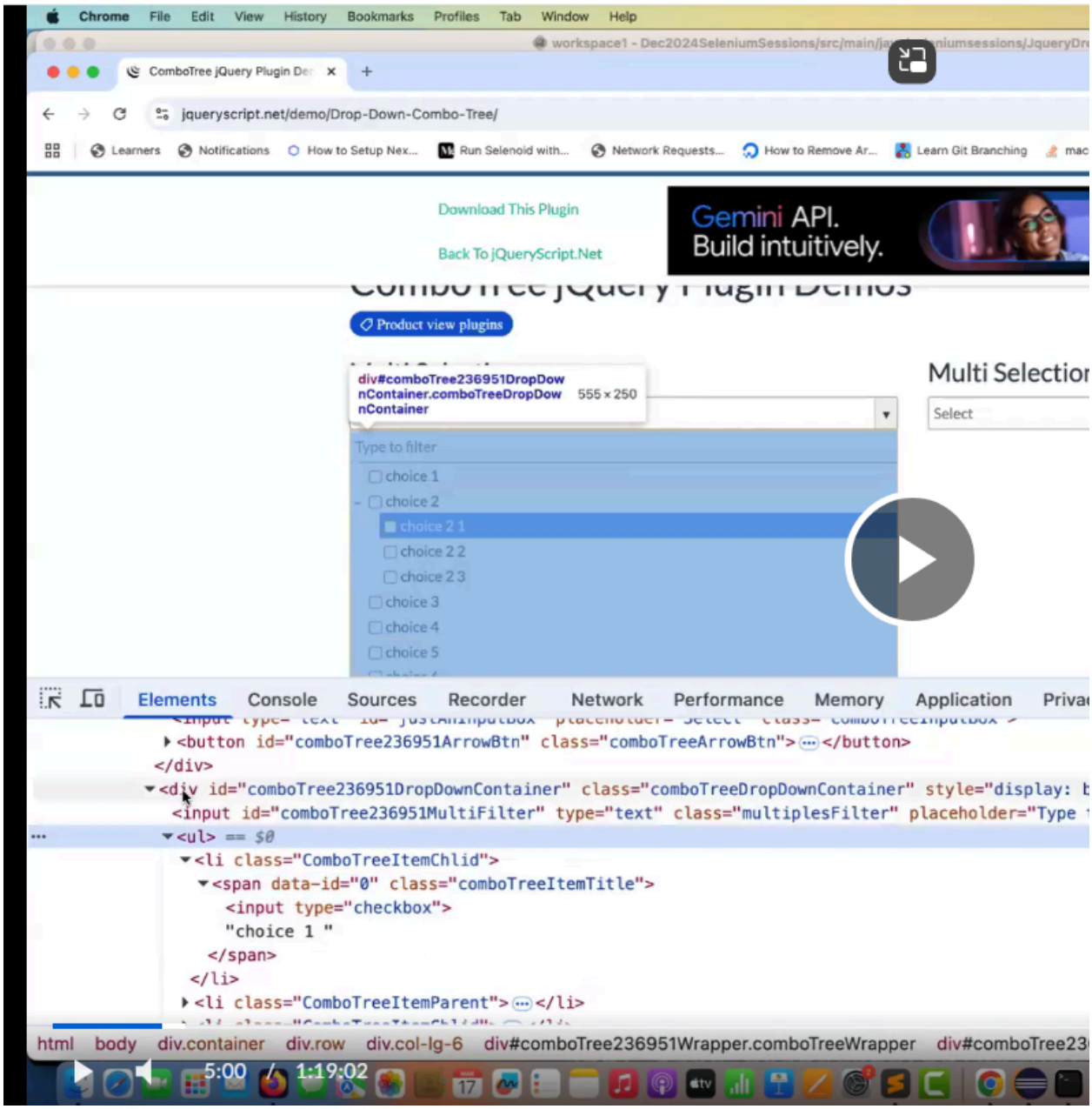
jquery drop down-

The **choice1** is under span not input. If you see closing bracket without slash, it also means that tag is closed.



Checking if choices are counted properly from first drop down-

```
ElementUtil.java   JqueryDropDownHandle.java   JqueryDropDownConcept.java
1  package seleniumsessions;
2
3  import java.util.List;
4
5  import org.openqa.selenium.By;
6  import org.openqa.selenium.WebDriver;
7  import org.openqa.selenium.WebElement;
8  import org.openqa.selenium.chrome.ChromeDriver;
9
10 public class JqueryDropDownHandle {
11
12     static WebDriver driver;
13
14     public static void main(String[] args) throws InterruptedException {
15
16         driver = new ChromeDriver();
17         driver.get("https://www.jqueryscript.net/demo/Drop-Down-Combo-Tree");
18
19         Thread.sleep(2000);
20
21         driver.findElement(By.id("justAnInputBox")).click();
22
23         Thread.sleep(2000);
24
25         String xpath = "(//div[@class='comboTreeDropDownContainer'])[1]//li/span[@class='comboTreeItemTitle']";
26
27         List<WebElement> choices = driver.findElements(By.xpath(xpath));
28         System.out.println(choices.size());
29
30
31
32     }
33
34 }
35
```

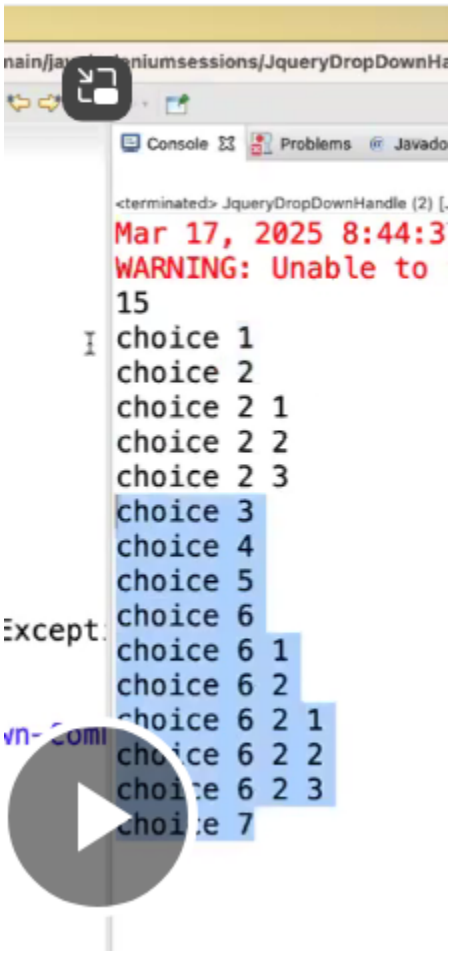
Clear picture of line 17-

```
driver.get("https://www.jqueryscript.net/demo/Drop-Down-Combo-Tree");
```

```
/seleniumsessions/JqueryDr
Console   Problems
<terminated> JqueryDropDown
Mar 17, 2025 8
WARNING: Unabl
15
```

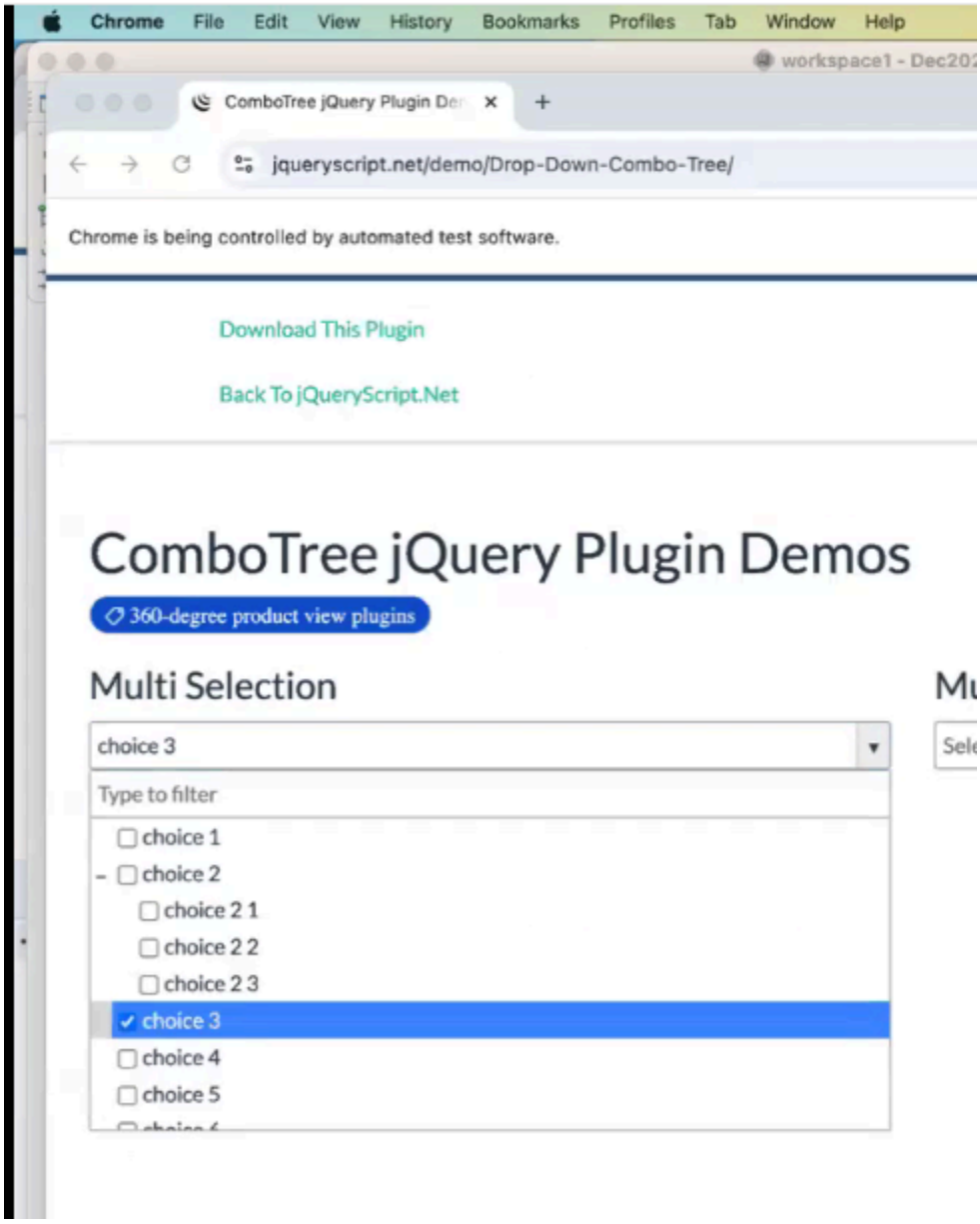
Print all the choices-

```
29
30          for(WebElement e : choices) {
31              String text = e.getText();
32              System.out.println(text);
33          }
34
35      }
```



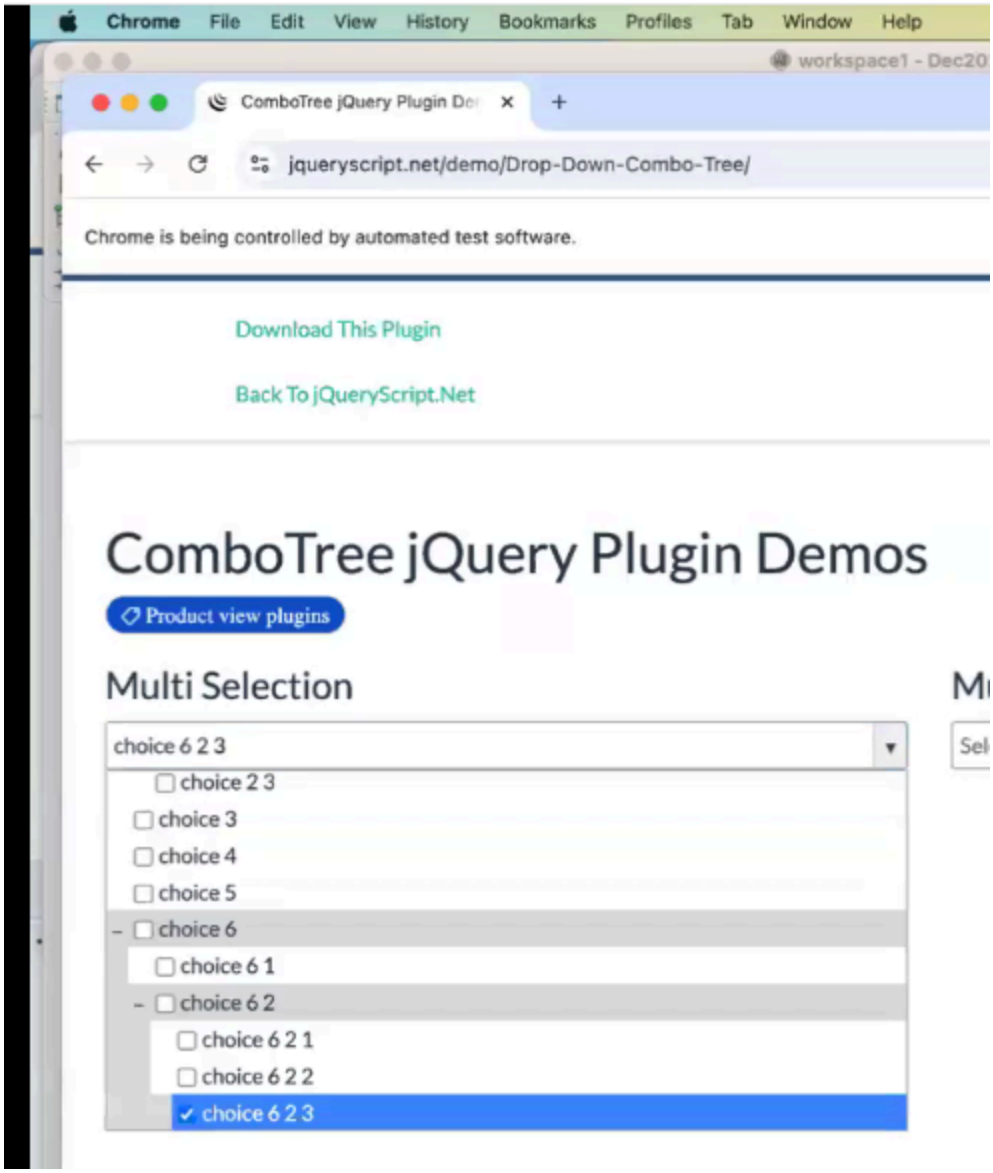
Click choice 3 and break-

```
29
30          for(WebElement e : choices) {
31              String text = e.getText();
32              System.out.println(text);
33                  if(text.contains("choice 3")) {
34                      e.click();
35                      break;
36                  }
37          }
38
39      }
```



Select sub choice-

```
29
30          for(WebElement e : choices) {
31              String text = e.getText();
32              System.out.println(text);
33                  if(text.contains("choice 6 2 3")) {
34                      e.click();
35                      break;
36                  }
37          }
38
39      }
40
41  }
42
```



Generic method-

```
28
29    public static void selectChoice(By choice, By choicesList, String choiceValue) throws InterruptedException {
30
31        driver.findElement(choice).click();
32
33        Thread.sleep(2000);
34
35
36        List<WebElement> choices = driver.findElements(choicesList);
37        System.out.println(choices.size());
38
39        for (WebElement e : choices) {
40            String text = e.getText();
41            System.out.println(text);
42            if (text.contains(choiceValue)) {
43                e.click();
44                break;
45            }
46        }
47
```

Clearer line 29-

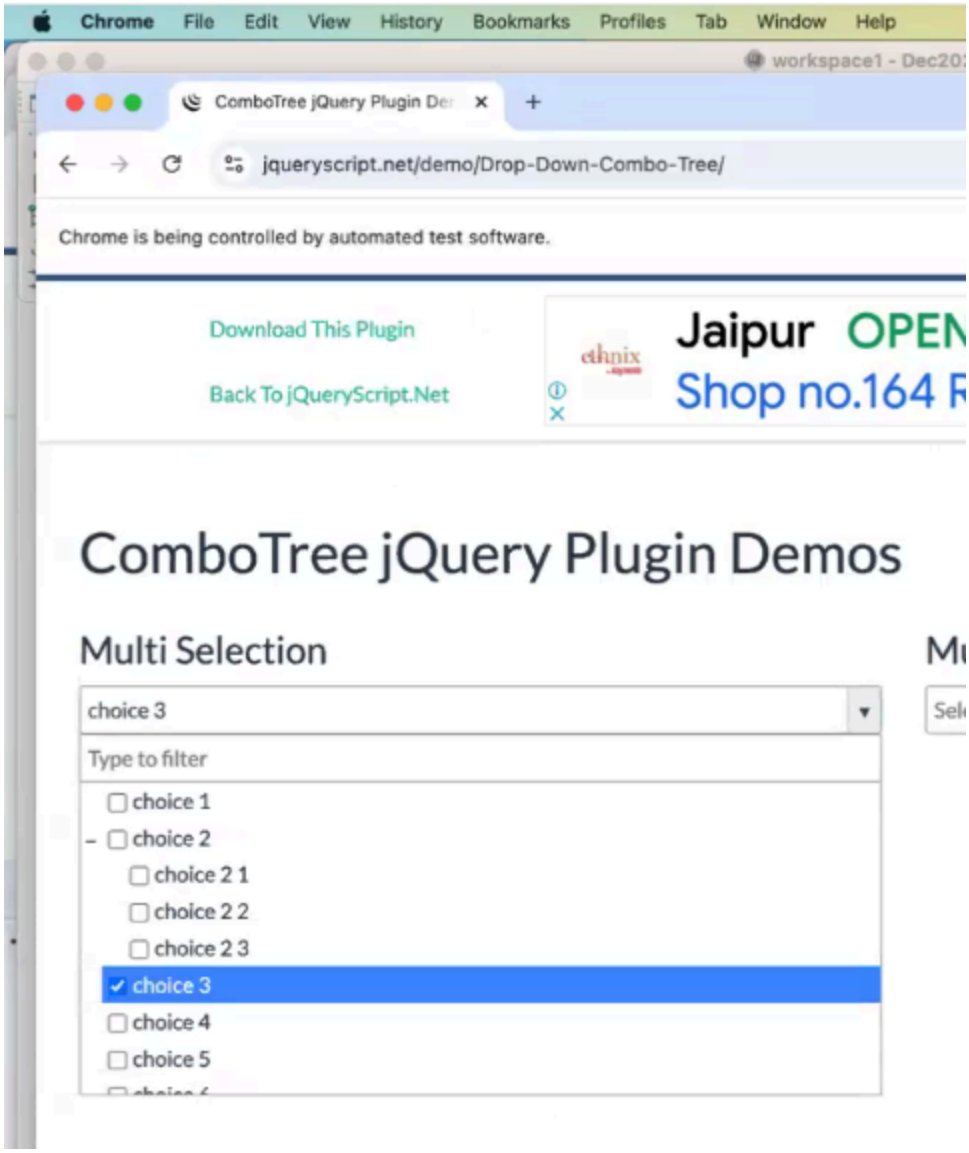
public static void selectChoice(By choice, By choicesList, String choiceValue) **throws** InterruptedException {

Updated caller code-

Select choice3 from the option.

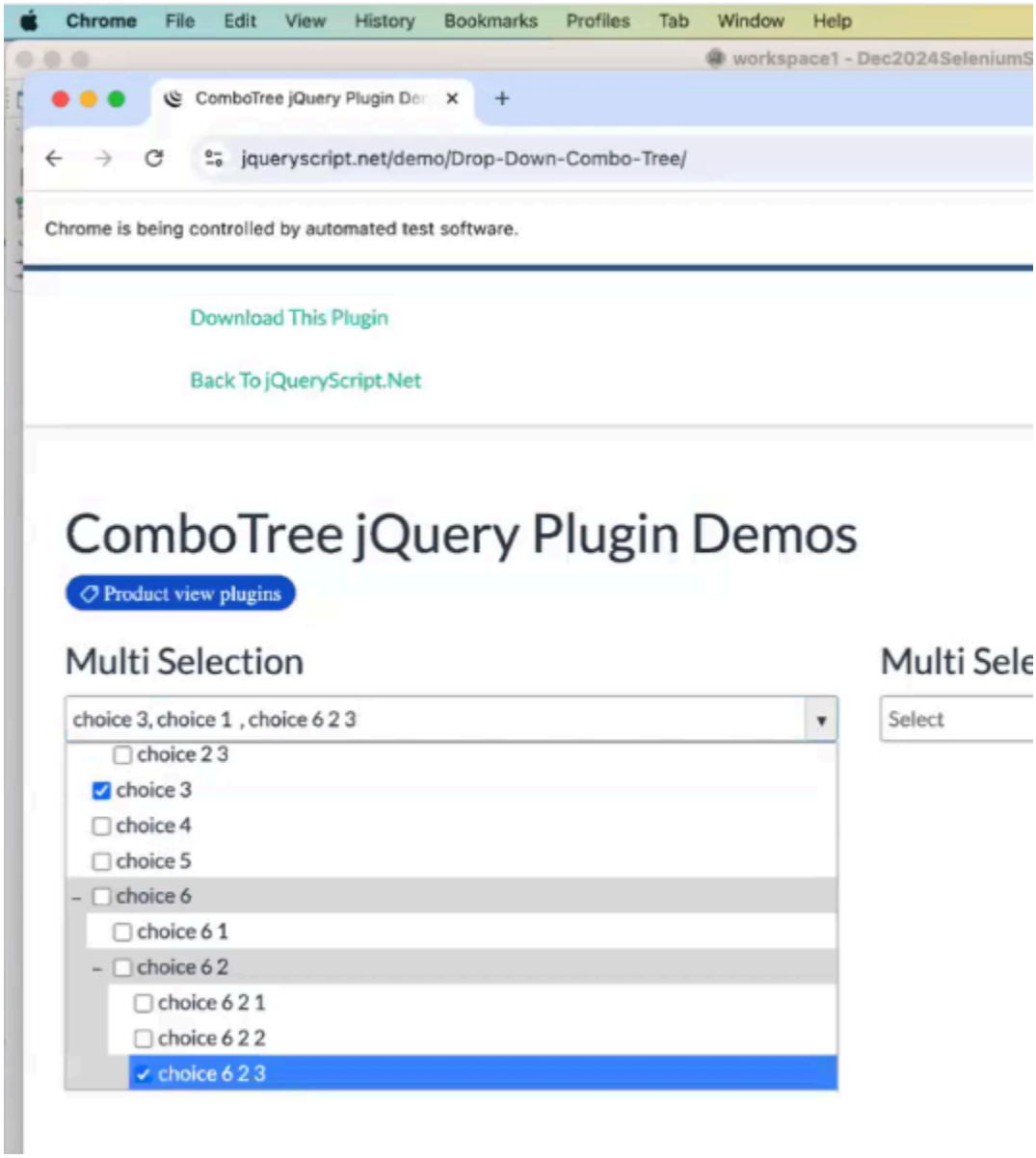
```
 9
10 public class JqueryDropDownHandle {
11
12     static WebDriver driver;
13
14     public static void main(String[] args) throws InterruptedException {
15
16         driver = new ChromeDriver();
17         driver.get("https://www.jqueryscript.net/demo/Drop-Down-Combo-Tree");
18
19         Thread.sleep(2000);
20
21         By choice = By.id("justAnInputBox");
22         By choicesList = By.xpath("(//div[@class='comboTreeDropDownContainer'])[1]//li/span[@class='comboTreeItemTitle']");
23
24         selectChoice(choice, choicesList, "choice 3");
25
26
```

um_Sessions · Updated 16 Jul 2025 by Naveen Khunteta



Multi selection-
Bad approach if we want to select loads of choices.
You cant keep giving 100's of statements for every choice.

```
//tc_1: single selection:
selectChoice(choice, choicesList, "choice 3");
selectChoice(choice, choicesList, "choice 1");
selectChoice(choice, choicesList, "choice 6 2 3");
```



Second approach-

Pass as many parameters to generic as you want. Not good approach as parameter count can change anytime.

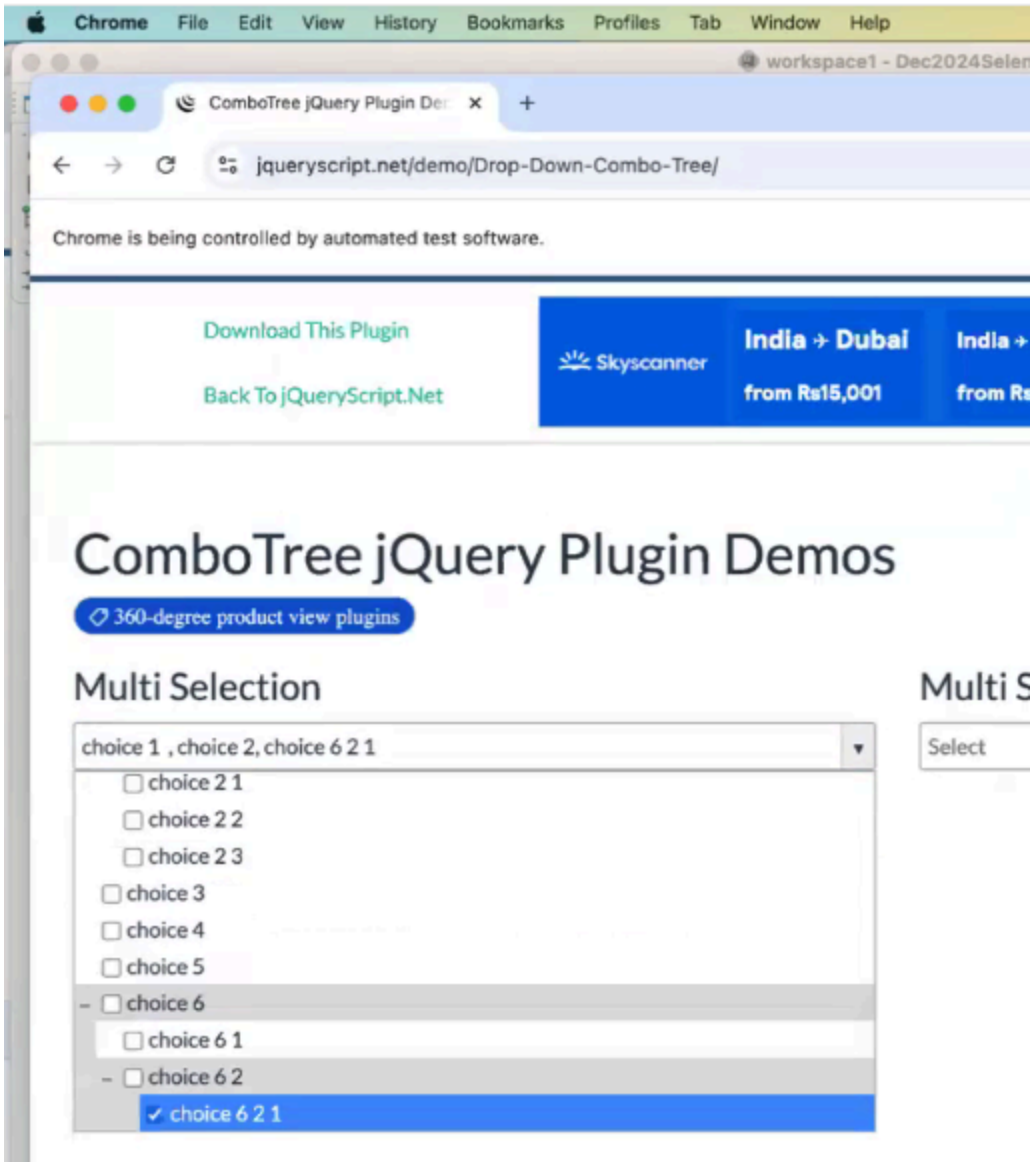
```
33
34    public static void selectChoice(By choice, By choicesList, String choiceValue1, String ch2, String ch3) throws InterruptedException {
35
36        driver.findElement(choice).click();
37
38        Thread.sleep(2000);
39
40        List<WebElement> choices = driver.findElements(choicesList);
41        System.out.println(choices.size());
42
43        for (WebElement e : choices) {
44            String text = e.getText();
45            System.out.println(text);
46            if (text.contains(choiceValue)) {
47                e.click();
48                break;
49            }
50        }
51
```

Clearer line 34-

public static void selectChoice(By choice, By choicesList, String choiceValue1, String ch2, String ch3) **throws** InterruptedException {

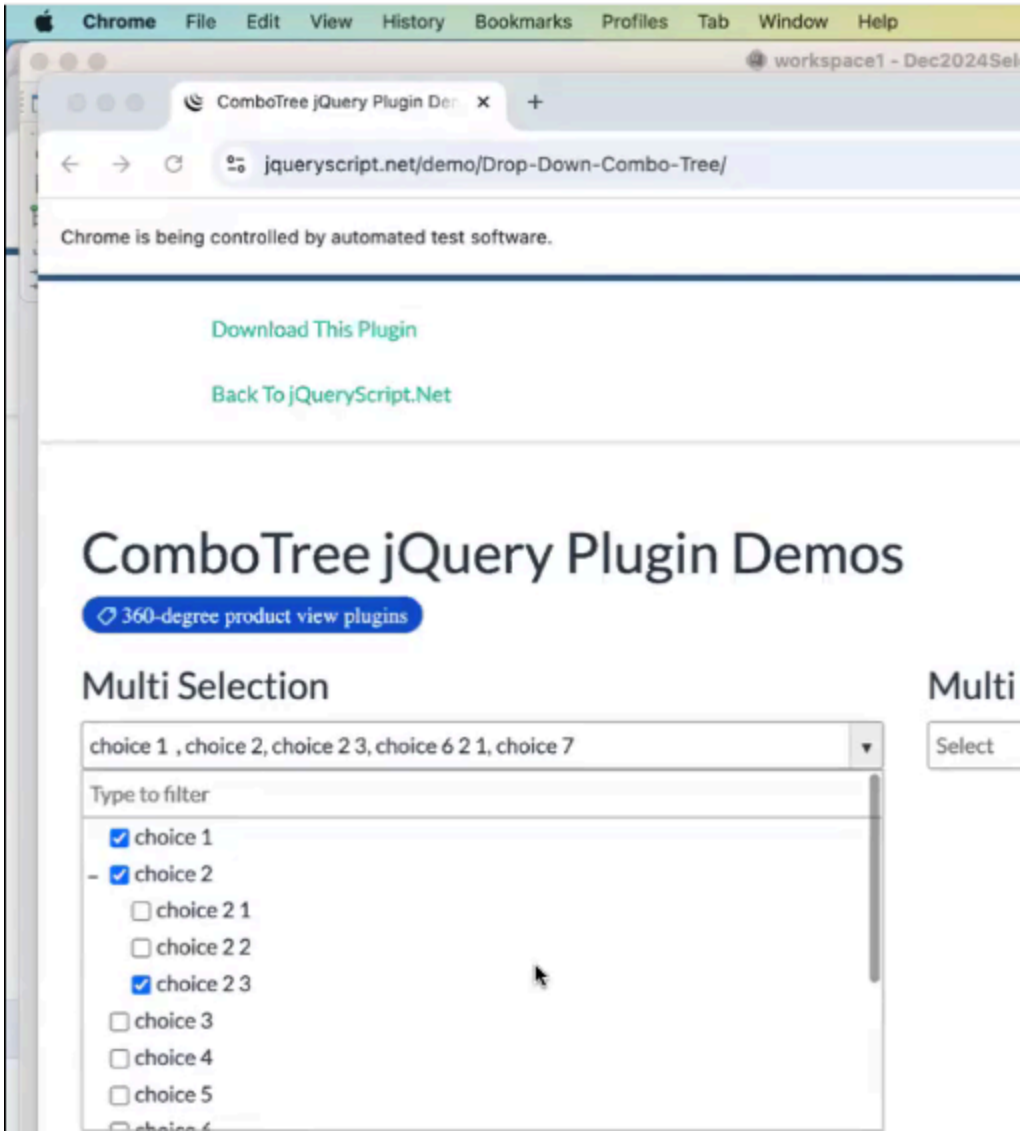
Awesome multi selection code using … - 24.99
We used trim so that the choice names we give and the one returned by application matches. In application there are choices with space.

```
26        //selectChoice(choice, choicesList, "choice 3");
27
28        //tc_2: multi selection:
29        selectChoice(choice, choicesList, "choice 1", "choice 2", "choice 6 2 1");
30
31
32    }
33
34    public static void selectChoice(By choice, By choicesList, String... choiceValue) throws InterruptedException {
35
36        driver.findElement(choice).click();
37
38        Thread.sleep(2000);
39
40        List<WebElement> choices = driver.findElements(choicesList);
41        System.out.println(choices.size());
42
43        for (WebElement e : choices) {
44            String text = e.getText();
45            System.out.println(text);
46
47                for(String value : choiceValue) {
48                    if(text.trim().equals(value)) {
49                        e.click();
50                        break;
51                    }
52                }
53
54        }
55
56    }
57
58 }
59
```



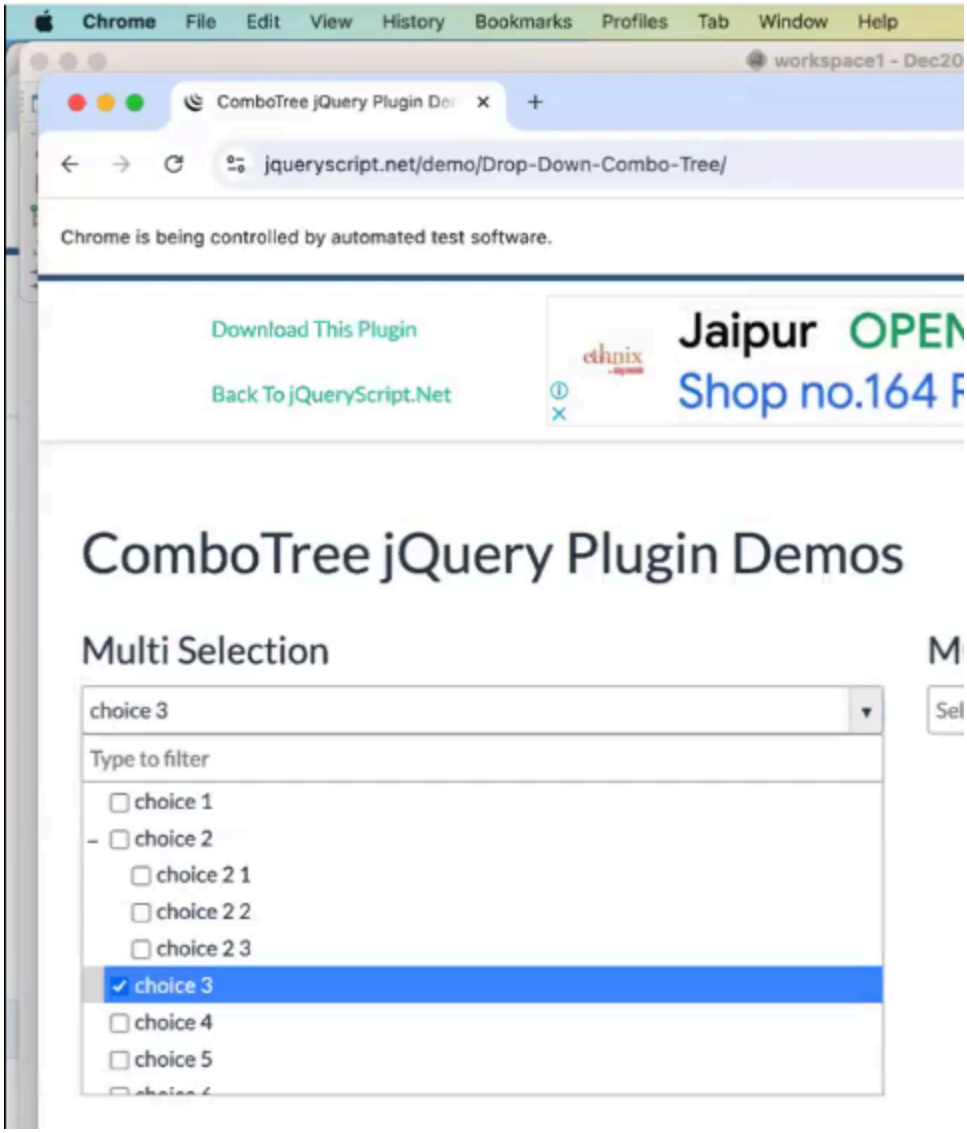
Send more choices-

```
27
28          //tc_2: multi selection:
29          selectChoice(choice, choicesList, "choice 1", "choice 2", "choice 6 2 1", "choice 7", "choice 2 3");
30
```



Works with single choice also-

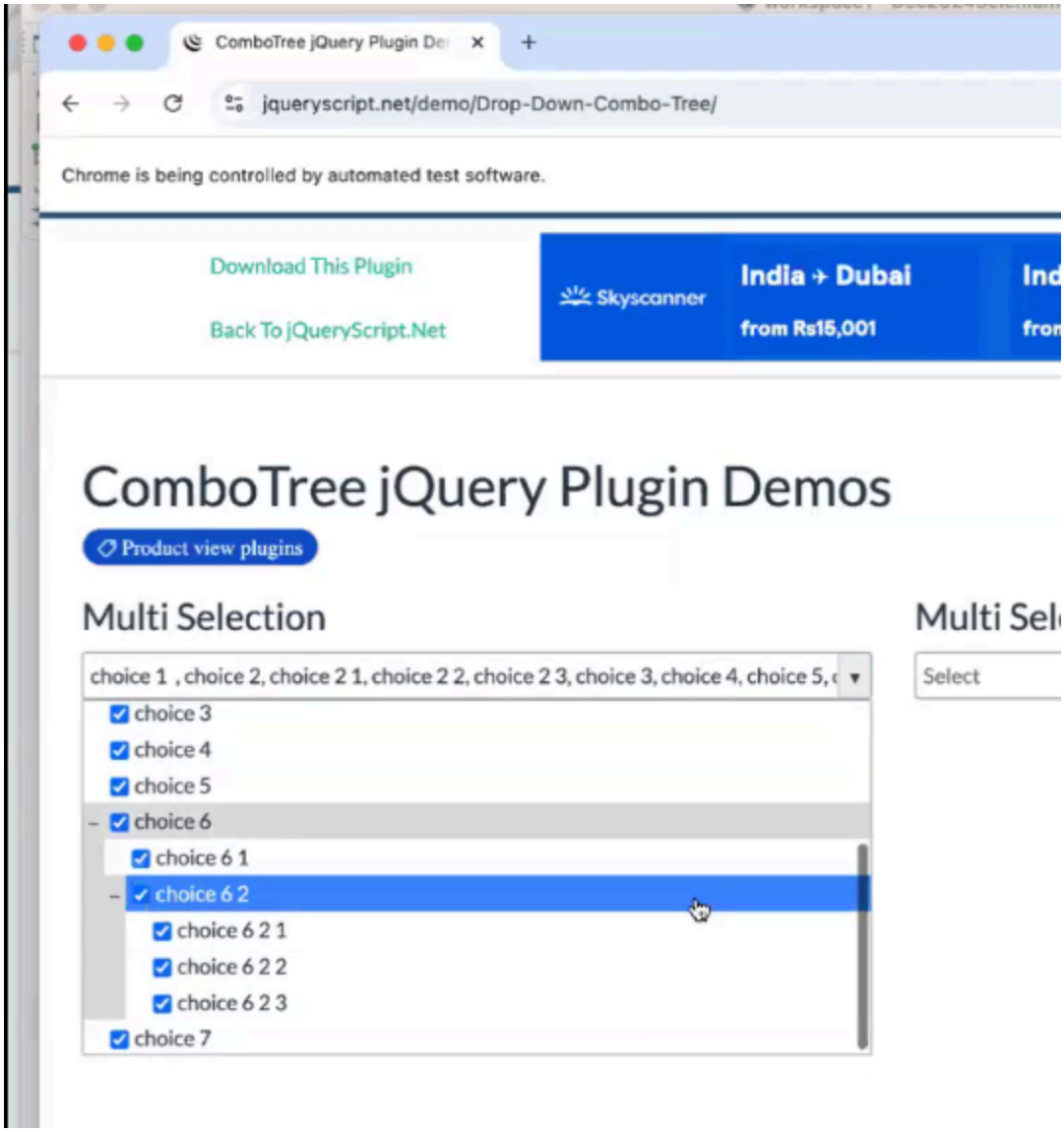
```
23                  .xpath("//div[@class='comboTreeDropDownC
24
25          //tc_1: single selection:
26          selectChoice(choice, choicesList, "choice 3");
27
```



What to do if we want to select all choices- We cant keep passing so many parameters.
We use identifier.
If the choice value is identifier then select all, or just select the ones needed in else block.

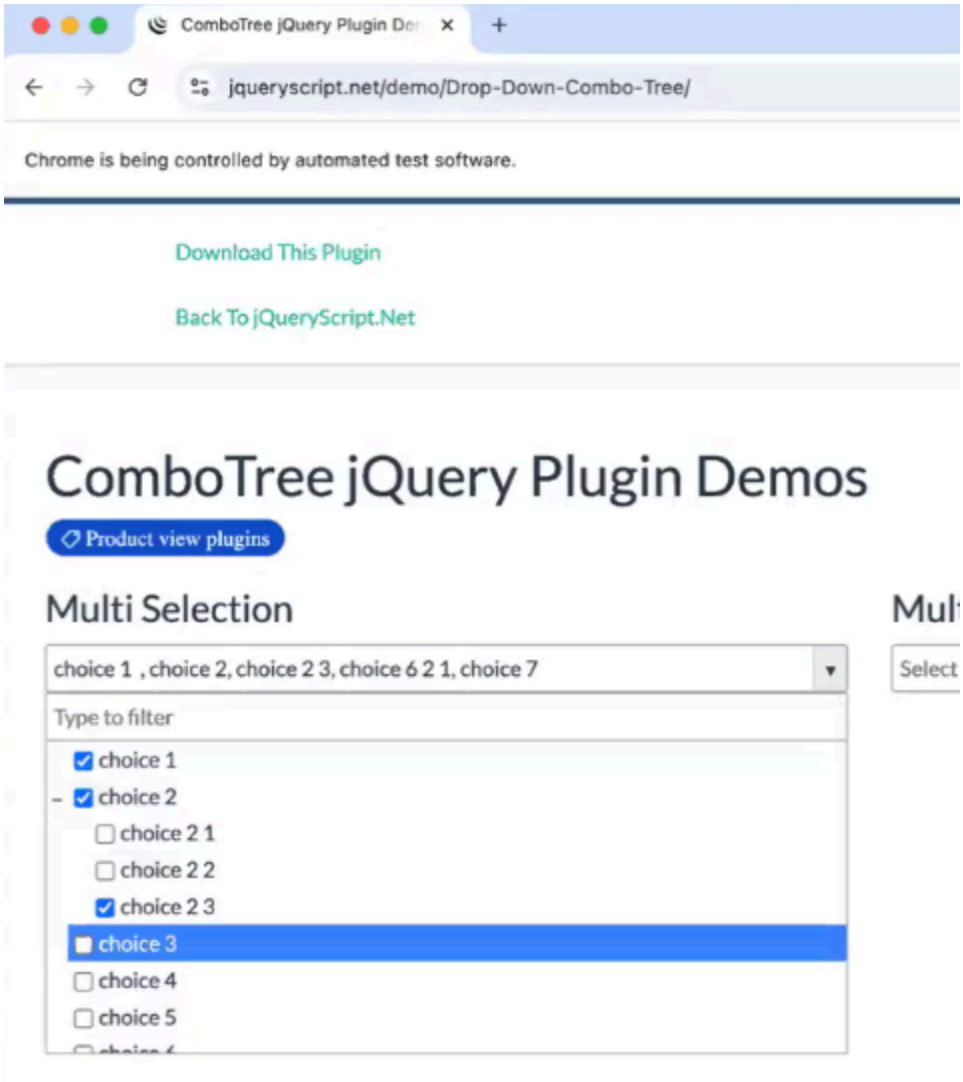
If the value you pass is “all” then just click all the choices.
Since choicevalue is array, the first value we pass will be **all** or **something else.** 36.59

```java
        // tc_3: all selection:

        selectChoice(choice, choicesList, "ALL");
    }

    public static void selectChoice(By choice, By choicesList, String... choiceValue) throws InterruptedException {

        driver.findElement(choice).click();
        Thread.sleep(2000);
        List<WebElement> choices = driver.findElements(choicesList);
        System.out.println(choices.size());

        if (choiceValue[0].equalsIgnoreCase("all")) {
            // logic to select all the choices:
            for (WebElement e : choices) {
                e.click();
            }
        } else {

            for (WebElement e : choices) {
                String text = e.getText();
                System.out.println(text);

                for (String value : choiceValue) {
                    if (text.trim().equals(value)) {
                        e.click();
                        break;
                    }
                }

            }
        }

    }

}
```



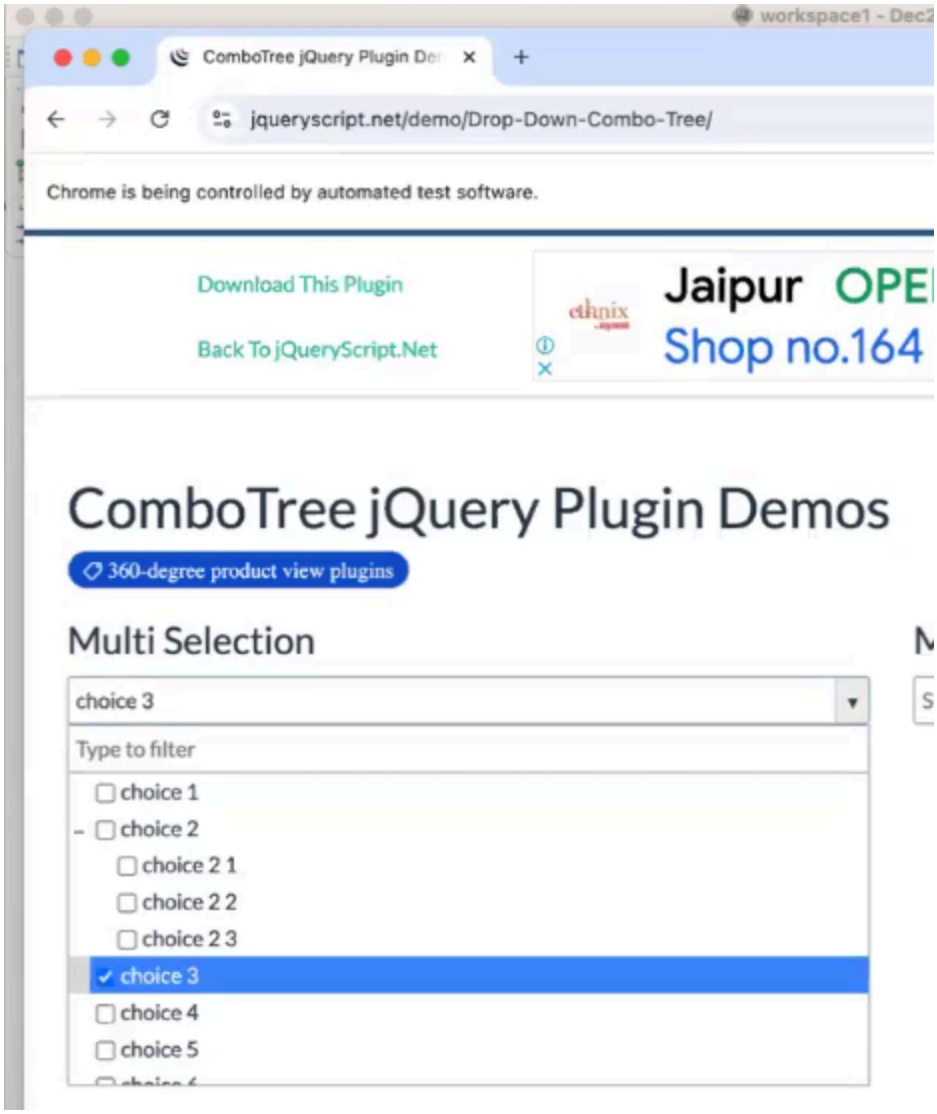
Check if multi selection works-

```
27
28        // tc_2: multi selection:
29        selectChoice(choice, choicesList, "choice 1", "choice 2", "choice 6 2 1", "choice 7", "choice 2 3");
30
31        // tc 3: all selection:
```



Check if single selection works-

```
24
25        // tc_1: single selection:
26        selectChoice(choice, choicesList, "choice 3");
27
```



See how java documentation works-
You gave comment and see how the comment looks like when you try to enter the method name.

```
35
36    /**
37     * this method is used to select the choices with three different use cases:
38     * 1. single selection : selectChoice(choice, choicesList, "choice 3");
39     * 2. multi selection : selectChoice(choice, choicesList, "choice 1", "choice 2", "choice 2 3");
40     * 3. all selection: use "all/All" to select all the choices: selectChoice(choice, choicesList, "ALL");
41     * @param choice
42     * @param choicesList
43     * @param choiceValue
44     * @throws InterruptedException
45     */
```

Move to element util-

```
214
215     //*****************drop down utils --- non select based*************//
216     /**
217      * this method is used to select the choices with three different use cases:
218      * 1. single selection : selectChoice(choice, choicesList, "choice 3");
219      * 2. multi selection : selectChoice(choice, choicesList, "choice 1", "choice 2", "choice 2 3");
220      * 3. all selection: use "all/All" to select all the choices: selectChoice(choice, choicesList, "ALL");
221      * @param choice
222      * @param choicesList
223      * @param choiceValue
224      * @throws InterruptedException
225      */
226     public void selectChoice(By choice, By choicesList, String... choiceValue) throws InterruptedException {
227         doClick(choice);
228         Thread.sleep(2000);
229         List<WebElement> choices = getElements(choicesList);
230         System.out.println(choices.size());
231
232         if (choiceValue[0].equalsIgnoreCase("all")) {
233             // logic to select all the choices:
234             for (WebElement e : choices) {
235                 e.click();
236             }
237         } else {
238
239             for (WebElement e : choices) {
240                 String text = e.getText();
241                 System.out.println(text);
242
243                 for (String value : choiceValue) {
244                     if (text.trim().equals(value)) {
245                         e.click();
246                         break;
247                     }
248                 }
249
250             }
251         }
252
253     }
```

String locators-

Naveen website.

Register page.

Firstname, lastname, email, phone.

Every locator stored in string format.

```
1  package seleniumsessions;
2
3  import org.openqa.selenium.WebDriver;
4  import org.openqa.selenium.chrome.ChromeDriver;
5
6  public class StringLocator {
7
8      static WebDriver driver;
9
10     public static void main(String[] args) {
11         driver = new ChromeDriver();
12         driver.get("https://naveenautomationlabs.com/opencart/index.php?route=account/register");
13
14         String firstName_ID = "input-firstname";
15         String lastName_ID = "input-lastname";
16         String email_Name = "email";
17         String telephone_Xpath = "//input[@placeholder='Telephone']";
18
```

Clearer line 17-

String telephone_Xpath = "//input[@placeholder='Telephone']";

```
29
30     public static WebElement getElement(By locator) {
31         return driver.findElement(locator);
32     }
```

55.00

Default value of non primitive is null.

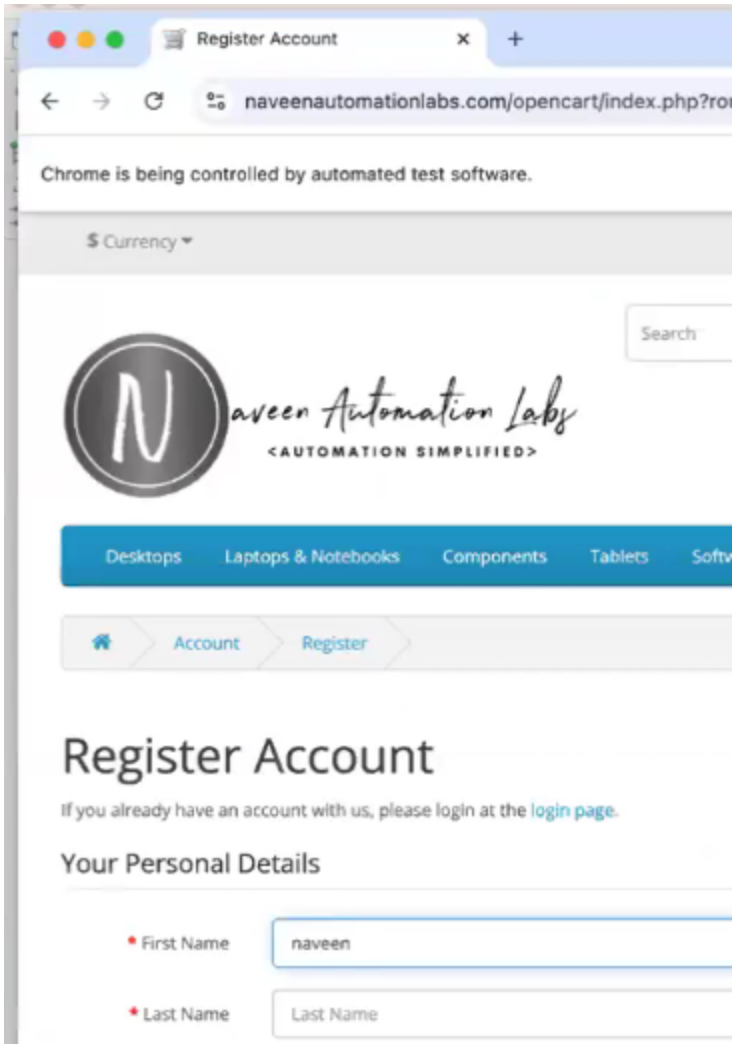
What if we dont want to use any of the locators directly-

```java
    public static By getBy(String locatorType, String locatorValue) {

        By locator = null;

        switch (locatorType.toUpperCase()) {
        case "ID":
            locator = By.id(locatorValue);
            break;
        case "NAME":
            locator = By.name(locatorValue);
            break;
        case "CLASS":
            locator = By.className(locatorValue);
            break;
        case "XPATH":
            locator = By.xpath(locatorValue);
            break;
        case "CSS":
            locator = By.cssSelector(locatorValue);
            break;
        case "LINKTEXT":
            locator = By.linkText(locatorValue);
            break;
        case "PARTIALLINKTEXT":
            locator = By.partialLinkText(locatorValue);
            break;
        case "TAGNAME":
            locator = By.tagName(locatorValue);
            break;

        default:
            System.out.println("plz pass the right locator type: " + locatorType);
            break;
        }

        return locator;

    }
```

Play

Call method-

```java
        getElement(getBy("ID", firstName_ID)).sendKeys("naveen");

    }
```

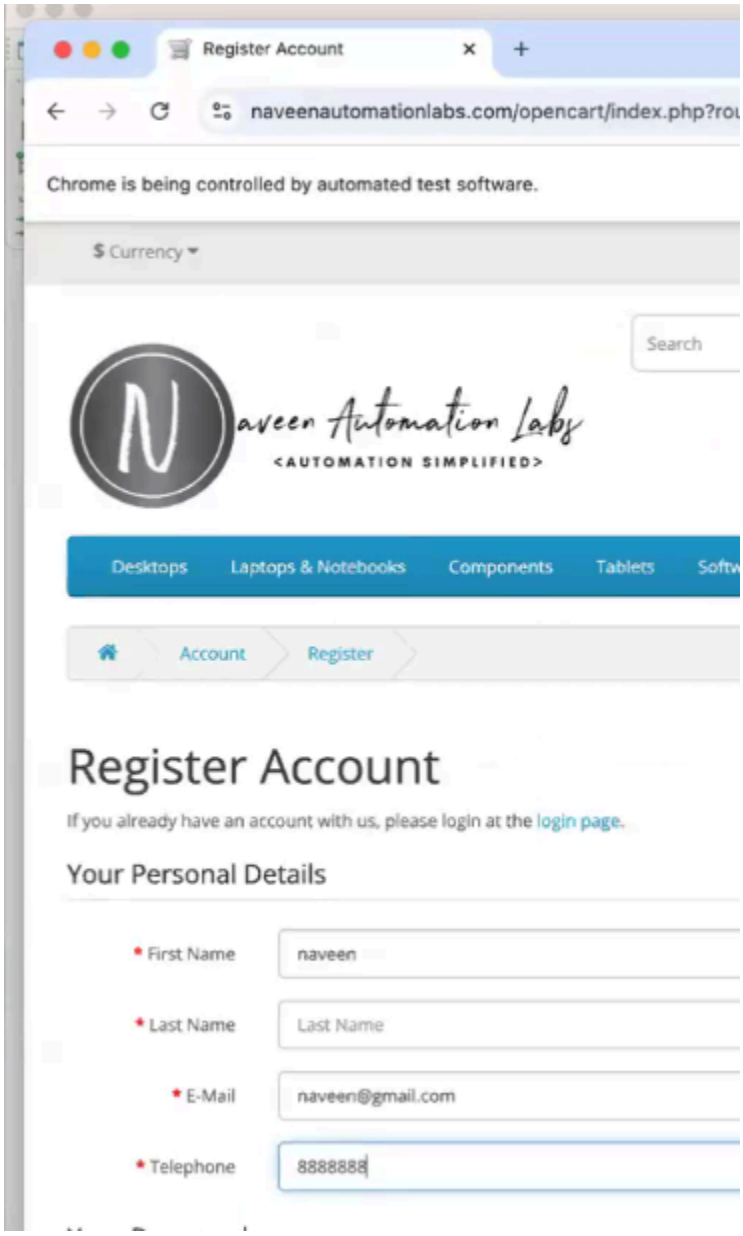


Overload the get element method-
We are including the string locators directly here only.

```
28
29⊖     public static WebElement getElement(String locatorType, String locatorValue ) {
30          return driver.findElement(getBy(locatorType, locatorValue));
31      }
```

Call the method-

```
20
21          getElement("ID", firstName_ID).sendKeys("naveen");
22          getElement("NAME", email_Name).sendKeys("naveen@gmail.com");
23          getElement("XPATH", telephone_Xpath).sendKeys("8888888");
24
```



Move to element util-

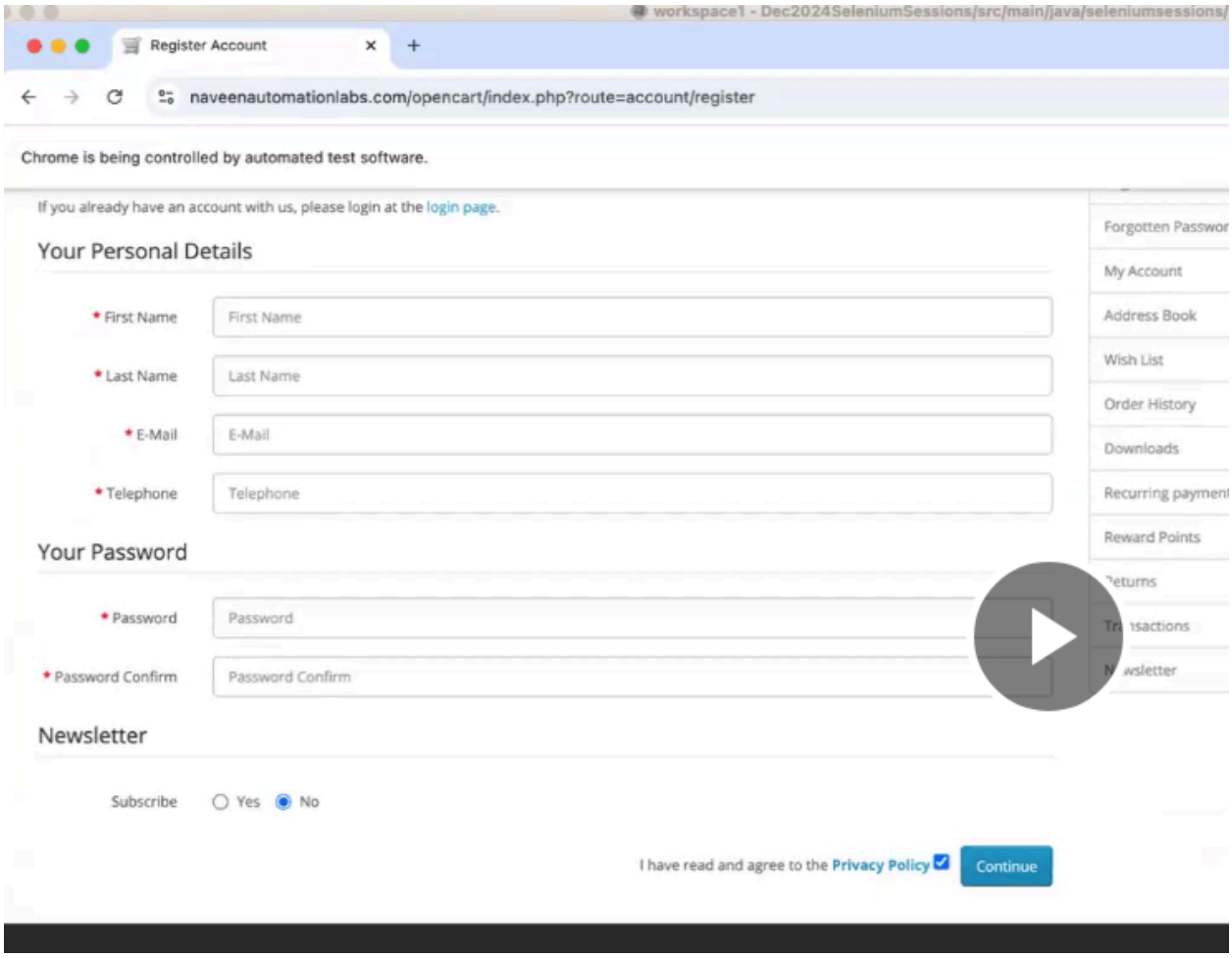
```java
public By getBy(String locatorType, String locatorValue) {

    By locator = null;

    switch (locatorType.toUpperCase()) {
    case "ID":
        locator = By.id(locatorValue);
        break;
    case "NAME":
        locator = By.name(locatorValue);
        break;
    case "CLASS":
        locator = By.className(locatorValue);
        break;
    case "XPATH":
        locator = By.xpath(locatorValue);
        break;
    case "CSS":
        locator = By.cssSelector(locatorValue);
        break;
    case "LINKTEXT":
        locator = By.linkText(locatorValue);
        break;
    case "PARTIALLINKTEXT":
        locator = By.partialLinkText(locatorValue);
        break;
    case "TAGNAME":
        locator = By.tagName(locatorValue);
        break;

    default:
        System.out.println("plz pass the right locator type: " + locatorType);
        break;
    }

    return locator;

}
```

```java
public WebElement getElement(String locatorType, String locatorValue) {
    return driver.findElement(getBy(locatorType, locatorValue));
}
```

Click on agree checkbox-

```java
String firstName_ID = "input-firstna
String lastName_ID = "input-lastname
String email_Name = "email";
String telephone_Xpath = "//input[@
String checkBox_name = "agree";
```

```
25
26          ElementUtil eleUtil = new ElementUtil(driver);
27          eleUtil.doClick(eleUtil.getBy("NAME", checkBox_name));
28
```

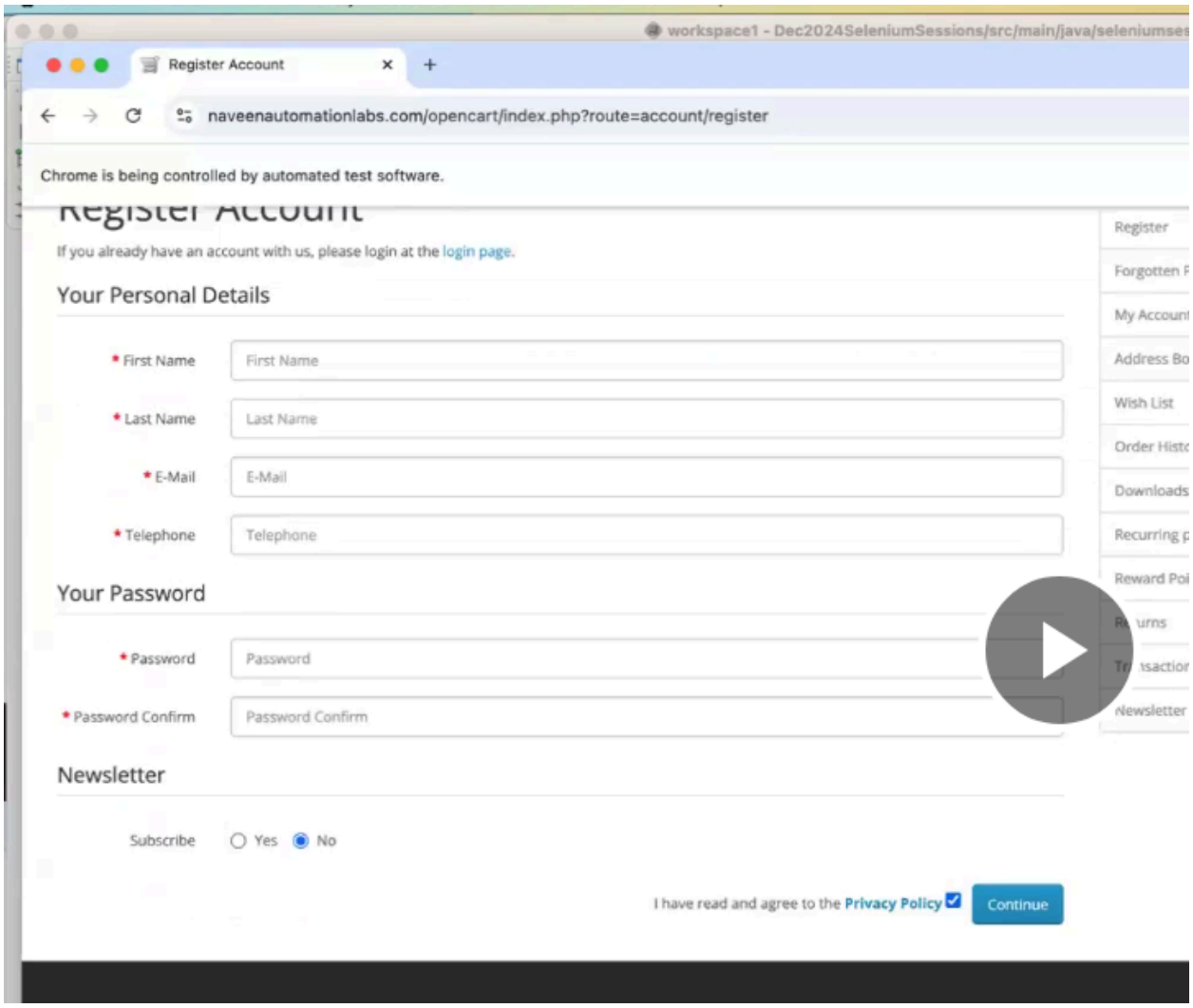


Overload do click on the element util page-

```
38      }
39
40      public void doClick(String locatorType, String locatorValue) {
41          getElement(locatorType, locatorValue).click();
42      }
```

Now in the test class we can call overloaded method directly-

```
25
26          ElementUtil eleUtil = new ElementUtil(driver);
27          eleUtil.doClick("NAME", checkBox_name);
28
```



Overload send keys in element util-

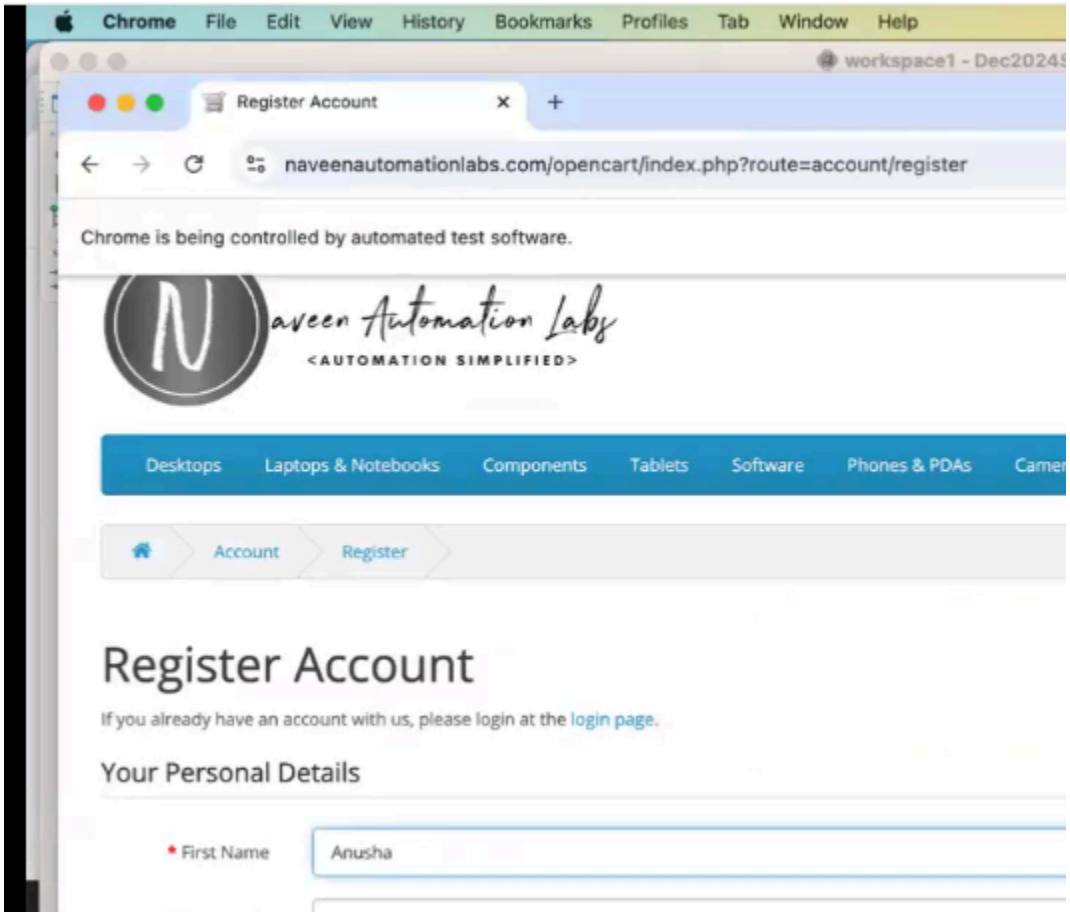
```
30
31    public void doSendKeys(String locatorType, String locatorValue, String value) {
32        nullCheck(value);
33        getElement(locatorType, locatorValue).sendKeys(value);
34    }
```

Clearer line 31-

public void doSendKeys(String locatorType, String locatorValue, String value) {

Call in the test class-

```
25
26          ElementUtil eleUtil = new ElementUtil(driver);
27          eleUtil.doClick("NAME", checkBox_name);
28          eleUtil.doSendKeys("ID", firstName_ID, "Anusha");
29
30 |
31      }
```



Overload another send keys which has char sequence-

Element util class.

```
40
41⊖     public void doSendKeys(String locatorType, String locatorValue, CharSequence... value) {
42          nullCheck(value);
43          getElement(locatorType, locatorValue).sendKeys(value);
44      }
45
```

Avvai word used - 1.02.17.
Naveen doesnt like string - 1.06.46