<u>**Selenium Java - Tutorial 12- Dropdowns**</u>

This is the next tutorial in selenium-java series. Please go through the previous tutorials before you start this one. In the last tutorial, we learned how to handle window tabs using iterator. In this tutorial we will see how to handle varipus types of dropdowns!

What you will Learn:
1. Static dropdowns
2. selectByIndex
3. selectByVisibleText
4. selectByValue
5. Dynamic dropdown – Parent/Child relationship

<u>**Static dropdown**</u>
The static dropdowns are the traditional ones that are identified by 'Select' tag. Navigate to http://goair.in & inspect passenger dropdown, you will see that it is represented by 'select' tag
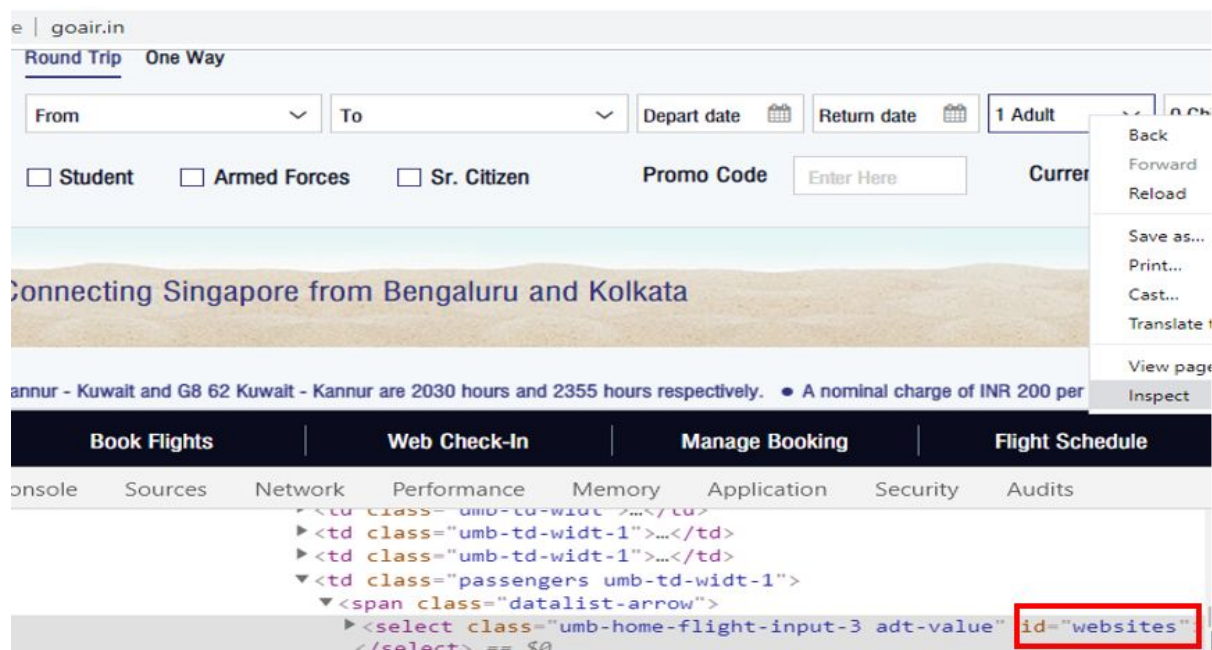

Figure 1

Let us use 'id' attribute to identify this dropdown

Figure 2

For static dropdowns, we can use an in-built 'Select' class in selenium. Let us create an object of 'Select' class

Figure 3

To resolve the error, import the 'Select' class

Figure 4

Next we will create a reference pointing to the object

Figure 5

Next using this reference, we can call various 'selectBy' methods

Figure 6

selectByIndex
Let us use these 3 select methods one by one.

Figure 7

Run, notice that '3 Adults' is selected from dropdown (index starts from 0)

Figure 8

selectByVisibleText

Comment line 17, add line 18

```
15          driver.get("http://goair.in/");
16          Select s = new Select(driver.findElement(By.id("websites")));
17          //s.selectByIndex(2);
18          s.selectByVisibleText("4 Adults");
```
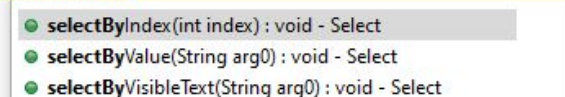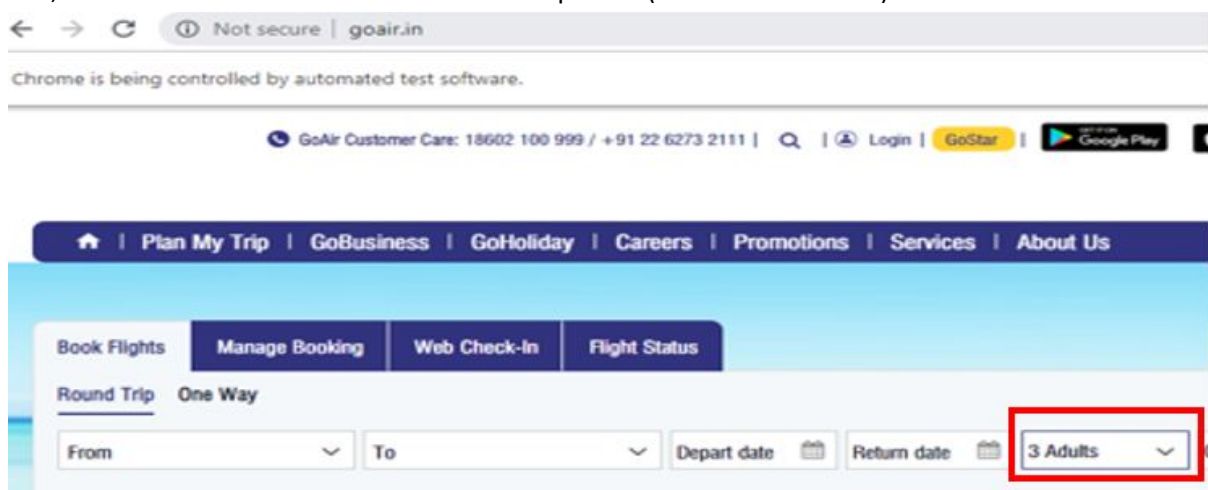
Figure 9

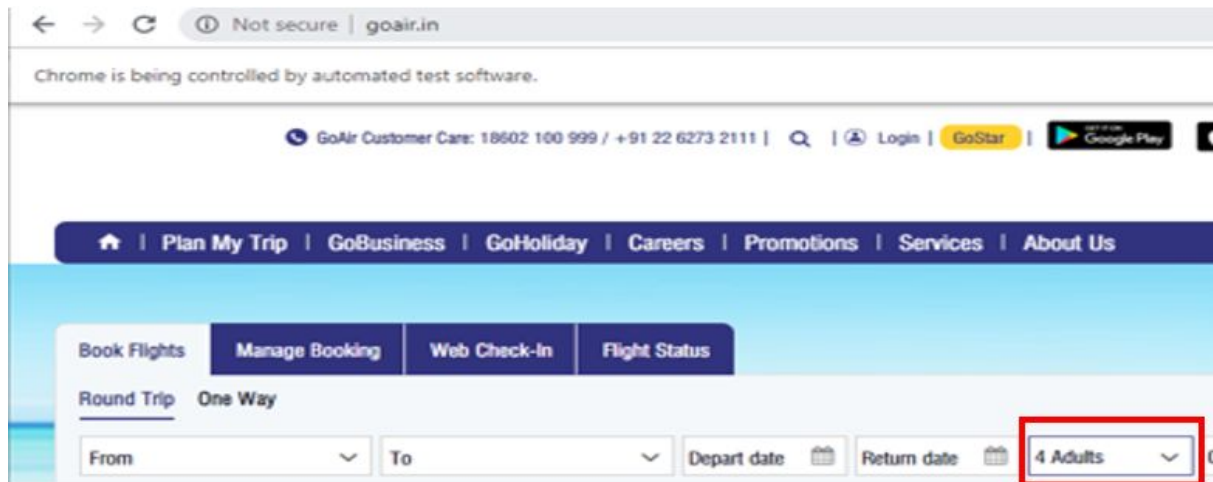Run, see that '4 Adults' is selected in the dropdown

Figure 10

selectByValue

Now expand 'select' tag, we can see the values like: 1 Adult, 2 Adults and so on...
So value="1" has value '1 Adult'

```
▼<select class="umb-home-flight-input-3
xpath="1"> == $0
    <option value="1">1 Adult</option>
    <option value="2">2 Adults</option>
    <option value="3">3 Adults</option>
    <option value="4">4 Adults</option>
    <option value="5">5 Adults</option>
    <option value="6">6 Adults</option>
</select>
```
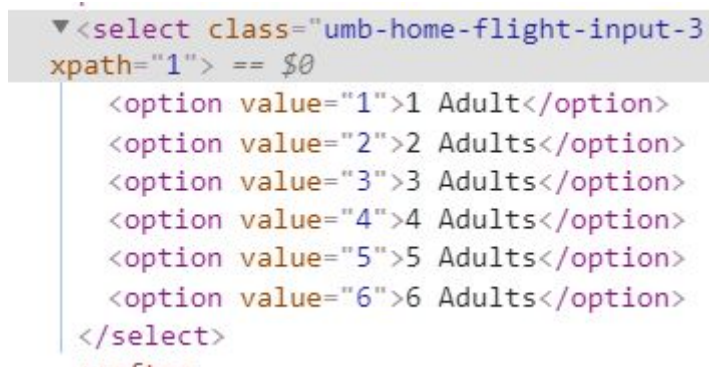
Figure 11

Comment line 18, add line 19

```
15          driver.get("http://goair.in/");
16          Select s = new Select(driver.findElement(By.id("websites")));
17          //s.selectByIndex(2);
18          //s.selectByVisibleText("4 Adults");
19          s.selectByValue("2");
```

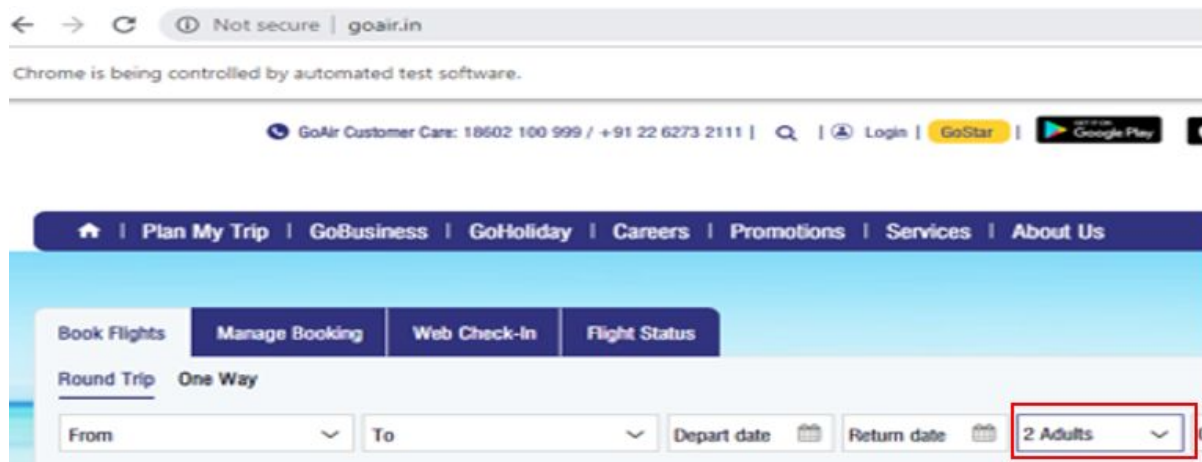*Figure 12*

Run, see that '2 Adults' is selected from dropdown

Figure 13

Dynamic dropdown – Parent/Child relationship

Dynamic dropdowns are not defined by 'select' tag. We can't use the methods of 'Select' class. To understand dynamic dropdowns, perform the below mentioned manual steps:

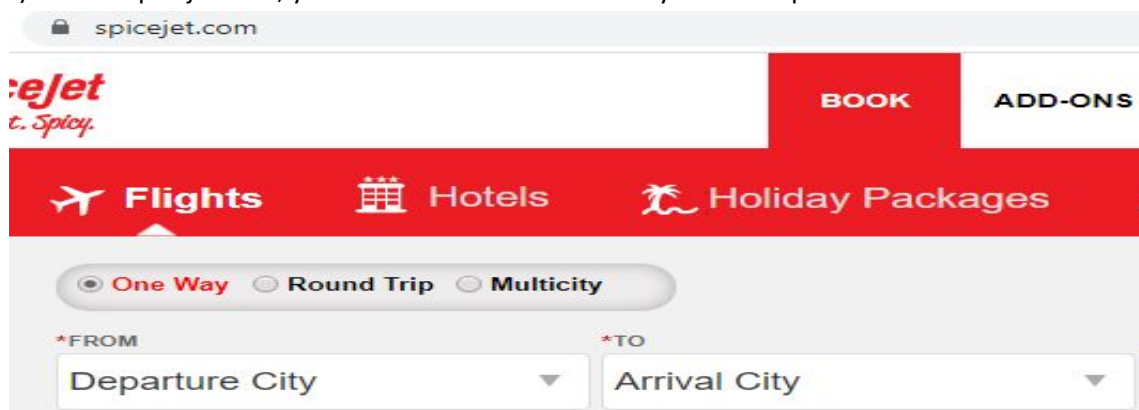a) Launch spicejet.com, you would see FROM and TO dynamic dropdowns

Figure 14

b) Click 'FROM' (Departure city) dropdown

Figure 15

c) Select any departure city, in this case 'Bengaluru (BLR)'. As soon you select any departure city, 'TO' (Arrival city) dropdown gets automatically pulled down
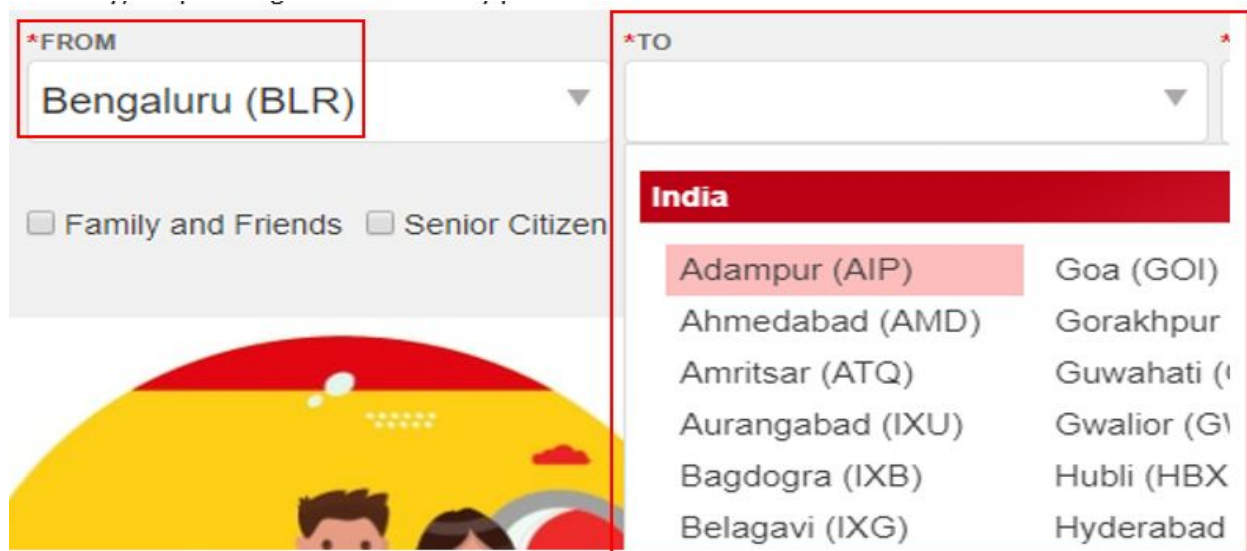

Figure 16

d) Select 'Chennai (MAA)' from 'TO' (arrival city dropdown).

Figure 17

So the manual steps performed are:
a) Launch spicejet.com
b) Click 'FROM' dropdown
c) Select 'Bengaluru (BLR)' from departure city dropdown
d) Select 'Chennai (MAA)' from arrival city dropdown

Let us now extract the xpaths of steps b), c) and d).
To find the xpath of 'FROM' dropdown, click 'FROM' dropdown.
Now right click and inspect the FROM field. The custom xpath would be:
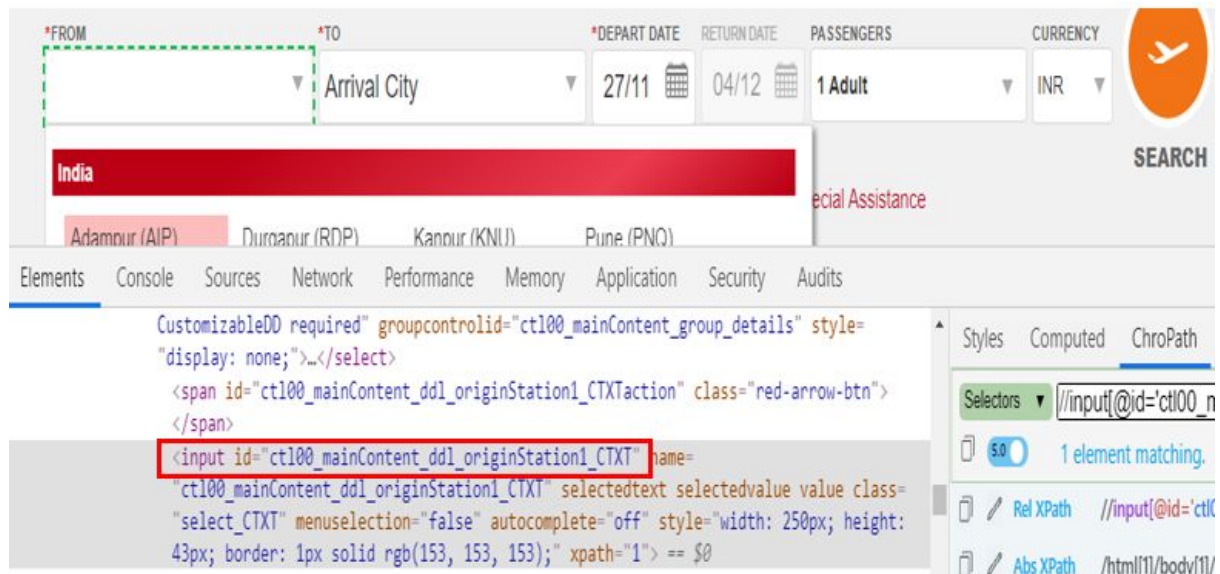**//input[@id='ctl00_mainContent_ddl_originStation1_CTXT']**

Figure 18

To find the xpath of 'Bengaluru (BLR)' in the 'FROM' dropdown, click ⬚ and click 'Bengaluru (BLR)' to get relative xpath: **//a[contains(text(),'Bengaluru (BLR)')]**
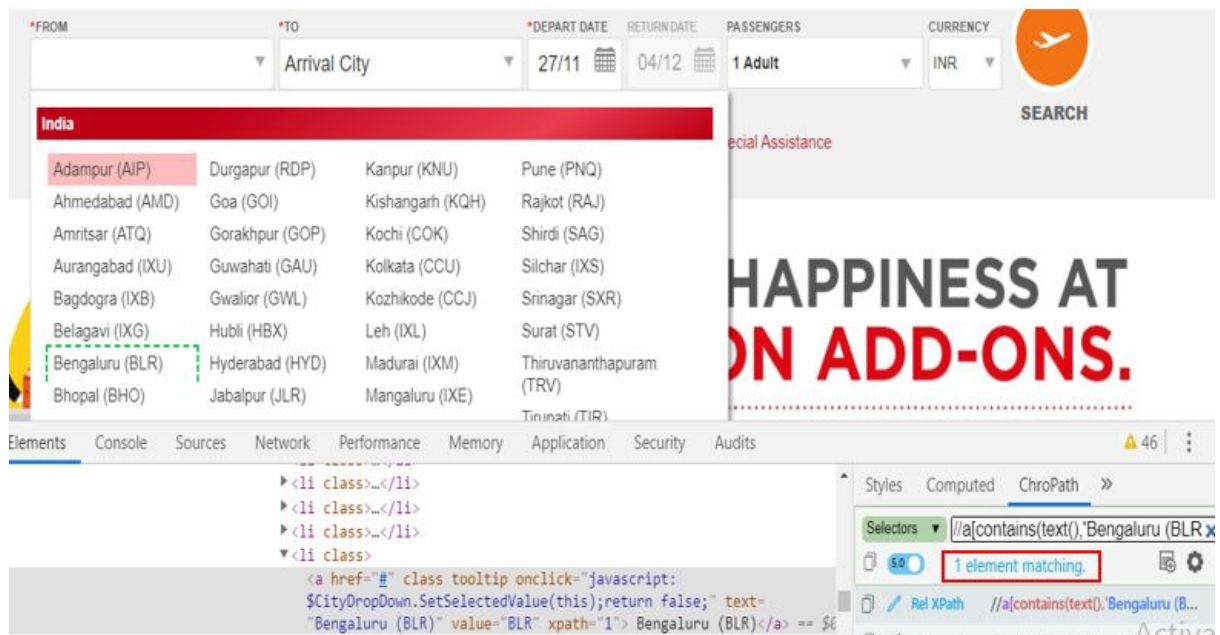Notice that we see the message '1 element matching' with this xpath

Figure 19

Next, in the 'TO' (arrival city) dropdown, we are going to select 'Chennai (MAA)'. The xpath would simply be: **//a[contains(text(),'Chennai (MAA)')]**
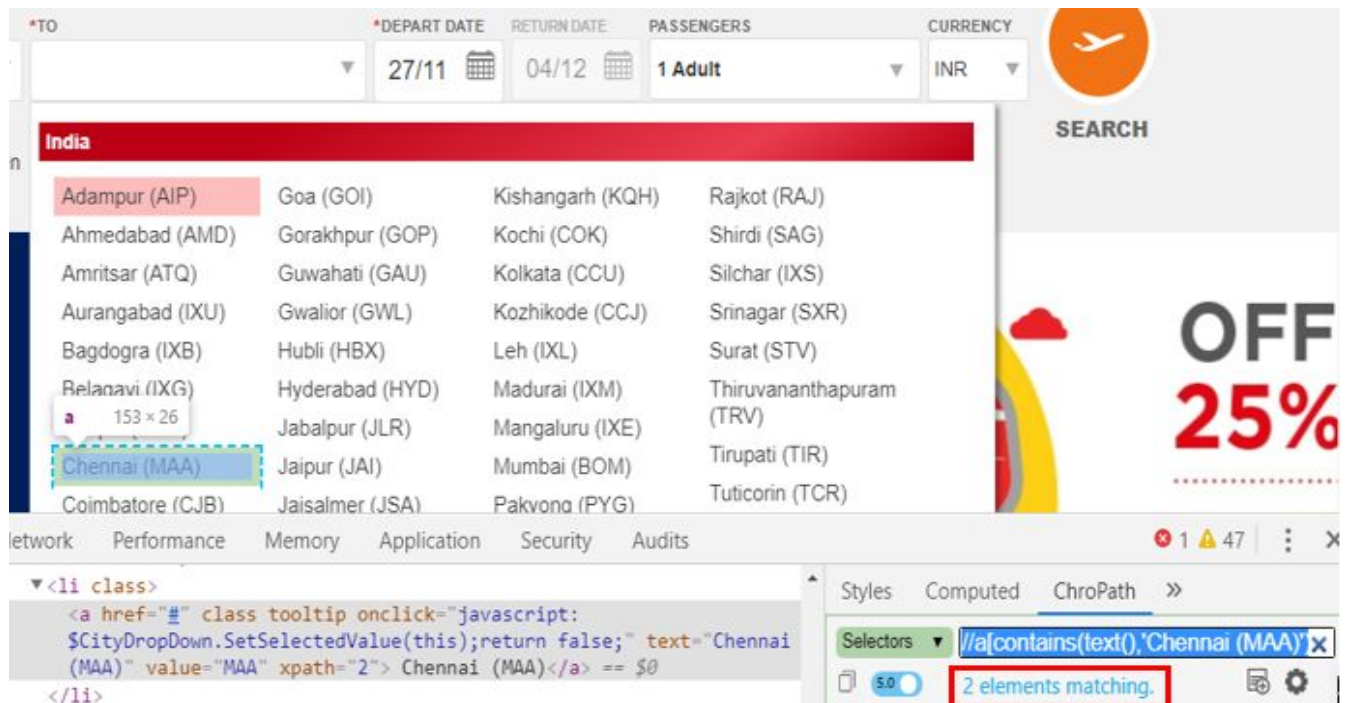
Figure 20

In above figure, notice the message '2 elements matching' for this xpath (we will talk about this later).

Let us now create our selenium script. Create a new class 'Spicejet', add steps a,b,c,d

```
5  public class Spicejet {
6
7⊖     public static void main(String[] args) {
8          System.setProperty("webdriver.chrome.driver", "C:\\Users\\DELL\\Desktop\\TRAINING\\Software\\chro
9
10         WebDriver driver = null;
11
12         driver = new ChromeDriver();
13         //step a: launch site
14         driver.get("https://spicejet.com");
15         //step b: Click 'Departure city' dropdown
16         driver.findElement(By.xpath("//input[@id='ctl00_mainContent_ddl_originStation1_CTXT']")).click();
17         //step c: Select 'Bengaluru (BLR)'from departure city
18         driver.findElement(By.xpath("//a[contains(text(),'Bengaluru (BLR)')]")).click();
19         //step d: Select 'Chennai (MAA)' from arrival city
20         driver.findElement(By.xpath("//a[contains(text(),'Chennai (MAA)')]")).click();
```

Figure 21

Run script. Notice that 'Bengaluru' is selected in the 'FROM' field, but nothing got selected in 'TO' field
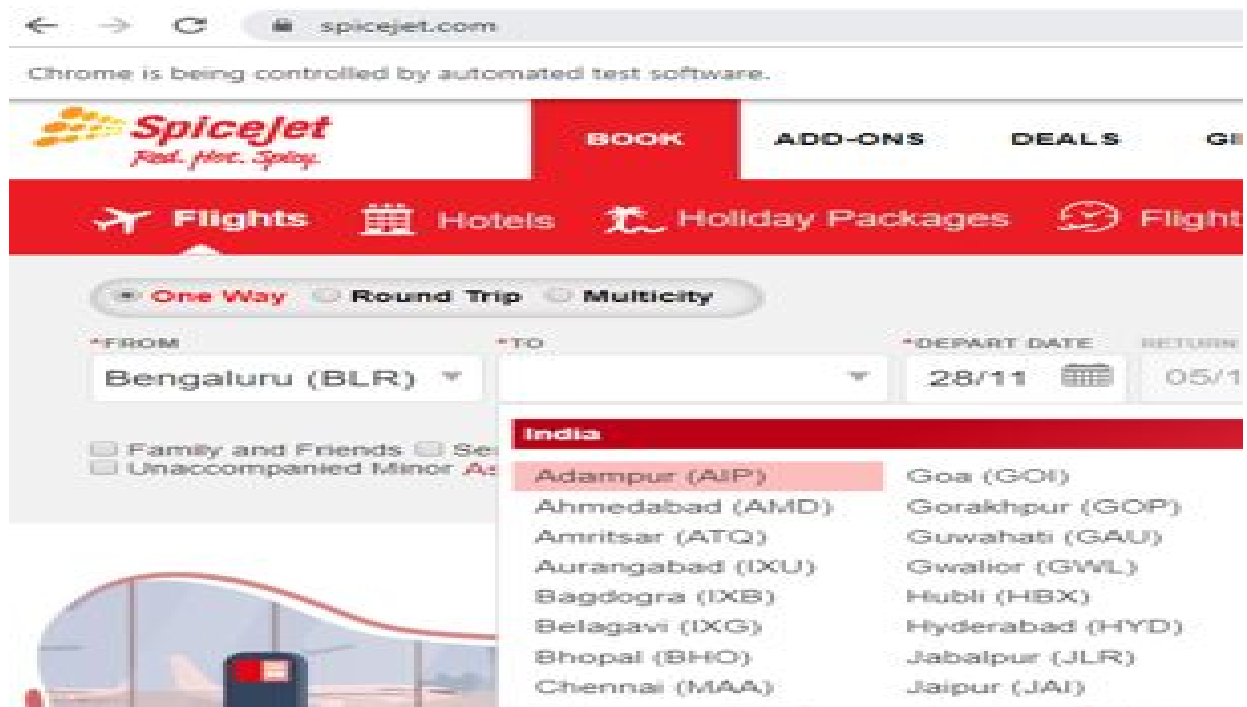
Figure 22

'ElementNotVisibleException' exception is thrown, see below. The question now arises is that, though 'Chennai' is visible in the 'TO' dropdown, why are we getting 'ElementNotVisibleException' ?

```
Nov 28, 2019 1:42:02 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: OSS
Exception in thread "main" org.openqa.selenium.ElementNotVisibleException: element not interactable
  (Session info: chrome=78.0.3904.108)
  (Driver info: chromedriver=2.42.591088 (7b2b2dca23cca0862f674758c9a3933e685c27d5),platform=Windows NT
Command duration or timeout: 0 milliseconds
```

Figure 23

The answer to this question lies in the message that we saw earlier '2 elements matching' for the xpath **//a[contains(text(),'Chennai (MAA)')]**. First 'Chennai' element match is in 'TO' dropdown. Which is the other one? The other one (hiding) is in the 'FROM' dropdown. So basically, selenium is trying to select 'Chennai' from the 'FROM' field (since selenium scans the page from left to right) instead of selecting 'Chennai' from 'TO' dropdown. Since 'Chennai' gets hidden after 'Bengaluru' is selected in the FROM field, selenium throws 'ElementNotVisibleException'. Selenium stops any scanning further.

So how to solve this problem? To solve this problem, we have to tell selenium some additional information along with custom xpath **//a[contains(text(),'Chennai (MAA)')]**

That additional information would be the xpath of 'TO' dropdown. Click 'TO' dropdown and inspect entire window. Below we have xpath of entire parent window
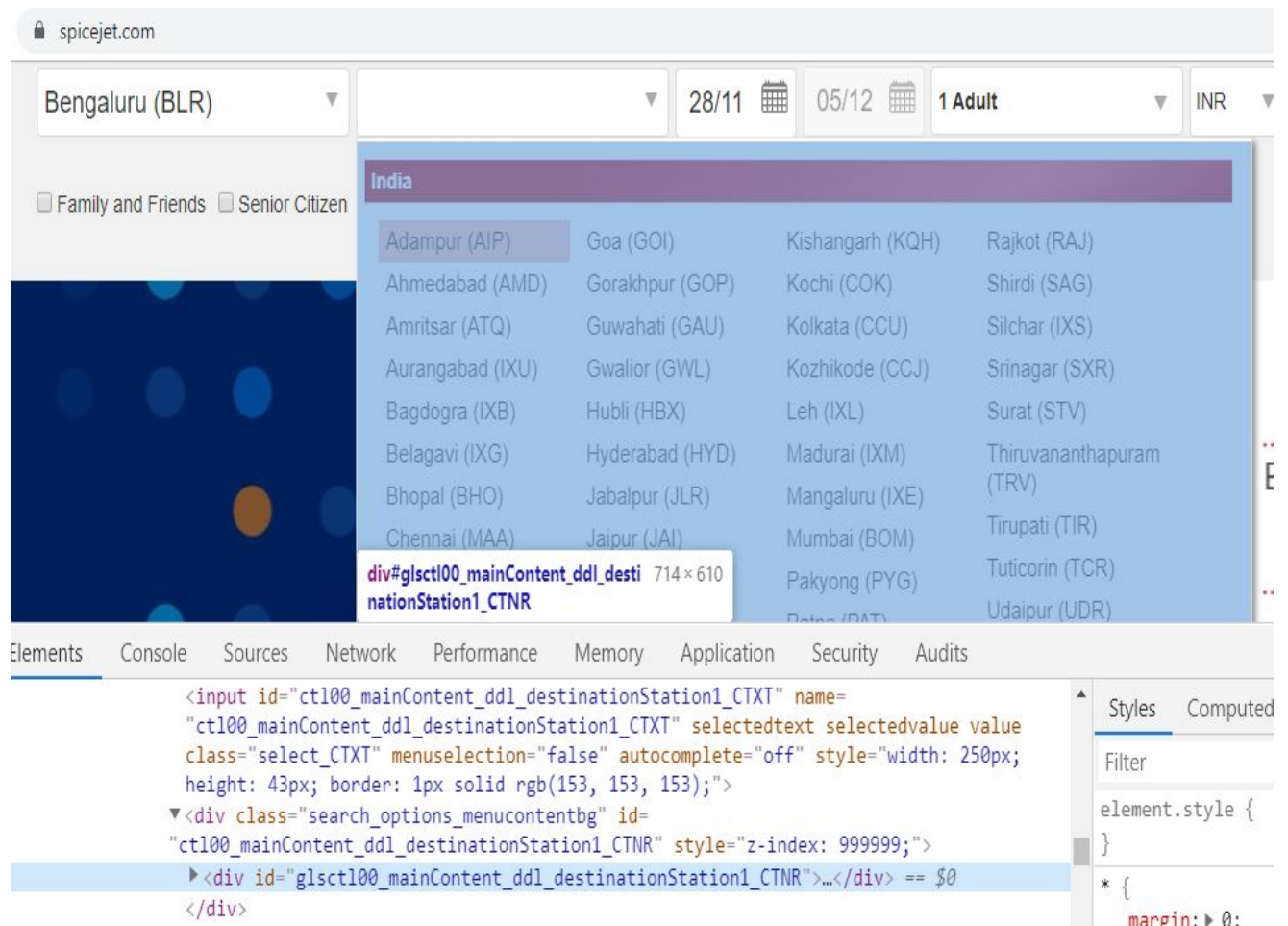**//div[@id='glsctl00_mainContent_ddl_destinationStation1_CTNR']**

Figure 24

Now concatenate the xpath of 'TO' dropdown with the xpath of 'Chennai'. Add one whitespace in between 2 xpaths:

//div[@id='glsctl00_mainContent_ddl_destinationStation1_CTNR'] //a[contains(text(),'Chennai (MAA)')]

Copy/paste this entire xpath in chropath & hit enter. Only 1 matching node will be found this time instead of 2. So, this time, the scope of search is restricted to only 'TO' dropdown area. Now selenium knows which box it should look for to search 'Chennai'. This is called parent-child relationship
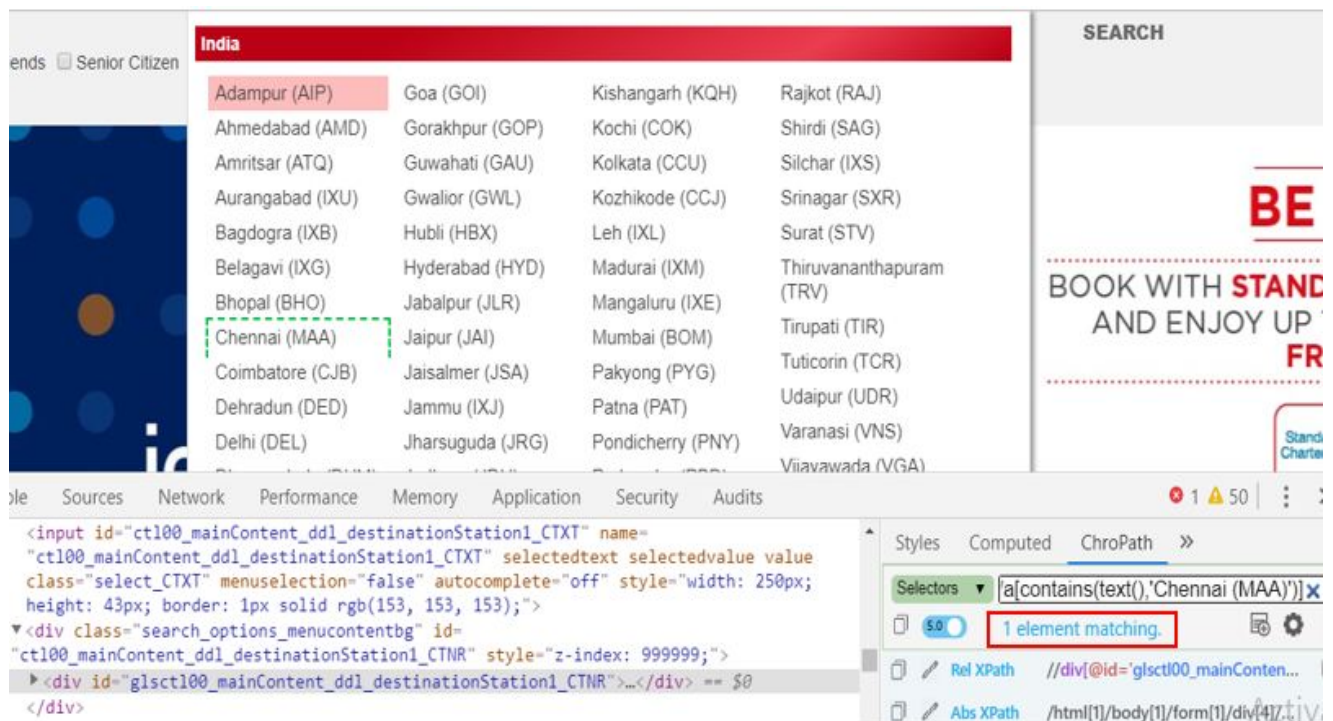
Figure 25

Comment line 20 and add line 21


```
15 //step b: Click 'Departure city' dropdown
16 driver.findElement(By.xpath("//input[@id='ctl00_mainContent_ddl_originStation1_CTXT']")).click();
17 //step c: Select 'Bengaluru (BLR)'from departure city
18 driver.findElement(By.xpath("//a[contains(text(),'Bengaluru (BLR)')]")).click();
19 //step d: Select 'Chennai (MAA)' from arrival city
20 //driver.findElement(By.xpath("//a[contains(text(),'Chennai (MAA)')]")).click();
21 driver.findElement(By.xpath("//div[@id='glsctl00_mainContent_ddl_destinationStation1_CTNR'] //a[contains(text(),'Chennai (MAA)')]")).cli
```

Figure 26

Run the script, notice that 'Chennai' gets selected this time in the 'TO' dropdown
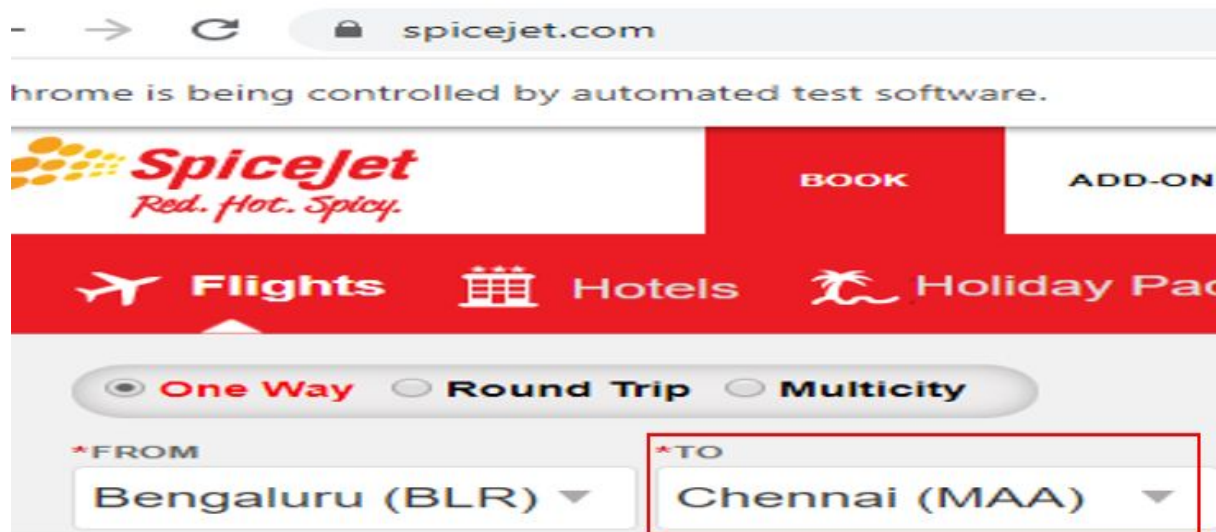
Figure 27

We can write

```
//a[contains(text(),'Chennai (MAA)')]
```

As

```
//a[@value='MAA']
```

Hence comment line#22, add line#23

```
12 driver = new ChromeDriver();
13 //step a: launch site
14 driver.get("https://spicejet.com");
15 //step b: Click 'Departure city' dropdown
16 driver.findElement(By.xpath("//input[@id='ctl00_mainContent_ddl_originStation1_CTXT']")).click();
17 //step c: Select 'Bengaluru (BLR)'from departure city
18 driver.findElement(By.xpath("//a[contains(text(),'Bengaluru (BLR)')]")).click();
19 //step d: Select 'Chennai (MAA)' from arrival city
20 //driver.findElement(By.xpath("//a[contains(text(),'Chennai (MAA)')]")).click();
21 //driver.findElement(By.xpath("(//a[@value='MAA'])[2]")).click();
22 //driver.findElement(By.xpath("//div[@id='glsctl00_mainContent_ddl_destinationStation1_CTNR'] //a[contains(text(),'Chennai
23 driver.findElement(By.xpath("//div[@id='glsctl00_mainContent_ddl_destinationStation1_CTNR'] //a[@value='MAA']")).click();
```

Figure 28

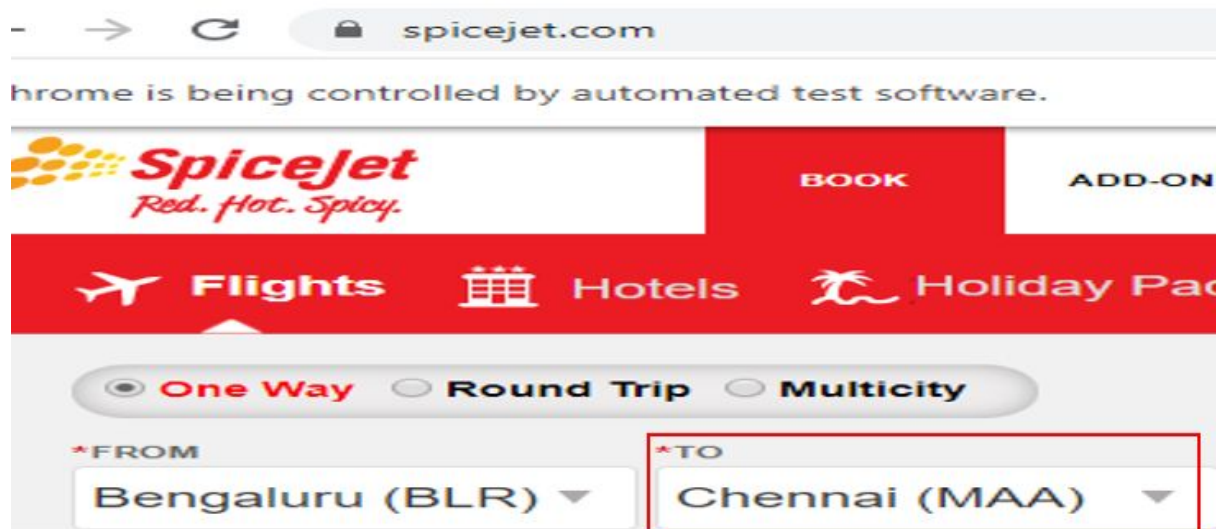Run the script, notice that 'Chennai' gets selected this time in the 'TO' dropdown

Figure 29

Another alternate is: Inspect Chennai in TO dropdown. Notice that it is represented by 'a' tag having the value 'MAA'
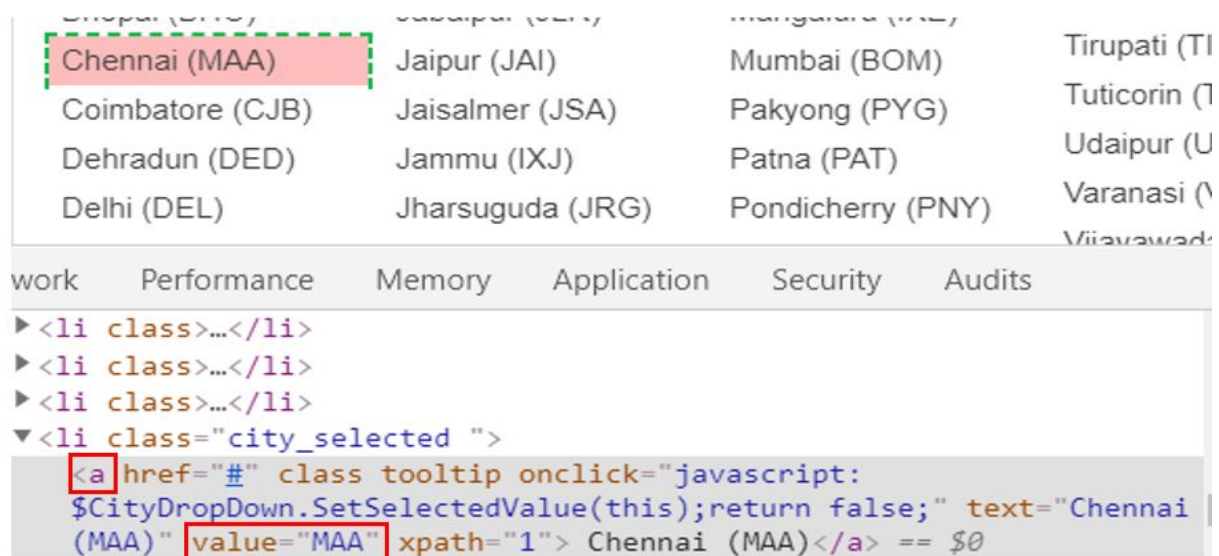

Figure 30

Comment line#23, add a new line#21 using xpath **(//a[@value='MAA'])[2]**. By [2], we are asking selenium to select the 2nd instance of Chennai in TO dropdown (the first instance would be in FROM dropdown)

```
12          driver = new ChromeDriver();
13          //step a: launch site
14          driver.get("https://spicejet.com");
15          //step b: Click 'Departure city' dropdown
16          driver.findElement(By.xpath("//input[@id='ctl00_mainContent_ddl_
17          //step c: Select 'Bengaluru (BLR)'from departure city
18          driver.findElement(By.xpath("//a[contains(text(),'Bengaluru (BLR)
19          //step d: Select 'Chennai (MAA)' from arrival city
20          //driver.findElement(By.xpath("//a[contains(text(),'Chennai (MAA)
21          driver.findElement(By.xpath("(//a[@value='MAA'])[2]")).click();
```

Figure 31

Run the script, notice that 'Chennai' gets selected this time in the 'TO' dropdown
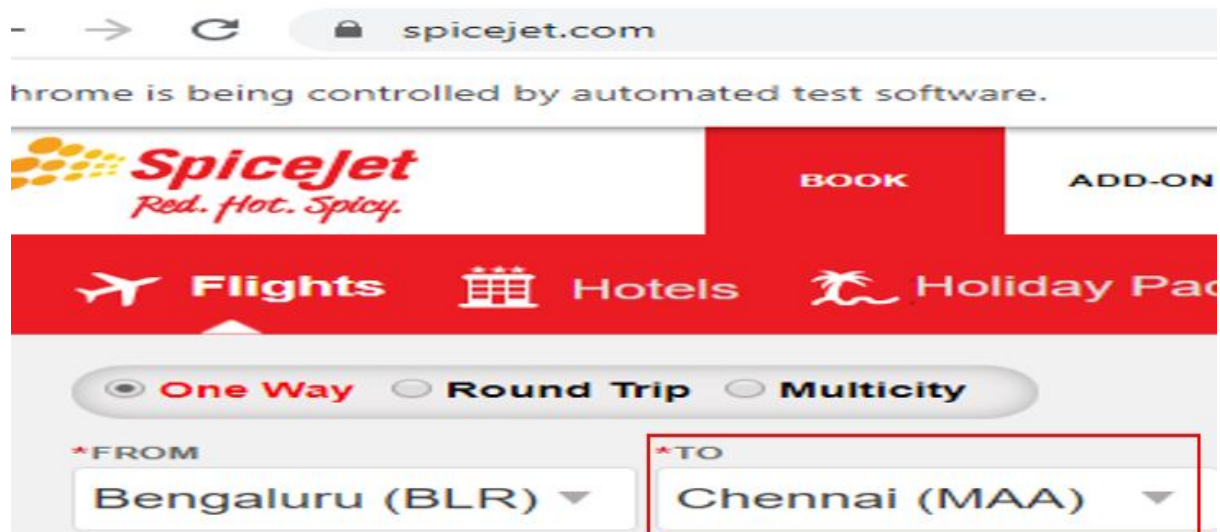

Figure 32

So this is how we handle the static and dynamic dropdowns. Thank you for reading!