**What is Automation Testing ?**

Automation testing, is a software testing technique that uses software tools to execute tests on a software application, system, or product.

**Selenium:**

* It is a open source (free for download)
* Automated testing tool,
* To test a “Web Application”.
* Selenium consists of Test suit (tools & components)

**Advantages:**

* It supports wide varieties of “browsers and languages”.
* Integrated with CICD tool (like Jenkins).
* It is a open source.

**Disadvantages:**

* Selenium is used for only test the web application.

**Selenium test suit tools or components:**

1. Selenium IDE (Integrated Development Environment ),

It is a record and play back tool. Here chrome and fire fox were plugin. We can use this tool without any coding knowledge. It will automatically generate the test script.

1. Selenium RC (Remote Control),

Currently selenium RC not in use.

1. Selenium Web Driver,

Selenium Web Driver is a advanced version of selenium RC. In selenium web driver we can interact with web elements.

1. Selenium Grid,

In Selenium Grid have parallel interaction with OS.

**TestNG Framework**:

* TestNG – Next Generation,
* For generate the report,
* Parallel execution.

**Difference between Manual and Automation testing:**

* In automation testing, Human errors are less compared to manual testing,
* Duration time less in automation testing,
* Manpower range is less compared to manual testing.

**Architecture of selenium:**

1. Selenium 3 – data transfer in 3 – JSON Wire Protocol. (Older version)
2. Selenium 4 – data transfer in 4 – W3C Protocol. (New Version).

**Initializing Web Browser:**

* .get(); - get is a method which is used for url up, (to launch the web application).
* .manage().window().maximize(); - This code is used for maximize the screen.

**Browser commands:**

* .getCurrentUrl() - This method is used for get the url.
* .getTitle() - This method is used to get the title of the web application.
* .getWindowHandle() - This method is used to get the window handle ID.
* .getPageSource() –

**Close & Quit:**

1. close() - It helps to close a single web page
2. quit() – This command helps to close all the related web page.

**Navigation Commands:**

* .navigate().to() – This method helps to open another web application.
* .navigate().back() – This method helps to came back to the previous web application.
* .navigate().forward() – This method is used to go forward to the web application.
* .navigate().refresh() – This method helps to refresh the screen.

**Locaters:**

1. ID, - Finds element by it’s unique ID attribute
2. Class name, - Finds element by it’s class attribute.
3. Name, - Finds element by it’s name attribute.
4. Tag name, - Finds element by it’s HTML tag name.
5. Link text, - Finds link by it’s by exact visible text.
6. Partial link text, - Finds a link by partial visible text.
7. cssSelector, - Finds element by cssSelector

Syntax for cssSelector: tagname[attribute= ‘value’]

1. x path – Finds element using on xpath expression.

**What is xpath in selenium ?**

Xpath is a language used for navigating through elements and attributes in an XML doc. In selenium, xpath is used to locate elements in HTML documents. Xpath can be absolute or relative.

* Absolute XPath: /html/body/div/div……..
* Relative XPath: //tagname[@attribute=’value’].

Without attribute XPath syntax: (using text)

//tagname[text()=’ text ‘]

In the case of (For bigger text)

//tagname[starts-with(text(), ‘ partial text ‘].

and syntax:

//tagname[@attribute= ‘value’ and@attribute= ‘value’]

or syntax:

//tagname[@attribute= ‘value’ or@attribute= ‘value -any word’]

**XPath axes method:**

* Child tag,

//tagname[@attribute = ‘value’]//child::tagname[@ attribute = ‘value’]

* Parent tag,

//tagname[@attribute = ‘value’]//parent::tagname[@ attribute = ‘value’]

* Preceding tab,

//tagname[@attribute = ‘value’]//preceding::tagname[@ attribute = ‘value’]

* Ancestor tag,

//tagname[@attribute = ‘value’]//ancestor::tagname.

* Following tag,

//tagname[@attribute = ‘value’]//following::tagname[@ attribute = ‘value’]

* Descendant.

//tagname[@attribute = ‘value’]// descendant::tagname.

**Web element commands:**

1. sendKeys() – To input the value.
2. click() – To click the button.
3. getText() – for confirmation purpose.
4. getCssValue() - To get the cssvalue of the function.
5. clear() – To clear the value.
6. getSize() – To get the size of the button.
7. getTagName() – To get the tag name.

**DropDown:**

1. select method
   * + selectByIndex
     + selectByValue
     + selectByVisibleText.

**Handling Radio Button and Check Box:**

* click(),

**Keyboard Action:**

keypress – To press the button.

keyRelease – To release the button.

Robot – It’s a predefined class

VK – virtual key,

keyEvent – predefined class.

**Frames:**