**Assisted Practice: 1.4 Handling Various Web Elements**

This section will guide you to:

* Handling Various Web Elements present on the page.

This lab has divided into different types, namely:

1.4.1 Edit box

1.4.2 Link

1.4.3 Button

1.4.4 Image, image link, an image button

1.4.5 Text area

1.4.6 Checkbox

1.4.7 Radio button

1.4.8 Dropdown list

1.4.9 Web table /HTML table

1.4.10 Frame

### 1.4.11 Switching between tabs in same browser window

1.4.12 Pushing the code to GitHub repositories

**Step 1.4.1:** Edit box

* Open Eclipse
* It is a basic text control that enables a user to type a small amount of text.
* Operations on Edit box
  + Enter a Value,
  + Clear the Value,
  + Check enabled status,
  + Check edit box existence,
  + Get the value

**Step 1.4.2:** Link

* link is more appropriately referred to as a hyperlink and connects one web page to another. It allows the user to click their way from page to page.
* Operations on Link
  + Click Link,
  + Check the link existence,
  + Check the link enabled status,
  + Return the Link Name

**Step 1.4.3:** Button

* This represents a clickable button, which can be used in forms and places in the document that needs a simple, standard button functionality.
* Operations on Button
  + Click
  + Check Enabled status
  + Display status

**Step 1.4.4:** Image, image link, an image button

* It helps in performing actions on images like clicking on the image link or the image button, etc.
* Operations Image
  + Three types of Image elements in Web Environment
  + General Image (No functionality)
  + Image Button (Submits)
  + Image Link (Redirects to another page/location)

**Step 1.4.5:** Text area

* It is an inline element used to designate a plain-text editing control containing multiple lines.
* Return / Capture Text Area or Error message from a web page

**Step 1.4.6:** Checkbox

* This is a selection box or a tick box which is a small interactive box that can be toggled by the user to indicate an affirmative or a negative choice.
* Operations on Check box
  + Check if the check box is displayed or not?
  + Check if the check box is enabled or not?
  + Check if the check box is Selected or not?
  + Select the Check box
  + Unselect the Check box

**Step 1.4.7:** Radio button

* It is an option button which is a graphical control element that allows the user to choose only one predefined set of mutually exclusive options.
* Operations on Radio Button
  + Select Radio Button
  + Verify if the Radio Button is Displayed or not?
  + Verify if the Radio Button is enabled or not?
  + Verify if the Radio Button is Selected or not?
* Example:

oRadioButton.get(1).click();

**Step 1.4.8:** Dropdown list

* It is a graphical control element, similar to the list box, which allows the user to choose one value from the list. When this drop-down list is inactive, it displays only a single value.
* Operations on Drop down list
  + Check the Dropdown box existence
  + Check if the Drop down is enabled or not?
  + Select an item
  + Items Count
* Example:

Select fruits = new Select(driver.findElement(By.id("fruits")));

fruits.selectByVisibleText("Banana");

fruits.selectByIndex(1);

**Step 1.4.9:** Web table /HTML table

* Operations on Web table /HTML Table
  + Get cell value
  + Rows Count
  + Cells Count

**Step 1.4.10:** Frame

* Operations on Frame
  + Switch from Top window to a frame
  + Switch from a frame to Top window
* Example
  + driver.switchTo().frame("iframe1");
  + driver.switchTo().frame("id of the element");

**Step 1.4.11:** Switching between tabs in same browser window

* Operations on Switching between tabs in same browser window
* Open a new tab using Ctrl + t
* Driver control automatically switches to the newly opened tab
* Perform the required operations here.
* Next switch back to the old tab using Ctrl + Tab. You need to keep pressing this unless you reach the desired tab.
* Once the desired tab is reached, then perform the operations in that tab.
* Example:

driver.switchTo().window(tabs2.get(1));

driver.switchTo().window(tabs2.get(0));

**Step 1.4.12:** Pushing the code to your GitHub repositories

* Open your command prompt and navigate to the folder where you have created your files.

**cd <folder path>**

* Initialize your repository using the following command:

**git init**

* Add all the files to your git repository using the following command:

**git add .**

* Commit the changes using the following command:

**git commit . -m “Changes have been committed.”**

* Push the files to the folder you initially created using the following command:

**git push -u origin master**