
 WebDriverManager.*firefoxdriver*().setup();  
  
 FirefoxOptions options = new FirefoxOptions();  
 options.setHeadless(headless);  
  
 WebDriver driver = new FirefoxDriver(options);  
  
 driver.manage().timeouts().implicitlyWait(8, TimeUnit.*SECONDS*);  
 driver.manage().timeouts().pageLoadTimeout(20,TimeUnit.*SECONDS*);  
 driver.manage().window().maximize();  
  
 return driver;  
 }  
}

d

package utilities;  
  
import java.io.FileInputStream;  
import java.io.IOException;  
import java.util.Properties;  
  
public class ConfigsReader {  
  
  
 private static Properties *properties*;  
  
 static {  
 //here we will load out config file so that class can read from it  
 try {  
 String path = "src/test/resources/meetup\_configs.properties";  
 FileInputStream input = new FileInputStream(path);  
  
 *properties* = new Properties();  
 *properties*.load(input);  
 input.close();  
 }catch (IOException e){  
 e.printStackTrace();  
 }  
 }  
  
 //only one method to read a value from configs.properties file  
 public static String getProperty(String key){  
 return *properties*.getProperty(key);  
 }  
 }

f

package utilities;  
  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.safari.SafariDriver;  
  
  
import java.util.concurrent.TimeUnit;  
  
public class Driver {  
 private Driver() {  
 }  
  
 ;  
 private static WebDriver *driver*;  
  
 //will not allow anyone to create a driver object directly  
 //1st method for creating a driver depending on browser specified in config.properties  
 public static WebDriver getDriver() {  
  
 if (*driver* == null) {  
 //we will create a new driver instance here and assign it to our driver variable  
 switch (ConfigsReader.*getProperty*("browser").toLowerCase()) {  
 case "chrome":  
 *driver* =ChromeWebDriver.*loadChromeDriver*(Boolean.*parseBoolean*(ConfigsReader.*getProperty*("headless")));  
 break;  
 case "firefox":  
 *driver* = FireFoxWebDriver.*loadFireFoxDriver*(Boolean.*parseBoolean*(ConfigsReader.*getProperty*("headless")));  
 break;  
// case "safari":  
// driver = new SafariDriver();  
// driver.manage().window().maximize();  
// driver.manage().timeouts().implicitlyWait(7, TimeUnit.SECONDS);  
// break;  
 default:  
 *driver* =ChromeWebDriver.*loadChromeDriver*(Boolean.*parseBoolean*(ConfigsReader.*getProperty*("headless")));  
 break;  
  
 }  
 }  
 return *driver*;  
  
 }  
  
 public static void closeDriver() {  
 if (*driver* == null) return;  
  
 try {  
 *driver*.close();  
 *driver*.quit();  
 *driver* = null;  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
}

**f**

public class AdvancedLocators {  
 @Test  
 public void test1() throws InterruptedException {  
 WebDriverManager.*chromedriver*().setup();  
 WebDriver driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(7, TimeUnit.*SECONDS*);  
  
 driver.get("https://www.google.com");  
 WebElement searchInputField = driver.findElement(By.*name*("q"));  
 searchInputField.sendKeys("devxschool");  
 driver.findElement(By.*name*("btnK")).click();  
 Thread.*sleep*(4000);  
 WebElement firstLink = driver.findElement(By.*tagName*("a"));  
 //element.getText() - returns a String = the text of the webelement  
 System.*out*.println(firstLink.getText());  
 //find aaLL THE LINKS ON THE PAGE  
 System.*out*.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");  
  
 List<WebElement> allLinksOnThePage = driver.findElements(By.*tagName*("a"));  
 for (WebElement link : allLinksOnThePage) {  
 System.*out*.println(link.getText());  
// if(link.getText().contains("DevX School")){  
// link.click();  
// break;  
// }  
 }  
 driver.close();  
  
 }  
  
 @Test  
 public void test2() throws InterruptedException {  
 WebDriverManager.*chromedriver*().setup();  
 WebDriver driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(7, TimeUnit.*SECONDS*);  
  
 driver.get("https://www.amazon.com/");  
  
 driver.findElement(By.*cssSelector*("#twotabsearchtextbox")).sendKeys("iPhone" + Keys.*ENTER*);  
 Thread.*sleep*(3000);  
 driver.findElement(By.*xpath*("//span[text()=‘See more’][1]")).click();  
 Thread.*sleep*(3000);  
 List<WebElement> brandsOfPhones = driver.findElements(By.*cssSelector*("pi[id^='p\_89']"));  
 for (WebElement element : brandsOfPhones) {  
 System.*out*.println(element.getText());  
 }  
 Assert.*assertTrue*("The brands displayed must be 10", brandsOfPhones.size() < 10);  
  
 driver.close();  
  
  
 }  
  
 @Test  
 public void test3() throws InterruptedException {  
 WebDriverManager.*chromedriver*().setup();  
 WebDriver driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(7, TimeUnit.*SECONDS*);  
 driver.get("https://demoqa.com/checkbox");  
 driver.findElement(By.*xpath*("//button[@title='Toggle']")).click();  
 Thread.*sleep*(2000);  
 List<WebElement> subfolders = driver.findElements(By.*xpath*("//\*[@class='rct-icon rct-icon-expand-close']/.."));  
 for (WebElement subfolder : subfolders) {  
 System.*out*.println(subfolder.getText());  
 }  
 subfolders.get(1).click();  
 Assert.*assertTrue*(subfolders.size() == 3 && subfolders.get(1).isSelected());  
 }  
  
  
 @Test  
 public void test4() throws InterruptedException {  
 WebDriverManager.*chromedriver*().setup();  
 WebDriver driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(7, TimeUnit.*SECONDS*);  
 driver.get("https://www.rediff.com/");  
 driver.findElement(By.*xpath*("//\*[contains(@class,'signin')]")).click();  
 driver.findElement(By.*xpath*("//\*[@id='login1']")).sendKeys("Hello");  
 driver.findElement(By.*xpath*("//input[@id='password']")).sendKeys("pa$$123");  
 Thread.*sleep*(3000);  
 driver.findElement(By.*xpath*("//\*[@type='submit']")).click();  
 Thread.*sleep*(3000);  
 driver.close();  
 //rct-icon rct-icon-expand-close  
  
 }  
  
 @Test  
 public void test5() throws InterruptedException {  
 WebDriverManager.*chromedriver*().setup();  
 WebDriver driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(7, TimeUnit.*SECONDS*);  
 driver.get("https://demoqa.com/radio-button");  
 WebElement getTextFromTheSite = driver.findElement(By.*xpath*("//\*[@class='mb-3']"));  
 System.*out*.println(getTextFromTheSite.getText());  
 driver.findElement(By.*xpath*("//\*[@class='custom-control-label']")).click();  
 WebElement showMessage = driver.findElement(By.*xpath*("//p[contains(text(),'You have selected ')]"));  
 Thread.*sleep*(4000);  
 List<WebElement> answersToRadioButtons = driver.findElements(By.*xpath*("//div[starts-with(@class,'custom-control')]"));  
 for (WebElement answer : answersToRadioButtons) {  
 if (!answer.isSelected()) {  
 System.*out*.println("answer is selected : " + answer.getText());  
 } else {  
 System.*out*.println("answer is not selected : " + answer.getText());  
 }  
 }  
 Assert.*assertTrue*(showMessage.isDisplayed());  
 driver.close();  
 }  
  
 @Test  
 public void test6() throws InterruptedException {  
 WebDriverManager.*chromedriver*().setup();  
 WebDriver driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(7, TimeUnit.*SECONDS*);  
 driver.get("http://www.qaclickacademy.com/interview.php");  
 driver.findElement(By.*xpath*("//li[text()=' Selenium ']")).click();  
  
 driver.findElement(By.*xpath*("//ul[@class='responsive-tabs\_\_list']/li[1]/following-sibling::li[2]")).click();  
  
 System.*out*.println(driver.findElement(By.*xpath*(".//\*[@id='tablist1-tab2']/parent::ul")).getAttribute("role"));  
 }  
  
 @Test  
 public void test7() throws InterruptedException {  
 WebDriverManager.*chromedriver*().setup();  
 WebDriver driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(7, TimeUnit.*SECONDS*);  
 driver.get("https://www.spicejet.com/");  
 Select passengers = new Select(driver.findElement(By.*xpath*("//div[@id='divpaxinfo']")));  
 // s.selectByValue();  
 Select adults = new Select(driver.findElement(By.*xpath*("//\*[@id='ctl00\_mainContent\_ddl\_Adult']")));  
 adults.selectByValue("2");  
 Thread.*sleep*(3000);  
 driver.close();  
 }  
  
 @Test  
 public void test8() throws InterruptedException {  
 WebDriverManager.*chromedriver*().setup();  
 WebDriver driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(7, TimeUnit.*SECONDS*);  
 driver.get("https://opensource-demo.orangehrmlive.com/");  
 String userName = "Admin";  
 String password = "admin123";  
 WebElement userNameInputFiled = driver.findElement(By.*xpath*("//input[@id='txtUsername']"));  
 WebElement passwordInputFiled = driver.findElement(By.*xpath*("//input[@id='txtPassword']"));  
 WebElement loginButton = driver.findElement(By.*xpath*("//input[@type='submit']"));  
 userNameInputFiled.sendKeys(userName);  
 passwordInputFiled.sendKeys(password);  
 loginButton.click();  
 Thread.*sleep*(3000);  
 WebElement welcomeText = driver.findElement(By.*xpath*("//a[contains(text(),'Welcome')]"));  
 Assert.*assertTrue*(welcomeText.getText().contains(userName));  
 ////a[@class='panelTrigger activated-welcome']  
  
  
 }  
}

**sss**

package basicLocatorsPractice;  
  
import io.github.bonigarcia.wdm.ChromeDriverManager;  
import io.github.bonigarcia.wdm.WebDriverManager;  
import org.junit.Assert;  
import org.junit.Test;  
import org.openqa.selenium.By;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.WebElement;  
import org.openqa.selenium.chrome.ChromeDriver;  
  
import java.sql.Time;  
import java.util.concurrent.TimeUnit;  
  
public class Locators {  
 @Test  
 public void test1() throws InterruptedException {  
 //to set up the driver instead system.setProperty we do this:  
 WebDriverManager.*chromedriver*().setup();  
  
 WebDriver driver = new ChromeDriver();  
 //gi driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(10, TimeUnit.*SECONDS*);  
 driver.get("https://www.google.ru/");  
  
 WebElement searchInputField = driver.findElement(By.*name*("q"));  
 String searchCriteria = "apple";  
 searchInputField.sendKeys(searchCriteria);  
 driver.findElement(By.*name*("btnK")).click();  
 Thread.*sleep*(5000);  
 Assert.*assertTrue*("The title doesn't contain the search criteria. Expected: " + searchCriteria + ". Actual: " + driver.getTitle(), driver.getTitle().contains(searchCriteria));  
  
 driver.close();  
 }  
  
 @Test  
 public void test2() throws InterruptedException{  
 WebDriverManager.*chromedriver*().setup();  
  
 WebDriver driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(10, TimeUnit.*SECONDS*);  
  
 driver.get("https://www.google.com");  
 driver.findElement(By.*linkText*("Gmail")).click();  
 Thread.*sleep*(3000);  
 WebElement createAnAccountButton = driver.findElement(By.*partialLinkText*("Create an"));  
  
 Assert.*assertTrue*(createAnAccountButton.isDisplayed());  
 driver.close();  
  
 }  
 @Test  
 public void test3()throws InterruptedException{  
 WebDriverManager.*chromedriver*().setup();  
  
 WebDriver driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(10, TimeUnit.*SECONDS*);  
  
 driver.get("https://demostore.x-cart.com/");  
 driver.findElement(By.*linkText*("Contact us")).click();  
 Thread.*sleep*(3000);  
 WebElement headind = driver.findElement(By.*tagName*("h1"));  
  
 Assert.*assertTrue*(headind.isDisplayed());  
 driver.close();  
 }  
 @Test  
 public void test4() throws InterruptedException {  
 WebDriverManager.*chromedriver*().setup();  
  
 WebDriver driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(10, TimeUnit.*SECONDS*);  
  
 driver.get("https://www.salesforce.com/");  
 driver.findElement(By.*xpath*("//\*[@id='globalnavbar-header-container']/div[2]/div/div[5]/div/div/a")).click();  
 driver.findElement(By.*id*("username")).sendKeys("Hello");  
 Thread.*sleep*(3000);  
 driver.findElement(By.*id*("password")).sendKeys("pa$$1234");  
 Thread.*sleep*(3000);  
 driver.findElement(By.*cssSelector*("#Login")).click();  
 WebElement errorMessageToLogin = driver.findElement(By.*cssSelector*("#error"));  
 System.*out*.println(errorMessageToLogin.getText());  
 Assert.*assertTrue*(errorMessageToLogin.isDisplayed());  
  
 driver.close();  
  
 }  
  
 @Test  
 public void test5()throws InterruptedException{  
  
 WebDriverManager.*chromedriver*().setup();  
 WebDriver driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(10,TimeUnit.*SECONDS*);  
  
 driver.get("https://www.facebook.com/");  
 driver.findElement(By.*xpath*("//\*[@id='email']")).sendKeys("Hello");  
 driver.findElement(By.*cssSelector*("input[type='submit']")).click();  
  
 Thread.*sleep*(3000);  
 driver.findElement(By.*cssSelector*("input[type='text']")).sendKeys("phone number");  
 Thread.*sleep*(3000);  
 driver.findElement(By.*xpath*("//\*[@id='month']")).click();  
 driver.close();  
  
  
 }  
  
}

**fff**

public class TaskSmallFromSofia {  
 @Test  
 public void test1() throws InterruptedException {  
 WebDriverManager.*chromedriver*().setup();  
  
 WebDriver driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(10, TimeUnit.*SECONDS*);  
  
 driver.get("https://www.practiceselenium.com/");  
 driver.findElement(By.*partialLinkText*("let's Talk")).click();  
 Thread.*sleep*(3000);  
 WebElement nameInput = driver.findElement(By.*name*("name"));  
  
 Assert.*assertTrue*(nameInput.isDisplayed());  
 //driver.close();  
  
  
 }  
  
 @Test  
 public void test2() throws InterruptedException {  
 WebDriverManager.*chromedriver*().setup();  
  
 WebDriver driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(10, TimeUnit.*SECONDS*);  
  
 driver.get("https://www.amazon.com/");  
 driver.findElement(By.*linkText*("Sell")).click();  
 Thread.*sleep*(4000);  
 WebElement title = driver.findElement(By.*tagName*("h1"));  
 Assert.*assertTrue*(title.isDisplayed());  
 driver.findElement(By.*linkText*("Sign up")).click();  
 Thread.*sleep*(4000);  
 WebElement emailInput = driver.findElement(By.*id*("ap\_email"));  
 WebElement passwordInputField = driver.findElement(By.*id*("ap\_password"));  
 WebElement nextButton = driver.findElement(By.*id*("signInSubmit"));  
  
 emailInput.sendKeys("invalidEmail");  
 passwordInputField.sendKeys("pass123");  
 nextButton.click();  
 Thread.*sleep*(2000);  
 WebElement errorMessage = driver.findElement(By.*tagName*("h4"));  
 Assert.*assertTrue*(errorMessage.isDisplayed());  
 driver.close();  
  
  
 }  
  
 @Test  
 public void test3() throws InterruptedException {  
 WebDriverManager.*chromedriver*().setup();  
  
 WebDriver driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(10, TimeUnit.*SECONDS*);  
  
 driver.get("https://www.practiceselenium.com/");  
 Assert.*assertTrue*(driver.findElement(By.*tagName*("h1")).isDisplayed());  
  
 }  
  
 @Test  
 public void test4() throws InterruptedException {  
 WebDriverManager.*chromedriver*().setup();  
  
 WebDriver driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(10, TimeUnit.*SECONDS*);  
  
 driver.get("https://www.practiceselenium.com/");  
 driver.findElement(By.*linkText*("Check Out")).click();  
 Thread.*sleep*(3000);  
 driver.findElement(By.*linkText*("Cancel")).click();  
  
  
 driver.close();  
 }  
}

**ddd**

public class AlertsPractice {  
 WebDriver driver;  
  
 @Before  
 public void setUp() {  
 WebDriverManager.*chromedriver*().setup();  
 driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(5, TimeUnit.*SECONDS*);  
 }  
  
 @After  
 public void tearDown() {  
 driver.close();  
 driver.quit();  
 }  
  
 @Test  
 public void test1() throws InterruptedException {  
 driver.navigate().to("https://demoqa.com/alerts");  
 driver.findElement(By.*id*("alertButton")).click();  
 Alert alert = driver.switchTo().alert();  
 alert.accept();  
 driver.findElement(By.*id*("timerAlertButton")).click();  
 new WebDriverWait(driver, 7).until(ExpectedConditions.*alertIsPresent*());  
 alert.accept();  
 driver.findElement(By.*id*("confirmButton")).click();  
 alert.dismiss();  
  
 String resultText = driver.findElement(By.*id*("confirmResult")).getText();  
  
 Assert.*assertEquals*("You selected Cancel", resultText);  
  
 driver.findElement(By.*id*("promtButton")).click();  
 System.*out*.println("The text of alert is: " + alert.getText());  
  
 alert.sendKeys("Baha");  
 alert.accept();  
 String promptTextResult = driver.findElement(By.*id*("promtButton")).getText();  
 Assert.*assertTrue*(promptTextResult.contains("Baha"));  
 }  
  
 @Test  
 public void test2() {  
 //Task #11. Navigate to https://chercher.tech/practice/practice-pop-ups-selenium-webdriver  
 // 2. Double click on the button “Double Click Me”  
 // 3. Verify the text of the alert contains “You double clicked me!!!”  
 // 4. Click ok on the alert  
 // 5. Click on the Promptbutton  
 // 6. Verify the text of the alert equals to “I am prompt”  
 // 7. Pass your name to the prompt  
 // 8.Clickon Confirmation box  
 // 9. Verify the text of the alert equals to “I am confirm”  
 // 10. Cancel the aler  
 driver.navigate().to("https://chercher.tech/practice/practice-pop-ups-selenium-webdriver");  
 WebElement doubleClickElem = driver.findElement(By.*xpath*("//input[@value='Double Click Me']"));  
 Actions actions = new Actions(driver);  
 actions.doubleClick(doubleClickElem).perform();  
 Alert alert = driver.switchTo().alert();  
 String alertText = alert.getText();  
 Assert.*assertTrue*(alert.getText().contains("You double clicked me!"));  
 alert.accept();  
 driver.findElement(By.*name*("prompt")).click();  
 Assert.*assertEquals*("I am prompt", alert.getText());  
 alert.sendKeys("DevX");  
 alert.accept();  
  
 driver.findElement(By.*xpath*("//\*[@name='confirmation']"));  
 Assert.*assertEquals*("I am confirm",alert.getText());  
 alert.dismiss();  
  
 }  
}

**eee**

public class Iframes {  
 WebDriver driver;  
  
 @Before  
 public void setUp() {  
 WebDriverManager.*chromedriver*().setup();  
 driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(5, TimeUnit.*SECONDS*);  
 }  
  
 @After  
 public void tearDown() {  
 driver.close();  
 driver.quit();  
 }  
 @Test  
 public void test1(){  
  
 driver.navigate().to("https://demoqa.com/frames");  
  
 driver.switchTo().frame("frame1");  
 WebElement textOfFrame1 = driver.findElement(By.*tagName*("h1"));  
 Assert.*assertTrue*(textOfFrame1.isDisplayed());  
  
 //driver.switchTo().defaultContent();  
  
 driver.switchTo().frame(1);  
 WebElement textOfFrame2 = driver.findElement(By.*id*("sampleHeading"));  
  
 Assert.*assertEquals*("This is a sample page",textOfFrame2.getText());  
  
 }  
 @Test  
 public void test2(){  
 driver.navigate().to("https://allwebco-templates.com/support/S\_script\_IFrame.htm");  
  
 driver.switchTo().frame(driver.findElement(By.*xpath*("//iframe[@class='framesample']")));  
  
 System.*out*.println(driver.findElement(By.*tagName*("div")).getText());  
  
 driver.switchTo().defaultContent();  
  
 driver.findElement(By.*xpath*("//a[text()='Free website scripts']")).click();  
  
 Assert.*assertTrue*(driver.findElement(By.*xpath*("//h1[@class='title titlespad hidemobile610']")).isDisplayed());  
 }  
 @Test  
 public void test3(){  
 driver.navigate().to("https://demoqa.com/nestedframes");  
  
 driver.switchTo().frame("frame1");  
 driver.switchTo().frame(0);  
 Assert.*assertEquals*("Child Iframe",driver.findElement(By.*tagName*("p")).getText());  
  
 driver.switchTo().parentFrame();  
 Assert.*assertEquals*("Parent frame",driver.findElement(By.*tagName*("body")).getText());  
  
 driver.switchTo().defaultContent();  
 }  
  
}

**ooo**

public class JavaScriptPractice {  
 WebDriver driver;  
  
 @Before  
 public void setUp() {  
 WebDriverManager.*chromedriver*().setup();  
 driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(5, TimeUnit.*SECONDS*);  
 }  
  
 @Test  
 public void testSofia() {  
 driver.navigate().to("https://opensource-demo.orangehrmlive.com/");  
 JavascriptExecutor js = (JavascriptExecutor) driver;  
 WebDriverWait wait = new WebDriverWait(driver, 3);  
 WebElement usenameInput = driver.findElement(By.*id*("txtUsername"));  
 WebElement passwordInput = driver.findElement(By.*id*("txtPassword"));  
 WebElement logInButton = driver.findElement(By.*id*("btnLogin"));  
 js.executeScript("arguments[0].setAttribute('style', arguments[1]);", usenameInput, "border: 2px solid red");  
 usenameInput.sendKeys("Admin");  
 js.executeScript("arguments[0].setAttribute('style', arguments[1]);", passwordInput, "border: 2px solid red");  
 passwordInput.sendKeys("admin123");  
 js.executeScript("arguments[0].setAttribute('style', arguments[1]);", logInButton, "border: 2px solid red");  
 logInButton.click();  
 WebElement assignLeave = driver.findElement(By.*linkText*("Assign Leave"));  
 js.executeScript("arguments[0].setAttribute('style', arguments[1]);", assignLeave, "border: 2px solid red");  
 assignLeave.click();  
 WebElement employeeName = driver.findElement(By.*id*("assignleave\_txtEmployee\_empName"));  
 js.executeScript("arguments[0].setAttribute('style', arguments[1]);", employeeName, "border: 2px solid red");  
 employeeName.sendKeys("Fiona Grace");  
 //WebElement employeeNameFromList = driver.findElement(By.xpath("//strong[text()='Fiona Grac']"));  
 //js.executeScript("arguments[0].setAttribute('style', arguments[1]);", employeeNameFromList, "border: 2px solid red");  
 //employeeNameFromList.click();  
 WebElement leaveType = driver.findElement(By.*id*("assignleave\_txtLeaveType"));  
 js.executeScript("arguments[0].setAttribute('style', arguments[1]);", leaveType, "border: 2px solid red");  
 Select dropdown = new Select(leaveType);  
 dropdown.selectByValue("1");  
 WebElement fromDate = driver.findElement(By.*id*("assignleave\_txtFromDate"));  
 js.executeScript("arguments[0].setAttribute('style', arguments[1]);", fromDate, "border: 2px solid red");  
 fromDate.click();  
 //  
// List<WebElement> allDates = driver.findElements(By.xpath("//td[@data-handler='selectDay']"));  
// //create local date time  
// int date = LocalDateTime.now().getDayOfMonth();  
// for (WebElement day : allDates) {  
// if (Integer.parseInt(day.getText()) == date)  
// driver.findElement(By.xpath("//a[text()=" + date + "]"));  
// }  
 //  
 // driver.findElement(By.xpath("//a[text()=29]")).click();  
 WebElement toDate = driver.findElement(By.*id*("assignleave\_txtToDate"));  
 js.executeScript("arguments[0].setAttribute('style', arguments[1]);", toDate, "border: 2px solid red");  
 toDate.click();  
 driver.findElement(By.*xpath*("//a[text()=29]")).click();  
 WebElement assignButton = driver.findElement(By.*id*("assignBtn"));  
 js.executeScript("arguments[0].setAttribute('style', arguments[1]);", assignButton, "border: 2px solid red");  
 assignButton.click();  
 wait.until(ExpectedConditions.*presenceOfElementLocated*(By.*xpath*("//p[text()='Employee does not have sufficient leave balance for leave request.']")));  
 WebElement errorMessage = driver.findElement(By.*xpath*("//p[text()='Employee does not have sufficient leave balance for leave request.']"));  
 Assert.*assertTrue*(errorMessage.isDisplayed());  
 }  
 //if .clear() doesn't work on the input field u can use:  
 // element.sendKeys(Keys.CONTROL+"a"+Keys.DELETE);  
  
  
 @After  
 public void tearDown() {  
 driver.close();  
 driver.quit();  
 }  
  
 @Test  
 public void test1() throws InterruptedException {  
 JavascriptExecutor js = (JavascriptExecutor) driver;  
 js.executeScript("window.location='https://www.etsy.com/'");  
 WebElement signInButton = driver.findElement(By.*xpath*("//nav[@aria-label='Main navigation']//button[contains(text(),'Sign in')]"));  
 js.executeScript("arguments[0].setAttribute('style',arguments[1]);", signInButton, "border: 2px solid red");  
 Thread.*sleep*(2000);  
 js.executeScript("arguments[0].click();", signInButton);  
 WebElement emailInput = driver.findElement(By.*id*("join\_neu\_email\_field"));  
 js.executeScript("arguments[0].setAttribute('style',arguments[1]);", signInButton, "border: 2px solid red");  
 Faker faker = new Faker();  
 String fakeEmail = faker.bothify("?????##@gmail.com");  
  
 js.executeScript("arguments[0].setAttribute('value', arguments[1]);", emailInput, fakeEmail);  
 WebElement passwordInput = driver.findElement(By.*id*("join\_neu\_password\_field"));  
 js.executeScript("arguments[0].setAttribute('style', arguments[1]);", passwordInput, "border: 2px solid red");  
 js.executeScript("arguments[0].setAttribute('value', arguments[1]);", passwordInput, "abs123");  
  
 WebElement sighinButton2 = driver.findElement(By.*xpath*("//button[@value='sign-in']"));  
 js.executeScript("arguments[0].setAttribute('style', arguments[1]);", sighinButton2, "border: 2px solid red");  
 Thread.*sleep*(2000);  
 js.executeScript("arguments[0].click();", sighinButton2);  
 }  
  
 @Test  
 public void test2() throws InterruptedException{  
 WebDriverWait wait = new WebDriverWait(driver,7);  
 JavascriptExecutor js = (JavascriptExecutor) driver;  
 js.executeScript("window.location='https://opensource-demo.orangehrmlive.com'");  
 WebElement inputUserName = driver.findElement(By.*xpath*("//input[@id='txtUsername']"));  
 js.executeScript("arguments[0].setAttribute('style',arguments[1]);", inputUserName, "border: 2px solid red");  
 inputUserName.sendKeys("Admin");  
 WebElement inputPassword = driver.findElement(By.*id*("txtPassword"));  
 js.executeScript("arguments[0].setAttribute('style',arguments[1]);", inputPassword, "border: 2px solid red");  
 inputPassword.sendKeys("admin123");  
 driver.findElement(By.*id*("btnLogin")).click();  
  
 js.executeScript("arguments[0].click();", driver.findElement(By.*xpath*("//img[@src='/webres\_5ebd1457b45137.49759927/orangehrmLeavePlugin/images/ApplyLeave.png']")));  
 WebElement employeeEnter = driver.findElement(By.*xpath*("//input[@id='assignleave\_txtEmployee\_empName']"));  
 js.executeScript("arguments[0].setAttribute('style', arguments[1]);", employeeEnter, "border: 2px solid red");  
 employeeEnter.sendKeys("Fiona Grace");  
 employeeEnter.click();  
 // WebElement fionaHoverOver = driver.findElement(By.xpath("//li[@class='ac\_even ac\_over']//strong[contains(text(),'Fiona Grac')]"));  
 // wait.until(ExpectedConditions.visibilityOf(fionaHoverOver));  
 // Thread.sleep(3000);  
 // fionaHoverOver.click();  
//  
 WebElement selectElement = driver.findElement(By.*xpath*("//select[@id='assignleave\_txtLeaveType']"));  
 js.executeScript("arguments[0].setAttribute('style', arguments[1]);",selectElement,"border: 2px solid red");  
 Select select = new Select(selectElement);  
 select.selectByVisibleText("Vacation US");  
 WebElement enterCalendar = driver.findElement(By.*xpath*("//input[@id='assignleave\_txtFromDate']"));  
 js.executeScript("arguments[0].setAttribute('style', arguments[1]);", enterCalendar, "border: 2px solid red");  
 WebElement enterCalendarCheckOut = driver.findElement(By.*xpath*("//input[@id='assignleave\_txtToDate']"));  
 enterCalendar.click();  
 int date = LocalDateTime.*now*().getDayOfMonth();  
 driver.findElement(By.*xpath*("//a[contains(text(),'29')]")).click();  
 js.executeScript("arguments[0].setAttribute('style', arguments[1]);", enterCalendarCheckOut, "border: 2px solid red");  
 enterCalendarCheckOut.click();  
 driver.findElement(By.*xpath*("//a[contains(text(),'29')]")).click();  
 WebElement area = driver.findElement(By.*xpath*("//textarea[@id='assignleave\_txtComment']"));  
 Thread.*sleep*(3000);  
 js.executeScript("arguments[0].setAttribute('style', arguments[1]);", area, "border: 2px solid red");  
 area.sendKeys("My comment!!!!");  
 WebElement assign = driver.findElement(By.*xpath*("//input[@id='assignBtn']"));  
 Thread.*sleep*(3000);  
 js.executeScript("arguments[0].click();", assign);  
 Thread.*sleep*(3000);  
   
 WebElement el = wait.until(ExpectedConditions.*presenceOfElementLocated*(By.*xpath*("//div[contains(text(),‘Failed to Assign’)]")));  
 Assert.*assertTrue*(el.getText().contains("Failed to Assign"));  
  
 }  
  
  
}

**iii**

public class DropDowns {  
 WebDriver driver;  
  
 @Before  
 public void setUp() {  
 WebDriverManager.*chromedriver*().setup();  
 driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(5, TimeUnit.*SECONDS*);  
  
 }  
  
 @After  
 public void tearDown() {  
 driver.close();  
 driver.quit();  
 }  
  
 @Test  
 public void test1() {  
  
 driver.navigate().to("https://www.devxschool.com/");  
  
 WebElement enrollNowButton = driver.findElement(By.*linkText*("Enroll Now"));  
 enrollNowButton.click();  
 WebElement dropDown = driver.findElement(By.*xpath*("//select[@id='form-field-ads']"));  
 Select selectElement = new Select(dropDown);  
 List<WebElement> selectedOptions = selectElement.getAllSelectedOptions();  
 for (WebElement option : selectedOptions) {  
 System.*out*.println(option.getText());  
 }  
 Assert.*assertTrue*(selectedOptions.size() == 1 &&  
 selectedOptions.get(0).getText().equals("From a friend"));  
  
 //selectElement.selectByVisibleText("Instagram");  
 //selectElement.selectByIndex(2);  
 selectElement.selectByValue("Instagram");  
  
 selectedOptions = selectElement.getAllSelectedOptions();  
 Assert.*assertTrue*(selectedOptions.size() == 1 &&  
 selectedOptions.get(0).getText().equals("Instagram"));  
 }  
  
  
 @Test  
 public void test2() {  
 driver.navigate().to("https://www.expedia.com/");  
 //find cruises button  
 WebElement cruisesButton = driver.findElement(By.*id*("tab-cruise-tab-hp"));  
 //click on it  
 cruisesButton.click();  
 //find select element  
 WebElement selectElement = driver.findElement(By.*id*("cruise-destination-hp-cruise"));  
 //create object of Select class  
 Select selectDestination = new Select(selectElement);  
 //select Alaska by value -->> value="alaska"  
 selectDestination.selectByValue("alaska");  
 //verify Alaska is selected  
 Assert.*assertTrue*(selectDestination.getAllSelectedOptions().get(0).getText().equals("Alaska"));  
 //select Africa by visible text -->> visible text is Africa  
 selectDestination.selectByVisibleText("Africa");  
 Assert.*assertTrue*(selectDestination.getAllSelectedOptions().get(0).getText().equals("Africa"));  
 //select Mexico by index -->> index of Mexico is 3  
 selectDestination.selectByIndex(3);  
 Assert.*assertTrue*(selectDestination.getAllSelectedOptions().get(0).getText().equals("Mexico"));  
 //print out all possible options  
 for (WebElement option : selectDestination.getOptions())  
 System.*out*.println(option.getText());  
 }  
  
 @Test  
 public void test3() {  
 driver.navigate().to("https://www.jquery-az.com/boots/demo.php?ex=63.0\_2");  
 //find select element  
 WebElement selectElement = driver.findElement(By.*id*("option-droup-demo"));  
 //create an object of Select class  
 Select dropdown = new Select(selectElement);  
 //dropdown.deselectAll();  
 //select Java  
 dropdown.selectByVisibleText("Java");  
 dropdown.selectByVisibleText("Oracle");  
 //deselect default values  
 dropdown.deselectByVisibleText("HTML");  
 dropdown.deselectByVisibleText("CSS");  
 List<WebElement> mySelectedOptions = dropdown.getAllSelectedOptions();  
 for (WebElement selectedOption : mySelectedOptions) {  
 Assert.*assertTrue*(selectedOption.getText().equals("Java") ||  
 selectedOption.getText().equals("Oracle"));  
 }  
 }  
  
 @Test  
 public void test4() {  
 driver.navigate().to("https://www.facebook.com/");  
 WebElement dropdownMonth = driver.findElement(By.*xpath*("//select[@id='month']"));  
 Select select = new Select(dropdownMonth);  
 //List<WebElement> list = select.getAllSelectedOptions();  
 Assert.*assertTrue*(  
 select.getAllSelectedOptions().get(0).getText().equals("Jul"));  
 select.selectByIndex(1);  
 Assert.*assertTrue*(select.getAllSelectedOptions().get(0).getText().equals("Jan"));  
 for (WebElement li : select.getOptions()) {  
 System.*out*.println(li.getText());  
 }  
 }  
  
 @Test  
 public void test5() {  
 driver.navigate().to("https://www.jquery-az.com/boots/demo.php?ex=63.0\_2");  
 WebElement outputCss = driver.findElement(By.*id*("option-droup-demo"));  
 Select select = new Select(outputCss);  
 select.selectByValue("Java");  
 select.selectByValue("Python");  
 select.deselectByValue("HTML");  
 select.deselectByValue("CSS");  
 List<WebElement>listWithDropDowns = select.getAllSelectedOptions();  
 for(WebElement option : listWithDropDowns){  
 Assert.*assertTrue*(option.getText().equals("Java") || option.getText().equals("Python"));  
 }  
  
 }  
  
}

**[[[**

public class PageNavigation {  
 WebDriver driver;  
  
 @Before  
 public void setUp() {  
 WebDriverManager.*chromedriver*().setup();  
 driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(5, TimeUnit.*SECONDS*);  
  
 }  
  
 @Test  
 public void test1() throws InterruptedException{  
 //does exactly the same as driver.get() - no difference between them  
 //driver.navigate().to("http://www.practiceselenium.com/");  
 driver.get("http://www.practiceselenium.com/welcome.html");  
 WebElement letsTalkTeaLink = driver.findElement(By.*xpath*("//li[@style='width: '][4]"));  
 letsTalkTeaLink.click();  
 //element.getAttribute("attribute name") -> return the value of that attribute  
 letsTalkTeaLink = driver.findElement(By.*xpath*("//li[@style='width: '][4]"));  
 String attribute = letsTalkTeaLink.getAttribute("class");  
 Assert.*assertTrue*("Let's talk tea must be active but it is not",attribute.equals("active"));  
 driver.navigate().back();  
 WebElement welcomeLink = driver.findElement(By.*xpath*("//li[@style='width: '][1]"));  
 Assert.*assertTrue*("Welcome link must be active but it is not", welcomeLink.getAttribute("class").equals("active"));  
 driver.navigate().forward();  
 letsTalkTeaLink = driver.findElement(By.*xpath*("//li[@style='width: '][4]"));  
 Assert.*assertTrue*("Let's talk tea must be active but it is not",letsTalkTeaLink.getAttribute("class").equals("active"));  
 WebElement nameInputField = driver.findElement(By.*xpath*("//input[@name='name']"));  
 Assert.*assertTrue*(nameInputField.isDisplayed());  
 driver.navigate().refresh();  
 //StaleElementException - element is old, not fresh, you have to find it again in order to interact with it  
 try{  
 nameInputField.sendKeys("My name is here");  
 }catch(StaleElementReferenceException e){  
 nameInputField = driver.findElement(By.*xpath*("//input[@name='name']"));  
 nameInputField.sendKeys("My name is here");  
 }  
 Thread.*sleep*(3000);  
 }  
  
 @After  
 public void tearDown() {  
 driver.close();  
 driver.quit();  
 }  
}

**www**

public class PracticeWaits {  
 WebDriver driver;  
  
 @Before  
 public void setUp(){  
 WebDriverManager.*chromedriver*().setup();  
 driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(3, TimeUnit.*SECONDS*);  
  
  
  
  
 }  
  
 @Test  
 public void test1() {  
  
 driver.get("https://www.etsy.com/");  
 driver.findElement(By.*xpath*("//button[@class='wt-btn wt-btn--small wt-btn--transparent wt-mr-xs-1 inline-overlay-trigger signin-header-action select-signin']")).click();  
 WebDriverWait wait = new WebDriverWait(driver,4);  
 wait.until(ExpectedConditions.*presenceOfElementLocated*(By.*xpath*("//input[@id='join\_neu\_email\_field']")));  
 WebElement emailInputField = driver.findElement(By.*xpath*("//input[@id='join\_neu\_email\_field']"));  
 emailInputField.sendKeys("someemail");  
 // Assert.assertTrue("Email input is not displayed", emailInputField.isDisplayed());  
 }  
  
 @Test  
 public void test2() {  
 driver.get("https://demoqa.com/dynamic-properties");  
 WebDriverWait wait = new WebDriverWait(driver,10);  
 wait.until(ExpectedConditions.*presenceOfElementLocated*(By.*xpath*("//button[@id='visibleAfter']")));  
 }  
  
 @Test  
 public void fileUpload(){  
 driver.get("https://demoqa.com/upload-download");  
 driver.findElement(By.*xpath*("//input[@id='uploadFile']")).sendKeys("/Users/mac/Desktop/DevX-School-Logo copy.png");  
  
  
  
 }  
 @Test  
 public void sendToGoogleImages(){  
 driver.get("https://images.google.com/");  
 driver.findElement(By.*xpath*("//div[@aria-label='Search by image']")).click();  
 WebDriverWait wait = new WebDriverWait(driver,7);  
 wait.until(ExpectedConditions.*presenceOfElementLocated*(By.*xpath*("//input[@id='awyMjb']"))).sendKeys("/Users/mac/Desktop/DevX-School-Logo copy.png");  
 }  
 @After  
 public void tearDown(){  
 //driver.close();  
 }  
  
}

**qqq**

public class AdvancedMoudeInteractions {  
 WebDriver driver;  
  
 @Before  
 public void setUp() {  
 WebDriverManager.*chromedriver*().setup();  
 driver = new ChromeDriver();  
 driver.manage().window().maximize();  
 driver.manage().timeouts().implicitlyWait(5, TimeUnit.*SECONDS*);  
 }  
  
 @After  
 public void tearDown() {  
 driver.close();  
 driver.quit();  
 }  
  
 @Test  
 public void test1() {  
 driver.navigate().to("https://demoqa.com/buttons");  
 //first find the elements  
 WebElement doubleClickButton = driver.findElement(By.*id*("doubleClickBtn"));  
 WebElement rightClickButton = driver.findElement(By.*id*("rightClickBtn"));  
 //create an obj of actions class  
 Actions actions = new Actions(driver);  
 //action to happen on element we always finish with .perform()->make the action happens  
 actions.doubleClick(doubleClickButton).perform();  
  
 Assert.*assertTrue*("double click message isn't displayed", driver.findElement(By.*id*("doubleClickMessage")).isDisplayed());  
  
 //now right click  
 actions.contextClick(rightClickButton).perform();  
  
 Assert.*assertTrue*("right click message isn't displayed", driver.findElement(By.*id*("rightClickMessage")).isDisplayed());  
  
  
 }  
  
 @Test  
 public void test2() throws InterruptedException {  
 driver.navigate().to("https://demoqa.com/tool-tips");  
 WebElement hoverButoon = driver.findElement(By.*id*("toolTipButton"));  
 Actions actions = new Actions(driver);  
  
 actions.moveToElement(hoverButoon).perform();  
 WebDriverWait wait = new WebDriverWait(driver, 3);  
 boolean attributeAppeared = wait.until(ExpectedConditions.*attributeToBe*(hoverButoon, "aria-describedby", "buttonToolTip"));  
 String attribute = hoverButoon.getAttribute("aria-describedby");//"buttonToolTip"  
  
 Assert.*assertEquals*("buttonToolTip", attribute);  
 actions.moveToElement(driver.findElement(By.*xpath*("//div[contains(text(),'Book Store Application')]"))).perform();  
 Thread.*sleep*(3000);  
 }  
  
 @Test  
 public void test3() throws InterruptedException {  
 driver.navigate().to("https://www.etsy.com/");  
 //i need to scroll to subscribe input field type email hit subscribe  
 WebElement subscribeInput = driver.findElement(By.*id*("email-list-signup-email-input"));  
 Actions actions = new Actions(driver);  
 actions.moveToElement(subscribeInput).click().sendKeys("devxschool@gmail.com" + Keys.*ENTER*).perform();  
 Thread.*sleep*(3000);  
 }  
  
 @Test  
 public void test4() throws InterruptedException {  
 driver.navigate().to("https://demoqa.com/slider");  
 WebElement slider = driver.findElement(By.*xpath*("//input[@type='range']"));  
 Actions actions = new Actions(driver);  
 actions.clickAndHold(slider).moveByOffset(20, 0).release().perform();  
 Thread.*sleep*(3000);  
 WebElement sliderValue = driver.findElement(By.*id*("sliderValue"));  
 Assert.*assertTrue*(Integer.*parseInt*(sliderValue.getAttribute("value")) > 25);  
 }  
  
 @Test  
 public void test5() throws InterruptedException {  
 driver.navigate().to("https://demoqa.com/droppable");  
 WebElement source = driver.findElement(By.*id*("draggable"));  
 WebElement target = driver.findElement(By.*id*("draggable"));  
  
 Actions actions = new Actions(driver);  
 actions.dragAndDrop(source, target).perform();  
 Thread.*sleep*(3000);  
 Assert.*assertTrue*(target.getText().contains("Dropped!"));  
  
 }  
  
 @Test  
 public void test6() throws InterruptedException {  
  
 driver.navigate().to("https://demoqa.com/droppable");  
 driver.findElement(By.*id*("droppableExample-tab-preventPropogation")).click();  
 WebElement source = driver.findElement(By.*xpath*("//div[@id='dragBox']"));  
 WebElement target = driver.findElement(By.*id*("notGreedyInnerDropBox"));  
 Actions actions = new Actions(driver);  
 actions.dragAndDrop(source, target).perform();  
 Thread.*sleep*(4000);  
  
 Assert.*assertTrue*(target.getText().contains("Dropped!"));  
 Assert.*assertTrue*(driver.findElement(By.*xpath*("//div[@id='notGreedyDropBox']")).getText().contains("Dropped!"));  
  
 }  
  
 @Test  
 public void test7() {  
 //1.Go to https://demoqa.com/progress-bar  
 // 2.Click start  
 // 3.Wait until the value reaches 100–Use EXPLICIT WAIT  
 // 4.Verify the button has changed to Reset  
 // 5.Click Reset  
 // 6.Verify the progress bar has reset to 0;  
 driver.navigate().to("https://demoqa.com/progress-bar");  
 driver.findElement(By.*id*("startStopButton")).click();  
 WebElement elemen = driver.findElement(By.*xpath*("//div[@role='progressbar']"));  
 WebDriverWait wait = new WebDriverWait(driver, 14);  
 wait.until(ExpectedConditions.*attributeToBe*(elemen, "aria-valuenow", "100"));  
 // wait.until(ExpectedConditions.presenceOfElementLocated(By.id("resetButton")));  
 driver.findElement(By.*id*("resetButton")).click();  
 // WebElement elemen = driver.findElement(By.xpath("//div[@role='progressbar']"));  
  
 Assert.*assertTrue*(Integer.*parseInt*(elemen.getAttribute("aria-valuenow")) == 0);  
 }  
  
 @Test  
 public void test8() {  
 //1. Go to https://demoqa.com/menu#  
 // 2. Hover over Main Item 2  
 // 3. Hover over SUBSUB LIST  
 // 4. Click on Sub Sub item  
 driver.navigate().to("https://demoqa.com/menu#");  
 WebElement mainItem = driver.findElement(By.*xpath*("//a[contains(text(),'Main Item 2')]"));  
 Actions actions = new Actions(driver);  
 actions.moveToElement(mainItem).perform();  
 actions.moveToElement(driver.findElement(By.*xpath*("//a[contains(text(),'SUB SUB LIST »')]"))).perform();  
 driver.findElement(By.*xpath*("//a[contains(text(),'Sub Sub Item 1')]")).click();  
  
 }  
  
 @Test  
 public void test9() throws InterruptedException {  
 //1.Go to https://www.seleniumeasy.com/test/drag-and-drop-demo.html  
 // 2.Drag and drop 1st and 3rd Draggables to Drop Here object  
 // 3.Verify your Dropped item list contains correct values  
 driver.navigate().to("https://www.seleniumeasy.com/test/drag-and-drop-demo.html");  
 WebElement draggable1 = driver.findElement(By.*xpath*("//div[@id='todrag']//span[contains(text(),'Draggable 1')]"));  
 WebElement draggable3 = driver.findElement(By.*xpath*("//div[@id='todrag']//span[contains(text(),'Draggable 3')]"));  
 WebElement target = driver.findElement(By.*xpath*("//div[@id='mydropzone']"));  
 Actions actions = new Actions(driver);  
 actions.moveToElement(target);  
 actions.moveToElement(target,10, 0);  
  
 actions.dragAndDrop(draggable1, target).build().perform();  
 actions.moveToElement(target,10, 0);  
 actions.moveToElement(target);  
  
 Thread.*sleep*(4000);  
 actions.moveToElement(target);  
 actions.moveToElement(target,10, 0);  
  
 actions.dragAndDrop(draggable3, target).build().perform();  
 actions.moveToElement(target,10, 0);  
  
 actions.moveToElement(target);  
  
 Thread.*sleep*(4000);  
 WebElement droppedListBox = driver.findElement(By.*xpath*("//div[@id='droppedlist']"));  
// actions.clickAndHold(draggable1);  
// actions.dragAndDropBy(draggable1, 10,0).release().perform();  
// Thread.sleep(4000);  
// actions.clickAndHold(draggable3);  
// actions.dragAndDropBy(draggable3, 10,0).release().perform();  
// Thread.sleep(4000);  
 Assert.*assertTrue*(droppedListBox.getAttribute("style").equals("visibility: visible;"));  
 //Assert.assertTrue(droppedListBox.getAttribute("style").equals("style"));  
 ////div[@id='droppedlist']//\*[contains(text(),'Draggable 1')]  
 // Assert.assertTrue(driver.findElement(By.xpath("//div[@id='droppedlist']//\*[contains(text(),'Draggable 1')]")).isDisplayed());  
 // Assert.assertTrue(driver.findElement(By.xpath("//div[@id='droppedlist']//\*[contains(text(),'Draggable 3')]")).isDisplayed());  
 //Assert.assertTrue(driver.findElement(By.xpath("//div[@style='visibility: visible;']")).isDisplayed());  
 //Assert.assertEquals();  
 }  
 @Test  
 public void test10() throws InterruptedException{  
 driver.navigate().to("https://www.globalsqa.com/demo-site/draganddrop/");  
 WebElement source1 = driver.findElement(By.*xpath*("//li[@class='ui-widget-content ui-corner-tr ui-draggable ui-draggable-handle']//img[@alt='The peaks of High Tatras']"));  
 WebElement source2 = driver.findElement(By.*xpath*("//li[@class='ui-widget-content ui-corner-tr ui-draggable ui-draggable-handle']//img[@alt='The chalet at the Green mountain lake']"));  
 WebElement target = driver.findElement(By.*xpath*("//div[@id='trash']"));  
 Actions actions = new Actions(driver);  
 actions.dragAndDrop(source1,target);  
 Thread.*sleep*(4000);  
 actions.dragAndDrop(source2,target);  
 Thread.*sleep*(4000);  
  
 List<WebElement>listWithItems = driver.findElements(By.*xpath*("//div[@id='trash']"));  
 for(WebElement element : listWithItems){  
 Assert.*assertTrue*(element.getText().contains("The peaks of High Tatras") && element.getText().contains("The chalet at the Green mountain lake"));  
 }  
  
  
 }  
  
}