

Program 1

Aim : Understanding of Test case, Test Scenario and Test report

Procedure :-

Test Case : A test case is a set of actions performed on a system to determine if it satisfies software requirements and functions correctly.

Test scenarios : Scenario Testing in software testing is a method in which actual scenarios are used for testing the software application instead of test cases .

Now there are some steps which is used in the test scenario and the test case

Step 1 : Test case id and Test Case Description

Test Case Id	Test Case Description
--------------	-----------------------

Step 2 : Test case id , Test Case Description , Test Case Data

Test case id	Test Case Description	Test data
--------------	-----------------------	-----------

Step 3 : Test case id , Test Case Description , Test Data ,Performing Action , Test Steps

Test case id	Test case description	Test steps	Test data
--------------	-----------------------	------------	-----------

Step 4:- Test case id , Test Case Description , Test Case Data ,Performing Action , Test Steps ,Expected output ,actual output ,status , Defective.

Test scenario id	Test scenario Description	Test case id	Test case description	Test Steps	Test data	Expected output	Actual output	Status	Defects
------------------	---------------------------	--------------	-----------------------	------------	-----------	-----------------	---------------	--------	---------

PROGRAMME 2

AIM : Design the test case for the Arithmetic Calculations.

Project Name	Arithmetic Calculation									
Module Name	Multiplication									
Created By	Gaurav Kumar									
Reviewed By	Bina Sanchan									
TS ID	TEST SCENARIO	TEST CASE ID	TEST CASE DESCRIPTION	TEST STEPS	TEST DATA	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS	DEFECT	Executed By
TS_03	Verify Multiplication	TS_3_01	Multiplication of two positive integer	enter first positive number enter second positive number	70 80	5600	5600	PASS	-	Gaurav Kumar
		TS_3_02	Multiplication of two negative integer	enter first negative number enter second negative number	-10, -100	11000	11000	PASS	-	Gaurav Kumar
		TS_3_03	Multiplication of one positive one negative integer	enter first negative enter second positive number	40,-2	-80	-80	PASS	D001	Gaurav Kumar
Project Name	Arithmetic Calculation									
Module Name	Division									
Created By	Gaurav Kumar									
Reviewed By	Bina Sanchan									
TS ID	TEST SCENARIO	TEST CASE ID	TEST CASE DESCRIPTION	TEST STEPS	TEST DATA	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS	DEFECT	Executed By
TS_04	Verify Division	TS_4_01	Division of two positive integer	enter first positive number enter second positive number	70 7	10	5600	PASS	-	Gaurav Kumar
		TS_4_02	Division of two negative integer	enter first negative number enter second negative number	-1000, -100	10	11000	PASS	-	Gaurav Kumar
		TS_4_03	Division of one positive one negative integer	enter first negative enter second positive number	40,-2	-20	-80	PASS	-	Gaurav Kumar
		TS_4_04	Division of a zero by any integer	enter zero enter any integer	0,67	0	67	Fail	D001	Gaurav Kumar
		TS_4_05	Division of any integer by zero	enter any integer enter zero	67, 0	Not Defined	0	Fail	D002	Gaurav Kumar

PROGRAME 3

AIM : Design the test case for the sorting of n numbers.

Project Name	Sorting								
Module Name	Sorting								
Created By	Gaurav Kumar								
Reviewed By	Bina Sanchan								
Test Scenario ID	Test Scenario Description	Test Case ID	Test Case Description	Test Steps	Test Data	Expected Result	Actual Result	Status	Executed By
TS_001	Check the sorting on integer numbers	TS_1_001	Check the sorting between N numbers	1. Enter the list of integer numbers. 2. Display the result.	numbers= 2,4,3,1	Result = 1,2,3,4	Result = 1,2,3,4	Pass	Gaurav Kumar
		TS_1_002	Check the sorting between N numbers	1. Enter the list of integer numbers. 2. Display the result.	numbers= -6,-2,-4,-3,-1	Result = -1,-2,-3,-4,-6	Result = -1,-2,-3,-4,-6	Pass	Gaurav Kumar
		TS_1_003	Check the sorting between N numbers	1. Enter the list of integer numbers between the range of 1 to 10. 2. Display the result.	numbers= 1,2,3,4,5,6,7,8,9,10	Result = 1,2,3,4,5,6,7,8,9,10	Result = 1,2,3,4,5,6,7,8,9,10	Pass	Gaurav Kumar
TS_002	Check the sorting on floating numbers	TS_2_001	Check the sorting between N numbers	1. Enter the list of floating numbers. 3. Display the result.	numbers= 1.8,1.6,1.3,1.4,1.7, 1.0	Result = 1.0, 1.3, 1.4,1.6, 1.7, 1.8	Result = 1.0, 1.3, 1.4,1.6, 1.7, 1.8	Pass	Gaurav Kumar
		TS_2_002	Check the sorting between N numbers	1. Enter the list of floating numbers. 3. Display the result.	numbers= -6.6,-2.2,-4.4,-3.3,-1.1	Result = -1.1,-2.2,-3.3,-4.4,-6.6	Result = -6.6,-4.4,-3.3,-2.2,-1.1	Fail	Gaurav Kumar
TS_003	Check the sorting on both integer and floating numbers	TS_3_001	Check the sorting between N numbers	1. Enter the list of numbers which contains both integer and float numbers. 3. Display the result.	numbers= 2,5,1.1,7	Result = 1.1,2,5,7	Result = 2,5,7,1.1	Fail	Gaurav Kumar
		TS_3_002	Check the sorting between N numbers	1. Enter the list of numbers which contains both integer and float numbers. 3. Display the result.	numbers= -9,2,5,1.1,-7.0	Result = -7.0, -9, 1.1,2,5	Result = 1.1,2,5,-7.0,-9	Fail	Gaurav Kumar

PROGRAMME 4

AIM : Develop a Login form and prepare test case report

PROGRAME 5

AIM : Develop a Student mark sheet application and prepare test case report.

Project Name	Student Marksheets Application									
Module Name	Student Marksheets Application									
Created By	Gaurav Kumar									
Reviewed By	Bina Sanchan									
TS ID	TEST SCENARIO	TEST CASE ID	TEST CASE DESCRIPTION	TEST STEPS	TEST DATA	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS	DEFECT	TESTER NAME
MODULE: UI CHECKING										
TS_01	Verify user interface works properly.	TS_1_01	check all the text boxes, and buttons.	check the functionality of page.	-	UI should be perfect Text boxes and buttons should be aligned	UI should be perfect Text boxes and buttons should be aligned	PASS	-	Gaurav Kumar
MODULE: STUDENT PORTAL										
TS_02	Verify user ID and show the marksheets.	TS_2_01	check the date of birth,user's roll number,and captcha code. Everything matches to the database then shows the user's marksheets.	1. input the correct DOB 2. enter the valid user's login id or roll number. 3.insert the correct captcha code. 4.press submit button	07/01/2002 RA00097 59762	marksheets displayed	marksheets displayed	PASS	-	Gaurav Kumar
		TS_2_02		1. input the incorrect DOB 2. enter the valid user's login id or roll number. 3.insert the correct captcha code. 4.press submit button	12/6/2001 RA00097 59762	data not found you enter the incorrect DOB	data not found you enter the incorrect DOB	PASS	-	Gaurav Kumar
		TS_2_03		1. input the correct DOB 2. enter the invalid user's login id or roll number. 3.insert the correct captcha code. 4.press submit button	07/01/2002 RA200022 59762	login id not found.	login id not found.	PASS	-	Gaurav Kumar
		TS_2_04		1. input the correct DOB 2. enter the valid user's login id or roll number. 3.insert the incorrect captcha code. 4.press submit button	07/01/2002 RA00097 59762	try again later you entered the incorrect captcha code	try again later you entered the incorrect captcha code	PASS	-	Gaurav Kumar

EXPERIMENT 6

AIM : Design Test Case For Triangle problem

Project Name	Traingle Problem											
Created By	Gaurav Kumar											
Reviewed By	Beena Sachan											
Reviewed Date												
TS ID	TEST SCENARIO	TEST CASE ID	TEST CASE DESCRIPTION			TEST STEPS	TEST DATA	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS	DEFECT	Executed By
TS_01	Verify Equilateral Traingle	TS_01_01	Verify when all three sides are entered of equal size			enter all three sides	10,10,10	Equilateral traingle	Equilateral traingle	Pass	_	Gaurav Kumar
		TS_01_02	Verify when only two of the sides are equal			enter all three sides	10,10,11	Not a Equilateral traingle	Equilateral traingle	Fail	D_01	Gaurav Kumar
		TS_01_03	Verify when none of the sides are equal			enter all three sides	5,7,3	Not a Equilateral traingle	Not a Equilateral traingle	Pass	_	Gaurav Kumar
TS_02	Verify Isoceles Traingle	TS_02_01	Verify when all three sides are entered of equal size			enter all three sides	10,10,10	Not an Isoceles traingle	Not an Isoceles traingle	Pass	_	Gaurav Kumar
		TS_02_02	Verify when only two of the sides are equal			enter all three sides	10,10,11	Isoceles traingle	Not an Isoceles traingle	Pass	_	Gaurav Kumar
		TS_02_03	Verify when none of the sides are equal			enter all three sides	5,7,3	Not an Isoceles traingle	Isoceles traingle	Fail	D_02	Gaurav Kumar
TS_03	Verify Scalane Traingle	TS_03_01	Verify when all three sides are entered of equal size			enter all three sides	10,10,10	Not a scalane triangle	Not a scalane triangle	Pass	_	Gaurav Kumar
		TS_03_02	Verify when only two of the sides are equal			enter all three sides	10,10,11	Not a scalane triangle	scalane triangle	Fail	D_03	Gaurav Kumar
		TS_03_03	Verify when none of the sides are equal			enter all three sides	5,7,3	scalane triangle	scalane triangle	Pass	_	Gaurav Kumar

Experiment - 7

Aim: Design the test case for Binary Search.

Project Name : Binary Search

Module Name : Binary Search

Created By : Gaurav Kumar

Reviewed By : Bina Sanchan

Test Scenario Id	Test Scenario Description	Test Case Id	Test Case Description	Test Steps	Test Data	Expected Output	Actual Output	Status	Defect	Executed By
Ts_01	Verifying Integer List	Ts_01_1	Finding index of a integer from the list	Enter the list Enter the element for searching.	1,2,3,4,5,6,7,8,9,5	Index 4	Index 4	Pass	-	Gaurav Kumar
		Ts_01_2	Finding index of a integer from the list	Enter the list Enter the element for searching.	1,2,3,4,6,7,8,9,10,11,5	Shows Error "Number not found"	Showed Error "Number Not Found"	Pass	-	Gaurav Kumar
		Ts_01_3	Finding index of a integer from the list	Enter the list Enter the element for searching.	-1,-2,-3,-4,-5,-6,-7,8,-6	Index 6	Index 6	Pass	-	Gaurav Kumar
		Ts_01_4	Finding index of a integer from the list	Enter the list Enter the element for searching.	-1,-2,-3,-4,-5,-6,-7,9,-8	Shows Error "Number not found"	Showed Error "Number Not Found"	Pass	-	Gaurav Kumar
Ts_02	Verifying Float List	Ts_02_1	Finding index of a float from the list	Enter the list Enter the element for searching.	1.1,1.2,1.3,1.4,1.45,2.56,3.45,1.1	Index 0	Index 0	Pass	-	Gaurav Kumar
		Ts_02_2	Finding index of a float from the list	Enter the list Enter the element for searching.	1.12,1.2,1.3,1.4,1.45,2.56,3.45,1.2	Shows Error "Number not found"	Showed Error "Number Not Found"	Pass	-	Gaurav Kumar
		Ts_02_3	Finding index of a float from the list	Enter the list Enter the element for searching.	-1.1,-1.12,-1.13,-2.1,-2.12,-2.13,-3.1,-3.12	Index 7	Index 7	Pass	-	Gaurav Kumar

EXPERIMENT 8

AIM : Develop a Employee salary processing application and prepare test case report

Project Name	Employee salary									
Created By	Gaurav Kumar									
Reviewed By	Beena Sachan									
Review date										
Test_Scenario_ID	Test_Scenario	Test_Case_Id	Test_Case_Description	Test Steps	Test Data	Expected Output	Actual Output	Status	Defect	Executed By
TS_01	Verify all UI controls visibility.	TS_01_01	Test all textfields and buttons are visible and aligned	NA	NA	Each control should be visible	Each control is visible	Pass	-	Gaurav Kumar
	check all UI controls accessibility	TS_01_02	Test all textfields and buttons are accessible	NA	NA	Each control should be accessible	Each control is accessible	Pass	-	Gaurav Kumar
TS_02	Verify Salary Calculation	TS_02_01	verify salary calculation when all entered details are correct	enter employee id enter total working days enter wage per day click on calculate button	eo1 30 1000	30000	30000	Pass	-	Gaurav Kumar
		TS_02_02	verify salary calculation when few details are missing	enter employee id enter total working days enter wage per day click on calculate button	eo1 30	missing data	missing data	Pass	-	Gaurav Kumar
		TS_02_03	verify salary calculation when few details are missing	enter employee id enter total working days enter wage per day click on calculate button	eo1 30	missing data	30000	fail	D_01	Gaurav Kumar

EXPERIMENT 9

AIM : Develop a flight Reservation system and prepare test case report

Project name	Flight Reservation system										
Created by	Gaurav Kumar										
Reviewed by	Beena Sachan										
Review Date											
Test Scenerio id	Test scenerio description	Test case id	Test case description	Test Steps	Test data	Expected result	Actual result	test result	defect number	Executed by	
Ts_01	verify login activity	Ts_01_01	verify if a user will be able to login with valid Sid and password.	enter user id enter password	roy@123 Gaurav	successfully login	successfully login	PASS	—	Gaurav Kumar	
		Ts_01_02	verify if a user will be able to login with valid id and invalid password	enter user id enter password	roy@123 Gaurav	invalid id or password	invalid id or password	PASS	—	Gaurav Kumar	
		Ts_01_03	verify if a user will be able to login with invalid id and valid password	enter user id enter password	Abc@123 Gaurav	invalid id or password	invalid id or password	PASS	—	Gaurav Kumar	
		Ts_01_04	verify if a user will be able to login with invalid id and password	enter user id enter password	Abc@123 Gaurav	invalid id and password	successfully login	FAIL	d_01	Gaurav Kumar	
Ts_02	verify application for showing available flights	Ts_02_01	verify if a user is able to get details of available flights if flights are available	enter source enter destination	Patna Delhi	flights available	flights available	PASS	—	Gaurav Kumar	
		Ts_02_02	verify if a user is able to get details of available flights if flights are not available	enter source enter destination	Patna Mumbai	flight not available	flights not available	PASS	—	Gaurav Kumar	
Ts_03	Verify that after selecting seats, entering passenger details and making payment the selected seats get booked.	Ts_03_01	verify after filling all details and paymets, ticket booked	enter details select seat make payment	G roy Sno 5 payed \$200	seat booked	seat Booked	PASS	—	Gaurav Kumar	
		Ts_03_02	verify after filling all details and without paymets, ticket booked	enter details select seat make payment	Gaurav R Sno8 —	seat not booked make payment first	seat Booked	FAIL	d_02	Gaurav Kumar	
Ts_04	Verify that the user receives confirmation mail along with tickets on the emaiilds provided.	Ts_04_01	verify that the ticket sent to provided email after ticket booked.	book ticket	ticket booked	ticket sent to email	ticket sent to email	PASS	—	Gaurav Kumar	
Ts_05	verify that after canceling the ticket, the ticket is cencled or not.	Ts_05_01	cancel the ticket	Enter reason for cenceling	any reason	Ticket cencled and payment reflected on linked account	Ticket cencled and payment reflected on linked account	PASS	—	Gaurav Kumar	
				click on candle button	press cancel						

EXPERIMENT 10

Aim: To Perform the record and playback using Selenium IDE

Procedure:

Recording Process:

1. Launch Firefox and open Selenium IDE from Tools Menu .
2. Ensure that the Selenium IDE is launched.
3. After the Selenium IDE is launched, ensure that the Recording Button is turned on by default. (i.e. Selenium IDE is in recording mode by default on launching).
4. While the recording button is enabled, open any website say www.google.com in the Firefox browser.
5. In the 'Google' page, click on 'I am feeling lucky' button.
6. Turnoff the 'Recording' button on the Selenium IDE to stop recording.
7. Ensure that the recorded commands are displayed under the table tab.

Playback Process:

8. Open any other site say www.yahoo.com in the Firefox browser and Click on the 'Play current test case' button on the Selenium IDE.
9. On clicking the 'Play current test case' button, observe that the recorded commands are executed automatically in the Firefox Browser.

EXPERIMENT 11

Aim: Validation of elements using Assert mechanism in Selenium IDE

Procedure: Assert will check whether the element is on the page, if it is not available, the test will fail the step and won't continue executing the remaining steps.

The following steps will show how assert mechanism validates elements:

Example: Check whether the Google Logo is available on the www.Google.com page. (i.e. The test will pass if the Google Logo exists on the Google page else it fails).

1. Launch Selenium IDE from Firefox Browser -> Tools Menu
2. Ensure that the Selenium IDE is launched and also ensure that 'Record' option is enabled by default.
3. Type www.google.com in the Firefox Address bar and press 'Enter' button on the keyboard
4. Ensure that the Google page is displayed in the Firefox Browser
5. In Google Page, right click on the UI element to be validated i.e Google Logo in this example .
6. In the right click menu options, select 'Show All Available commands' option.
7. Ensure that the sub-menu option for the 'Show All Available commands' option is displayed and select 'AssertElementPresent' option.
8. In Selenium IDE -> Table tab, ensure that assertElementPresent element is added.
9. Click on the 'Record' button to turn off the recording process.
10. Playback or Run the above recorded validation, by click on the 'Play current test case' option on the Selenium IDE.
11. After the test Run, ensure that the 'assertElementPresent' command under the Selenium IDE -> Table tab is displayed in Green color (i.e. Green means the step got passed and the UI element is present i.e. Google Logo in this example is present on the Google Page) .
12. If the element is not present (i.e. Google Logo in this example), then the step will fail and displayed in red color.
13. If in case the test fails as shown in the step 12, all the steps to be executed after the failed step won't be executed.

EXPERIMENT 12

Aim: Validation of elements using verify mechanism in Selenium IDE

Procedure: The following steps will show how verify mechanism validates elements:

Example: Check whether the Google Logo is available on the www.Google.com page. (i.e. The test will pass if the Google Logo exists on the Google page else it fails).

1. Launch Selenium IDE from Firefox Browser -> Tools Menu
2. Ensure that the Selenium IDE is launched and also ensure that 'Record' option is enabled by default.
3. Type www.google.com in the Firefox Address bar and press 'Enter' button on the keyboard
4. Ensure that the Google page is displayed in the Firefox Browser.
5. In Google Page, right click on the UI element to be validated i.e Google Logo in this example .
6. In the right click menu options, select 'Show All Available commands' option .
7. Ensure that the sub-menu options for the 'Show All Available commands' option is displayed and select 'VerifyElementPresent' option
8. In Selenium IDE -> Table tab, ensure that verifyElementPresent element is added .
9. Click on the 'Record' button to turn off the recording process .
10. Playback or Run the above recorded validation, by click on the 'Play current test case' option on the Selenium IDE
11. After the test Run, ensure that the 'verifyElementPresent' command under the Selenium IDE -> Table tab is displayed in Green color (i.e.Green means the step got passed and the UI element is present .
12. If the element is not present (i.e. Google Logo in this example), then the step will fail and displayed in red color .
13. If in case the test fails as shown in the step 12, all the steps to be executed after the failed step will also get executed.

EXPERIMENT 13

Aim: Understanding of working with selenium component(locator) and Selenium Webdriver.

Procedure:

Locator: In Selenium Automation, Locators are used to locate the UI (User Interface) elements of a page like Text Box field, Button etc

Different Types of Locators to identify UI elements

Selenium uses different types of locators to identify the UI elements on the page.

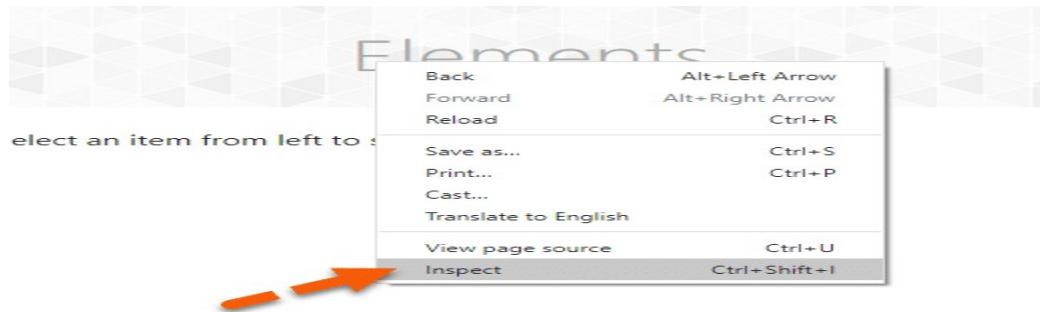
The following is the list of locators based on the priority and recommendations to use:

1. ID
2. Name
3. Class Name
4. Link
5. CSS
6. XPath

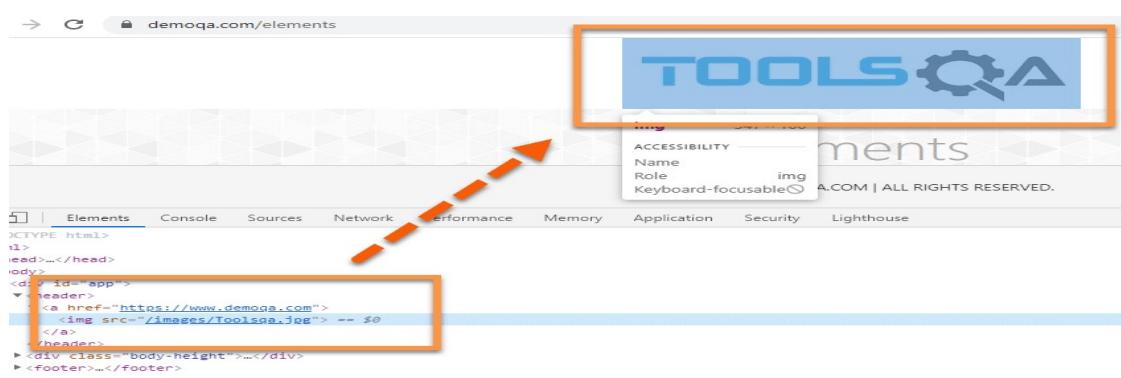
How to locate a web element in DOM?

steps to identify the web element in **Google chrome**.

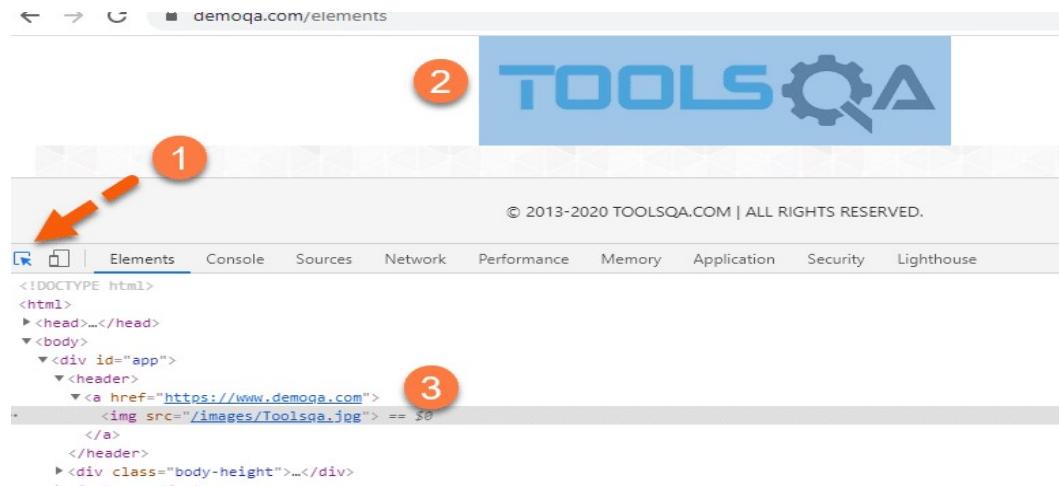
1. **DOM** can be accessed in Google Chrome either by pressing **F12** or by **right click** on the web page and then by selecting **Inspect** (as shown in the screenshot below).



2. Once we click on the "**Inspect option**", it will open the **Developer Tools console**, as shown below. By default, it will open the "**Elements**" tab, which represents the complete **DOM** structure of the web page. Now, if we hover the mouse pointer over the **HTML** tags in the DOM, it will highlight the corresponding elements it represents on the webpage.



3. Now, the main point is, how do we find the web element in the **DOM**. Click on the "**Mouse Icon**" arrow (as designated by **Marker 1** in below screenshot) and then select the web element on the web page. It will automatically highlight the corresponding **HTML** element in the **DOM**. Suppose we want to find the **HTML** elements corresponding to the banner image (as shown below by marker 3). When we select the mouse point and click on the banner image, it will automatically highlight the corresponding **HTML** element, as shown by **marker 2**, in the below screenshot:

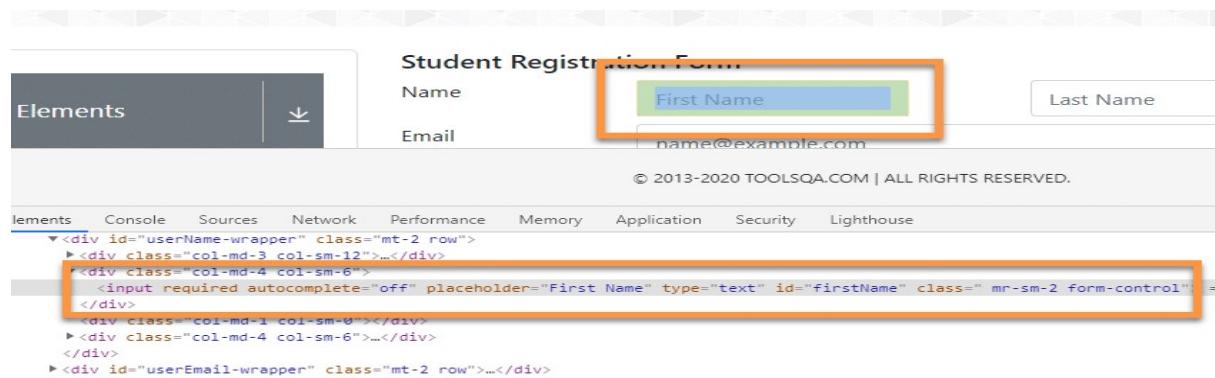


So, this way, we can easily search the **HTML** element in the **DOM** corresponding to a web element on the webpage.

How to locate a web element by using the "id" attribute?

"**ID**" as a locator is one of the most common ways of identifying elements on a web page. According to **W3C**, an ID for a web element always needs to be unique. The **ID** is one of the fastest and unique ways of locating web elements and is considered as one of the most reliable methods for determining an element. But with the advancement in technologies and more of the **dynamic web pages**, "**IDs**" are generated dynamically and generally not the reliable way to locate a web element, as they change for different users.

Suppose we have to locate the text box of First Name on the web page "<https://demoqa.com/automation-practice-form>". When we inspect it in **DOM**, we see the following **DOM** structure:



As we can see, the "**Input**" **HTML** tag has the following properties and attributes:
<input required="" autocomplete="off" placeholder="First Name" type="text" id="firstName" class="mr-sm-2 form-control">

As we can see, the HTML tag contains the attribute “*id*” inside the input tag. The *id* here used is the “**“firstName”**” which we can use to locate this element in the web page. Now, to find the “**“First Name”**” text box on the web page, we can use the following syntax:

```
By.id("firstName");
```

As we can see that we have used the "**By.id()**" method, and we have passed the "**“firstName”**", which is the "*id*" of the text box we are trying to locate. So, this way, we can locate any web element which has a specified "*id*" attribute associated with it.

Selenium WebDriver:

Selenium WebDriver is a set of open-source **APIs**, which provided the capabilities to interact with any of the modern web-browsers and then, in-turn to automate the user actions with that browser. It is an essential component of the **Selenium** family. WebDriver was integrated with *Selenium RC* to overcome a few of the limitations of *Selenium RC*. For any Selenium test script, there are generally the following 7 steps, which apply to all the test cases and all the applications under test (*AUT*):

1. *Create an instance of WebDriver specific to the Browser:*

- Eg: To create an instance of the Firefox driver, we can use the following commands:

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
WebDriver driver = new FirefoxDriver();
```

2. *Navigate to the desired Web page which needs to be automated:*

- Eg: To navigate to the "<https://demoqa.com/text-box>", we can use the following command:

```
driver.get("https://demoqa.com/text-box")
```

3. *Locate an HTML element on the Web page:*

- In order to interact with a web page, we need to locate the HTML elements on the web page. We can use any of the element locator strategies mentioned at "**Selenium Locators**". Eg: if we want to get the "**“Full Name”**" text box, we can use the following commands:

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebElement;
WebElement usernameElement = driver.findElement(By.id("userName"));
```

4. *Perform an action on an HTML element:*

We can perform certain actions on the HTML elements, such as type something using the **SendKeys** method, click on the element if it is a button. Eg: if we want to type the name in the identified text box, we can use the following commands:

```
usernameElement.sendKeys("Ravinder Singh");
```

5. *Run tests and record test results using a test framework.*

And, we are done with using the *WebDriver* to identify and perform the needed actions on the Web Application. Depending on the browser, on which we need to test our application, we can use the corresponding *WebDriver*.

EXPERIMENT 14

Q: Perform the following automation using Selenium WebDriver:

- Launch a new Chrome browser.
- Open Shop.DemoQA.com
- Get Page Title name and Title length
- Print Page Title and Title length on the Eclipse Console.
- Get Page URL and verify if it is a correct page opened
- Close the Browser.

Aim: Perform the following automation using Selenium WebDriver :

- Launch a new Chrome browser.
- Open Shop.DemoQA.com
- Get Page Title name and Title length
- Print Page Title and Title length on the Eclipse Console.
- Get Page URL and verify if it is a correct page opened
- Close the Browser.

Procedure:

```
package automationFramework;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class WebDriverCommands {
public static void main(String[] args) {
    String driverExecutablePath = "D:\\Drivers\\chromedriver.exe";
    System.setProperty("webdriver.chrome.driver", driverExecutablePath);
    // Create a new instance of the FireFox driver
    WebDriver driver = new ChromeDriver();
    // Storing the Application Url in the String variable
    String url = "https://www.shop.demoqa.com";
    //Launch the ToolsQA WebSite
    driver.get(url);
    // Storing Title name in the String variable
    String title = driver.getTitle();
    // Storing Title length in the Int variable
    int titleLength = driver.getTitle().length();
    // Printing Title & Title length in the Console window
    System.out.println("Title of the page is : " + title);
    System.out.println("Length of the title is : " + titleLength);

    // Storing URL in String variable
    String actualUrl = driver.getCurrentUrl();

    if (actualUrl.equals(url)){
        System.out.println("Verification Successful - The correct Url is opened.");
    }
    else {
        System.out.println("Verification Failed - An incorrect Url is opened.");
        //In case of Fail, you like to print the actual and expected URL for the record
        System.out.println("Actual URL is : " + actualUrl);
        System.out.println("Expected URL is : " + url);
    }
    //Closing browser
    driver.close();
}
}
```

EXPERIMENT 15

Q: Perform the following automation using Selenium WebDriver.

- Firstly, open the browser.
- Secondly, navigate to the **ToolsQA Demo Website**.
- Thirdly, maximize the browser window.
- After that, retrieve the title of the page.
- Fifthly, log in to the Website by specifying credentials

Aim: Perform the following automation using Selenium WebDriver.

- Firstly, open the browser.
- Secondly, navigate to the **ToolsQA Demo Website**.
- Thirdly, maximize the browser window.
- After that, retrieve the title of the page.
- Fifthly, log in to the Website by specifying credentials

Procedure:

```
package firstPackage;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

public class MyFirstTestClass {

    public static void main(String[] args) {
        //Setting the driver path
        System.setProperty("webdriver.chrome.driver", "E:\\Softwares\\chromedriver.exe");

        //Creating WebDriver instance
        WebDriver driver = new ChromeDriver();
        //Navigate to web page
        driver.get("https://demoqa.com/login");
        //Maximizing window
        driver.manage().window().maximize();
        //Retrieving web page title
        String title = driver.getTitle();
        System.out.println("The page title is : " +title);

        //Locating web element
        WebElement uName = driver.findElement(By.id("userName"));
        WebElement pswd = driver.findElement(By.id('password'));
        WebElement loginBtn = driver.findElement(By.id("login"));
        //Performing actions on web elements
        uName.sendKeys("testuser");
        pswd.sendKeys("Password@123");
        loginBtn.click();
        //Closing browser session
        driver.quit();

    }
}
```