3. Locating Elements through CSS and XPath

1. To find the element present on the page using CSS Selector.

- Using CSS Selectors in Selenium. As we all know, CSS stands for Cascading Style Sheets. By using CSS selectors, we can find or select HTML elements on the basis of their id, class or other attributes. CSS is faster and simpler than XPath particularly in case of IE Browser where Path works very slowly.
- Open Eclipse
- Use Path as a CSS Selector
- CSS Selector have many formats, namely
    a. **Tag and ID**
        - Syntax: “css = tag#id”
        - Example: driver.findElement(By.cssSelector(“input#email”));

    b. **Tag and Class**
        - Syntax: “css = tag.class”
        - Example: driver.findElement(By.cssSelector(”input.inputtext”));

    C. **Tag and Attribute**
        - Syntax: “css = tag[attribute=value]”
        - Example: driver.findElement(By.cssSelector(“input[name=lastName]”));

    c. **Tag, Class and Attribute**
        - Syntax: “tag.class[attribute=value]”
        - Example:
          driver. findElement(By.cssSelector(“input.inputtext[tabindex=1]”));

    d. **Inner text**
        - Syntax: “css = tag.contains(“innertext”)”
        - Example: driver.findElement(By.cssSelector(font:contains(“Boston”)));

2. To find the element present on the page using Path.

- In Selenium automation, if the elements are not found by the general locators like id, class, name, etc. then XPath is used to find an element on the web page.
- XPath contains the path of the element situated at the web page. Standard syntax for creating XPath is:
  XPath=//tagname[@attribute='value']

    - **//:** Select current node.
    - **Tagname:** Tagname of the particular node.
    - **@:** Select attribute.
    - **Attribute:** Attribute name of the node.
    - **Value:** Value of the attribute.

- Types of XPath:
  There are two types of XPath:
  a. **Absolute XPath**
     - It is direct way to find the element, but the disadvantage of the absolute XPath is that if there are any changes made in the path of the element then that XPath gets failed.
     - The key characteristic of XPath is that it begins with the single forward slash (/), which means you can select the element from the root node.
     - Syntax for absolute Path: html/body/div[1]/div[1]/div/h4[1]/b
     - Example:
       driver.findElement(By.xpath("html/body/div[1]/div[1]/div/h4[1]/b"));
     - Writing absolute XPath on the elements which are present in the webpage will be very lengthy. To reduce the length, we use relative XPath.

  b. **Relative XPath**
     - For relative XPath the path starts from the middle of the HTML DOM structure. It starts with the double forward slash (//), which means it can search the element anywhere at the webpage.
     - You can start from the middle of the HTML DOM structure and no need to write long XPath.
     - Syntax for relativeXPath: //*[@class='relativexapath']
     - Example: driver.findElement(By.xpath("//*[@class='relativexapath']"))

Pushing the code to your GitHub repositories : -

- Open your folder where the Project . And then click the right button to open the git bash command prompt.
- Before that, open the github and create a new repository.
- Initialize your repository using the following command:

git init

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

git add .

- To check the status of the repository use the below command:

git status

- Commit the changes using the following command:

git commit . -m "Changes have been committed."

- To add the files to the repository use the (URL) from the github and use the command;

git remote add origin <url>

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

git push origin master.