

MCAL35 Software Testing and Quality Assurance Lab Index

Sr. No.	Topic Name	Date	CO	Sign
Manual Testing				
1.	Implementation of Black Box Testing Prepare test cases for below given applications using Boundary Value Analysis and Equivalence Class Partitioning i. BMI Calculator: Input: Person's Weight and Height Output: Body Mass Index The range of Weight : 3 kg to 300 kg The range of Height : 0.3 meter to 2.4 meter ii. Triangle Problem : Triangle Problem accepts three integers – a, b, c as three sides of the triangle. It returns the type of triangle (Scalene, Isosceles, Equilateral, Not a Triangle) formed by a, b, c.		CO1	
2.	Implementation of White Box Testing Data Flow Analysis, Control Flow Analysis, Cyclomatic Complexity		CO1	
3.	i. Prepare test cases on ATM Machine using Unit and System Testing ii. Prepare a test cases on Login Page of Gmail using Unit and System Testing iii. Prepare a test cases on Calculator using Unit and System Testing		CO1	
Automation Testing				
4.	Introduction to Selenium i. Write down a steps and process of Selenium IDE Installation on any 1 browser.(eg.Firefox) ii. Record and run test cases on demotour website for login page through Mozilla Firefox. iii. Record and run test cases on demotour website for login page through Google Chrome. iv. Record and run test cases on Registration form of any website v. Record and run test cases on any website to check the validations of elements. (eg. IRCTC , MSRTC , etc)		CO2	
5	Implement Web Drivers on Chrome & Firefox Browsers.		CO2	
6	Demonstrate handling multiple frames in selenium		CO2	
7	Implement Browser command and navigation Commands.		CO2	
8	Implement the find element command		CO2	

9	Demonstrate the Locator(id,css selector, path)		CO2	
10	Demonstrate different types of alerts		CO2	
11	Demonstrate : <ul style="list-style-type: none">• Handling Drop Down,• List Boxes• Command Button,• Radio buttons & text boxes.• Waits command in selenium		CO2	
12	Installation of TestNg , running testNg		CO3	
13	TestNg annotations		CO3	
14	Demonstrate Validation testing		CO4	
15	Perform regression testing		CO4	

1. Implementation of White Box Testing

i. Data Flow Analysis

- Statements where variables receive values,
- Statements where these values are used or referenced.
- A variable is defined but not used or referenced,
- A variable is used but never defined,
- A variable is defined twice before it is used
- To find a variable that is used but never defined,
- To find a variable that is defined but never used,
- To find a variable that is defined multiple times before it is used,
- Deallocating a variable before it is used.
- Time consuming and costly process
- Requires knowledge of programming languages

Data Flow Testing is a type of structural testing. It is a method that is used to find the test paths of a program according to the locations of definitions and uses of variables in the program. It has nothing to do with data flow diagrams.

It is concerned with:

If a statement is a loop or if condition then its DEF set is empty and USE set is based on the condition of statement s.

Data Flow Testing uses the control flow graph to find the situations that can interrupt the flow of the program.

Reference or define anomalies in the flow of the data are detected at the time of associations between values and variables. These anomalies are:

Advantages of Data Flow Testing:

Data Flow Testing is used to find the following issues-

Disadvantages of Data Flow Testing

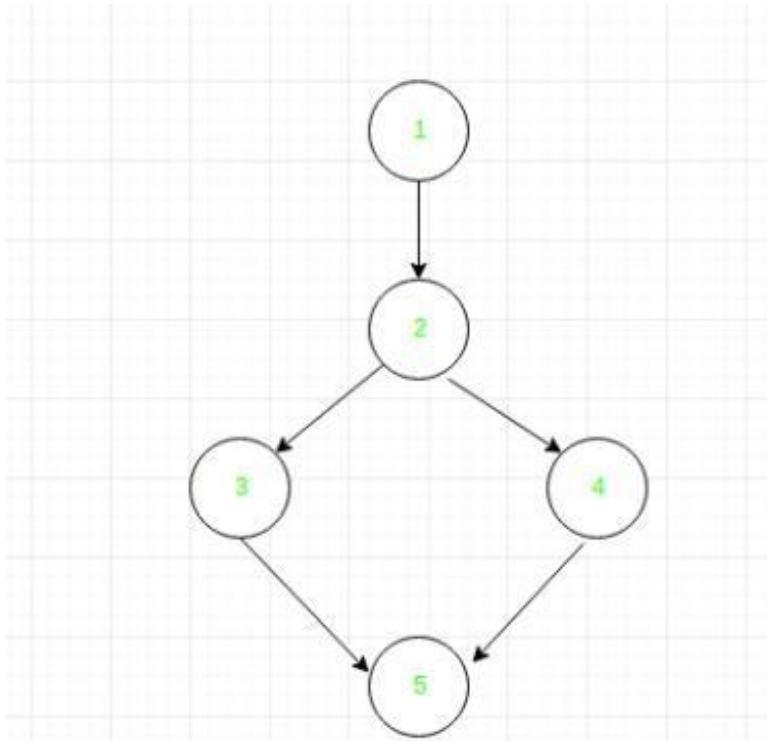
Example:

1. **read x, y;**
2. **if(x>y)**
3. **a = x+1 else**

4. $a = y - 1$

5. print a;

Control flow graph of above example:



Define/use of variables of above example:

Variable	Defined at node	Used at node
x	1	2, 3
y	1	2, 4
a	3, 4	5

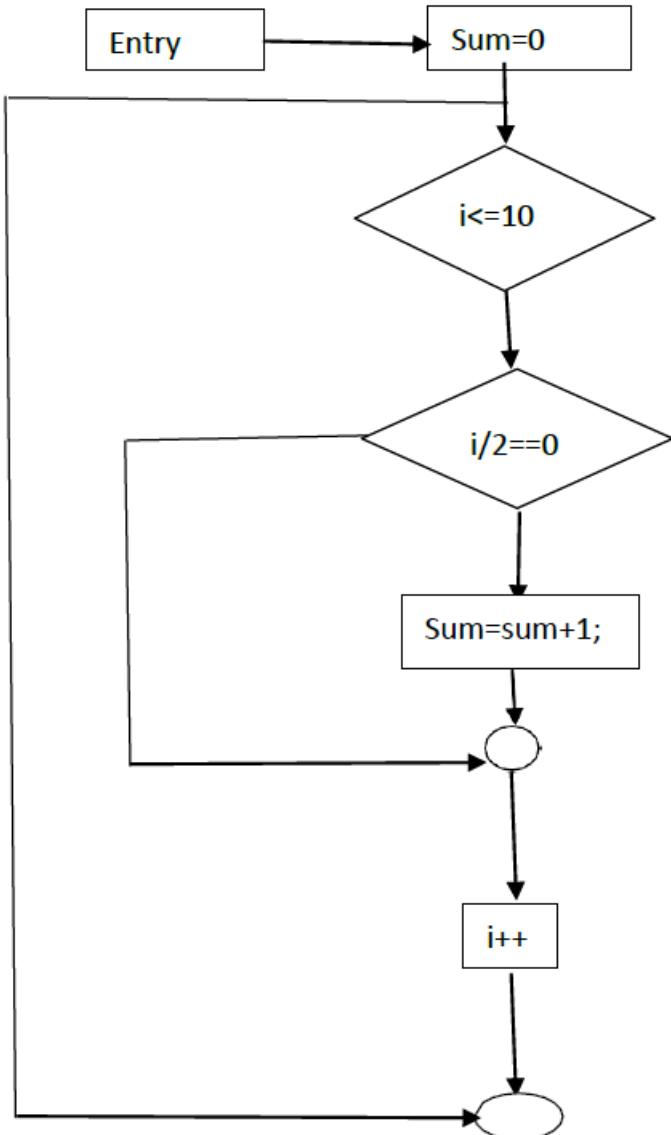
ii. Control Flow Analysis

-Control flow testing uses the control structure of the program to develop the test cases for the program.

- The test cases are developed to sufficiently cover the whole structure of the program.
- The control structure of a program can be represented by the control flow graph of the program.

Control Flow Graph:

```
int evensum(int i)
{
    int sum=0; while (i<=10)
    {
        if(i/2==0) sum=sum+i; i++;
    }
    return sum;
}
```



Statement Converage = (Total number of executed Statement) * 100
Total no of Statement

Cyclomatic Complexity

If A=354

Then If B > C

Then A = B

Else A = C

End If

Print A

Method 1:

Cyclomatic Complexity = Total number of Closed regions in the + 1

$$=2+1$$

$$=3$$

Method 2:

Cyclomatic complexity = E-N+2, E= Edges and N= Nodes

$$=8-7+2$$

$$=3$$

Method 3:

Cyclomatic Complexity = P + 1

$$=2+1$$

$$=3$$

Test cases

1.Prepare test cases on ATM Machin using Unit and SystemTesting

TEST CASE NAME	TEST CASE DESCRIPTION	P	STEP NAME	STEP DESCRIPTION	TEST DATA	EXPECTED RESULT
TC01_Bank_Money withdrawal_verify card insertion with valid cards	This test case to validate card insertion functionality with valid card	P0	Step1	Insert valid card in the insertion point of atm	Valid atm card	Atm should display language page with following objects ENGLISH,TELUGU,HINDI
TC02_Bank_Money withdrawal_verify card insertion with invalid cards	This test case to validate card insertion functionality with invalid card		Step1	Insert invalid card of other bank card in the insertion point of atm	Invalid card like other bank card	Atm should not accept the card and display a message "please insert valid atm card"
			Step2	Insert invalid card of expired atm card in the insertion point of atm	Invalid card like expired atm card	Atm should not accept the card and display a message "Sorry unable to process your request code 1234"
TC03_Bank_Money withdrawal_verify card insertion with valid cards in wrong angle	This test case to validate card insertion functionality with valid card in wrong angle		Step1	Insert invalid card of expired atm card in the insertion point of atm in wrong angle	Valid atm card	Atm should not accept the card and display a message "Sorry unable to process your request code 1222"
TC04_Bank_Money withdrawal_verify language selection	This test case to verify the language selection functionality		Step1	Insert valid card in the insertion point of atm	Valid atm card	Atm should display language page with following objects ENGLISH,TELUGU,HINDI
			Step2	Click on corresponding language to be used		Atm should display the pin number page in corresponding language selected
TC04_Bank_Money withdrawal_verify pin number entry with valid pin number	This test case is to verify the functionality of pin number functionality with valid pin number		Step1	Insert valid card in the insertion point of atm	Valid atm card	Atm should display language page with following objects ENGLISH,TELUGU,HINDI
			Step2	Click on corresponding language to be used		Atm should display the pin number page in corresponding language selected
			Step3	Enter the valid pin number	Valid pin number	Atm should display the account type selection page

2.Prepare a test cases on Login Page of Gmail using Unit and System Testing

Sr.No	Action	Inputs	Expected	Actual	Result
1	EmailTextbox	nash123@gmail.com	Not Accepted	Accepted	Pass
2	EmailTextbox	abc@gmail	Not Accepted	Accepted	Pass
3	EmailTextbox	1st name and Last name not matched from Sign up	Not Accepted	Accepted	Fail
4	EmailTextbox	123@gmail.com	Not Accepted	Accepted	Fail
5	Password Box	Passwd	Not Accepted	Accepted	Pass
6	Password Box	abc123@	Not Accepted	Accepted	Fail
7	Password Box	Abc123@	Accepted	Accepted	Pass
8	Password Box	Abc12@	Not Accepted	Accepted	Pass
9	Button	Email is Null and Only Password entered	Not Accepted	Accepted	Pass
		Email and Password not Matched with Sign up page	Not Accepted	Not Accepted	Pass
10	Button		Not Accepted	Accepted	Pass

3.Prepare a test case on Calculator using Unit and System Testing

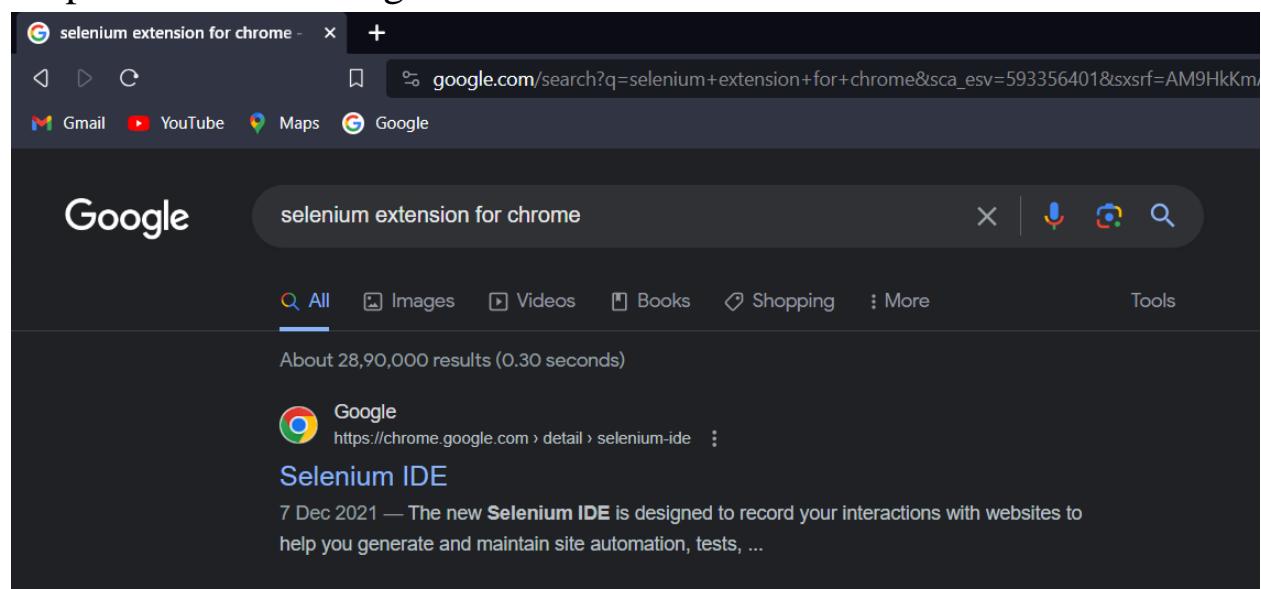
No.	TC-ID	Testcase Objective	Prerequisite	Steps	Input data	Expected Result	Actual Result	Remarks/ Status
1	TC1	To add two integer and display the result on ten digit calculator	Calculator is switched on	1.Key in a valid integer from - 999999999 to +999999999 2.Key in operator + 3.Key in second operand,a valid integer from - 999999999 To +999999999	135 + 100	235(addition, above ten digits will be expressed in exponential form)	235	Pass
2	TC2	To subtract two Integer and display the result on ten digit calculator	Calculator is switched on	1.Key in a valid integer from - 999999999 to +999999999 2.Key in operator - 3.Key in second operand,a valid integer from - 999999999 To +999999999	135 - 100	35(subtraction,above ten digits will be expressed in exponential form)	35	Pass
3	TC3	To multiply two integer and display the result on ten digit calculator	Calculator is switched on	1.Key in a valid integer from - 999999999 to +999999999 2.Key in operator * 3.Key in second operand,a valid integer from - 999999999 To +999999999	100 x 400	40000(multiplication,above ten digits will be expressed in exponential form)	40000	Pass
4	TC4	To divide two integer and display the result on ten digit calculator	Calculator is switched on	1.Key in a valid integer from - 999999999 to +999999999 2.Key in operator / 3.Key in second operand,a valid integer from - 999999999 To +999999999	100 / 25	40(division,above ten digits will be expressed in exponential form)	40	Pass
5	TC5	To clear the screen	Calculator is switched on	Press C		Symbol '0' should appear on screen	Symbol '0' appears on screen	Pass

4.Introduction to Selenium:

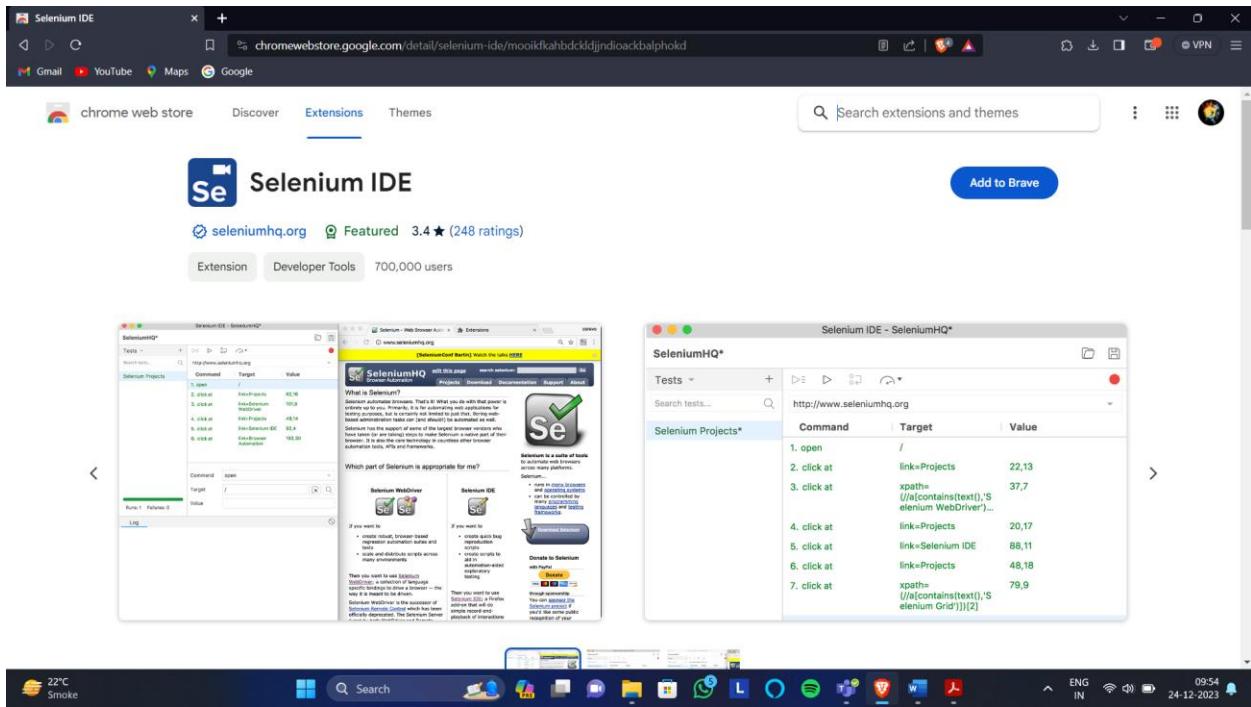
1)Write down a steps and process of Selenium IDE Installationon any 1 browser (eg.Firefox).

Installation in Goolge Chrome:

Step1: Search on Google chrome for “selenium extension for chrome”



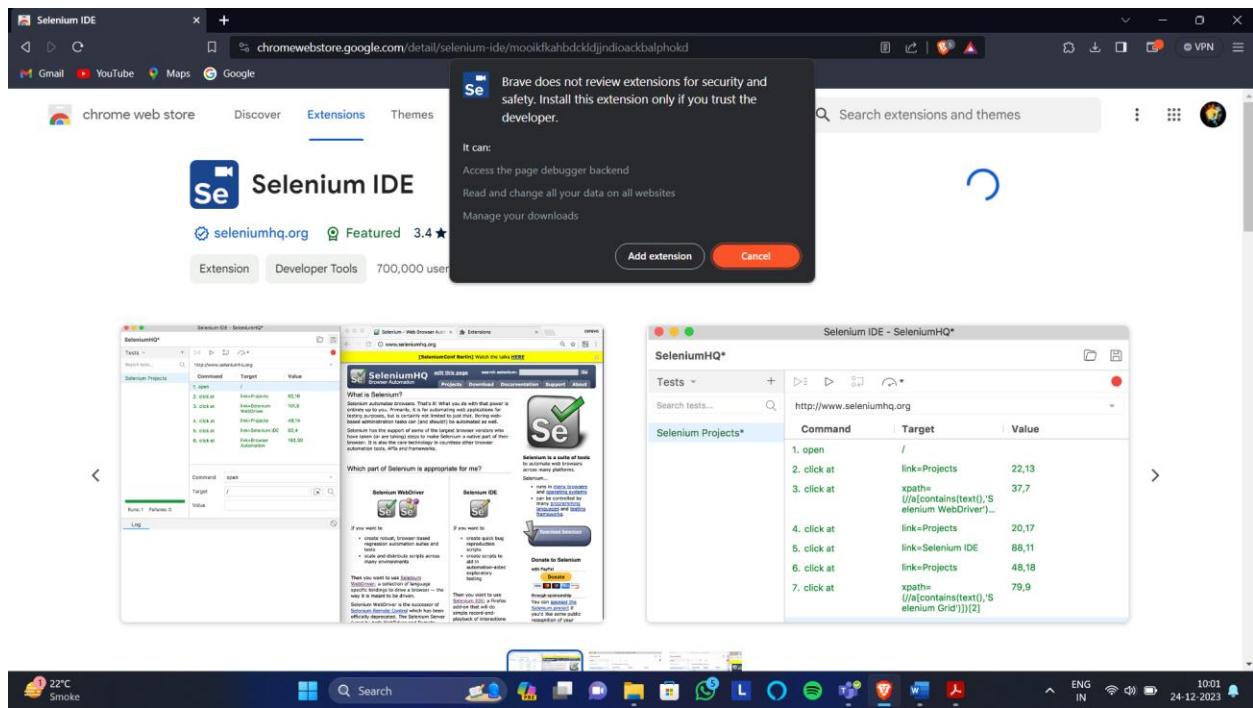
Step 2: click on add to chrome:



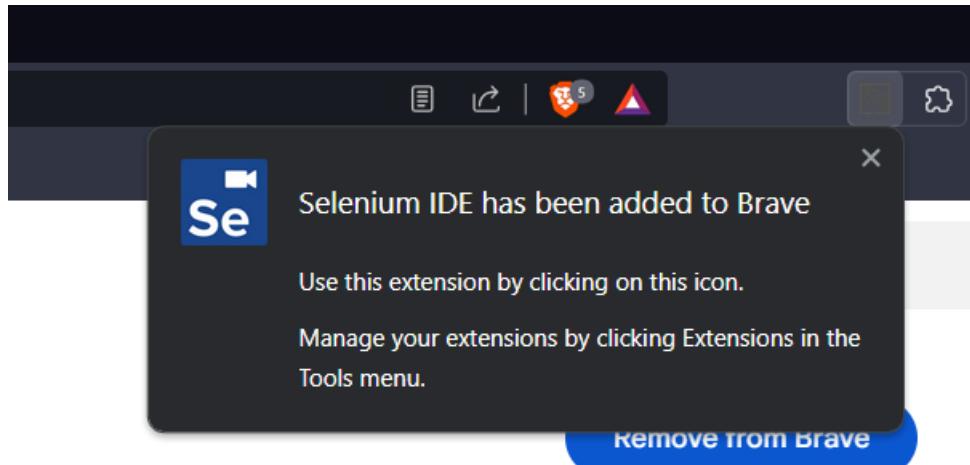
Step : Click on Add to Chrome



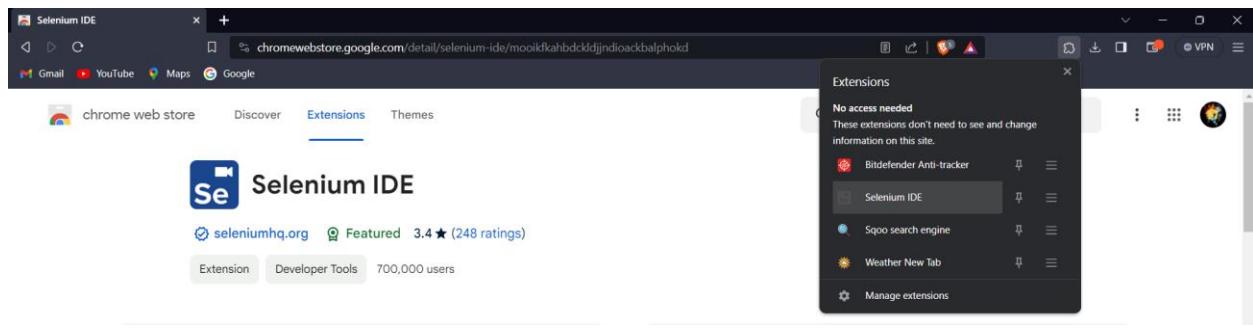
Step 4: Click on Add Extension



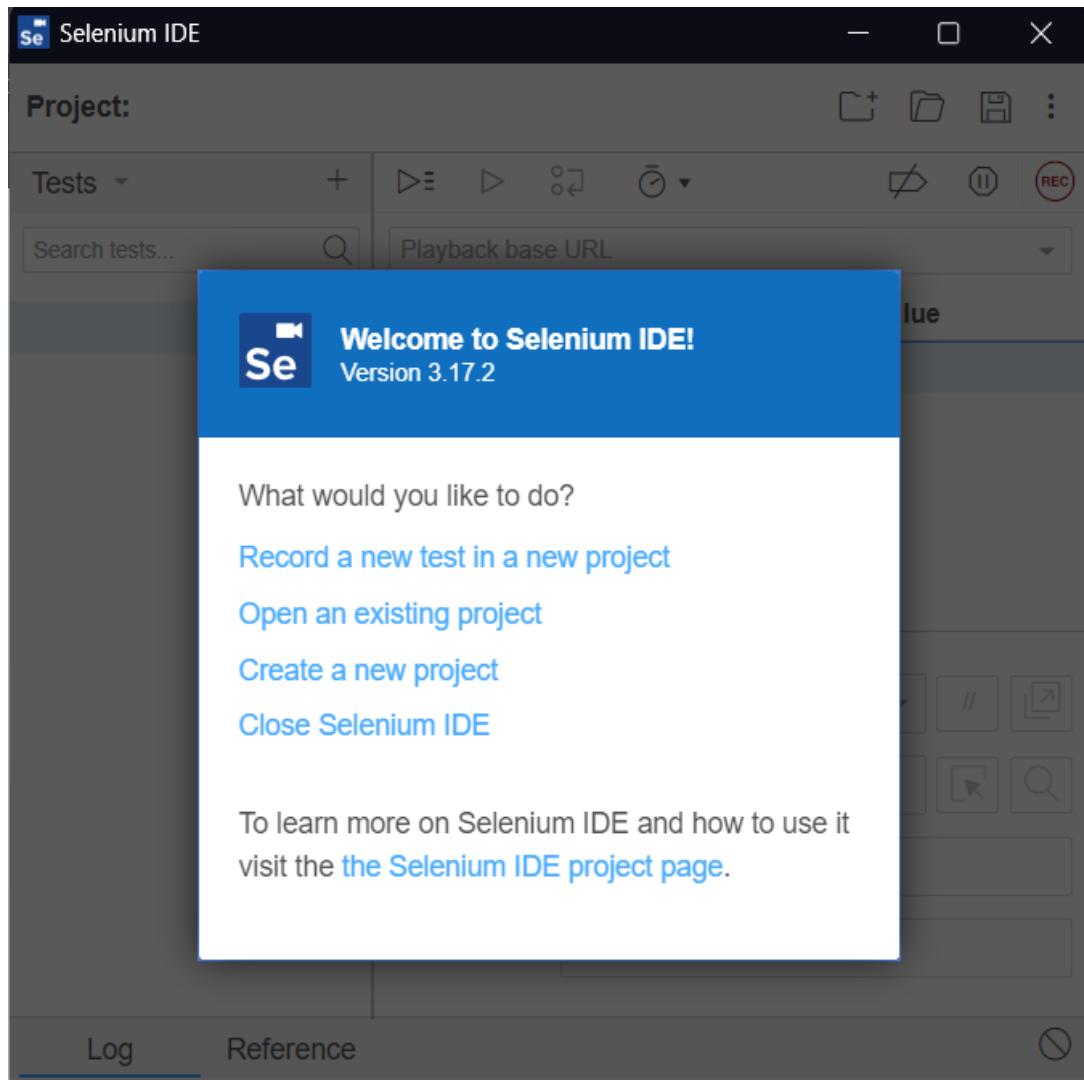
Step 5: Installation Done and the following pop-up will be shown



Step 6: To open Selenium in Chrome, Click on Following



Step 7: Selenium Window will open



2) Record and run test cases on demotour website for login page through Mozilla Firefox.

Case1: Both username and password are incorrect

The screenshot shows the Selenium IDE interface with the following details:

- Project:** Demotour_Moz*
- Test Case:** TestCase1*
- Test Steps:** 7 steps recorded:
 - Step 4: type name=userNa me value=mhvk
 - Step 5: click name=passw ord
 - Step 6: type name=passw ord value=ukhcyud
 - Step 7: click name=submit
- Log:** A list of executed commands with timestamps:
 2. setWindowSize on 1411x753 OK 23:43:24
 3. click on name=userName OK 23:43:24
 4. type on name=userName with value mhvk OK 23:43:26
 5. click on name=password OK 23:43:27
 6. type on name=password with value ukhcyud OK 23:43:27
 7. click on name=submit OK 23:43:27
- Status:** 'TestCase1' completed successfully 23:43:27

Case2: Username is correct but password is incorrect

Extension: (Selenium IDE) - Selenium IDE - Demotour_Moz* — Mozilla Firefox

Project: Demotour_Moz*

Tests + ⏪ ⏴ ⏵ ⏷ ⏸ ⏹ ⏺ REC

Search tests... Run current test Ctrl+R m/test/newtours/

	Command	Target	Value
✓ TestCase1*			
✓ TestCase2*	5 ✓ click	name=passw ord	
TestCase3*	6 ✓ type	name=passw ord	jfyui
	7 ✓ click	name=submit	
	8 ✓ close		

Command Target Value Description

Log Reference

- 3. click on name=userName OK 23:44:41
- 4. type on name=userName with value sameer12 OK 23:44:44
- 5. click on name=password OK 23:44:44
- 6. type on name=password with value jfyui OK 23:44:44
- 7. click on name=submit OK 23:44:44
- 8. close OK 23:44:45

'TestCase2' completed successfully 23:44:45

Case3: Username is incorrect but password is correct

Extension: (Selenium IDE) - Selenium IDE - Demotour_Moz* — Mozilla Firefox

Project: Demotour_Moz*

Tests + Test Case 1* Test Case 2* Test Case 3*

Search tests... RECORD

https://demo.guru99.com/test/newtours/

	Command	Target	Value
7	✓ type	name=userNa me	kvhjl
8	✓ click	name=passw ord	
9	✓ type	name=passw ord	sam
10	✓ click	name=submit	

Command RECORD

Target RECORD

Value

Description

Log Reference RECORD

Log	Reference	Date
5. click on name=password OK		23:45:52
6. click on name=userName OK		23:45:53
7. type on name=userName with value kvhjl OK		23:45:53
8. click on name=password OK		23:45:53
9. type on name=password with value sam OK		23:45:53
10. click on name=submit OK		23:45:53
'TestCase3' completed successfully		23:45:53

Case 4: Both Username and Password are correct

The screenshot shows the Selenium IDE interface with the following details:

- Project:** Demotour_Moz*
- Tests:** TestCase1*, TestCase2*, TestCase3*, TestCase4* (selected)
- Run current test:** Ctrl+R or m/test/newtours/
- Table of Test Steps:**

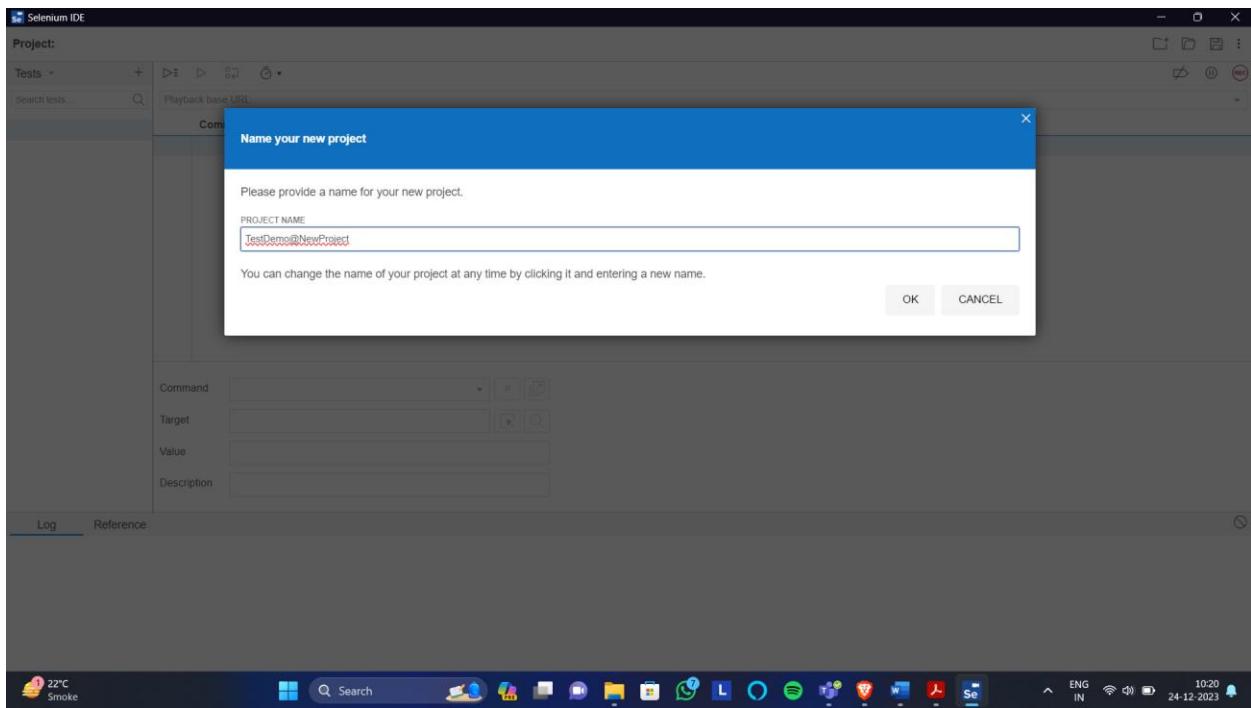
	Command	Target	Value
4	✓ click	name=userNa me	
5	✓ type	name=userNa me	sameer12
6	✓ click	css=div:nth-c hild(5)	
7	✓ click	name=submit	

- Input Fields:** Command, Target, Value, Description.
- Log:** A list of log entries corresponding to the steps in the table.
- Log Entries:**

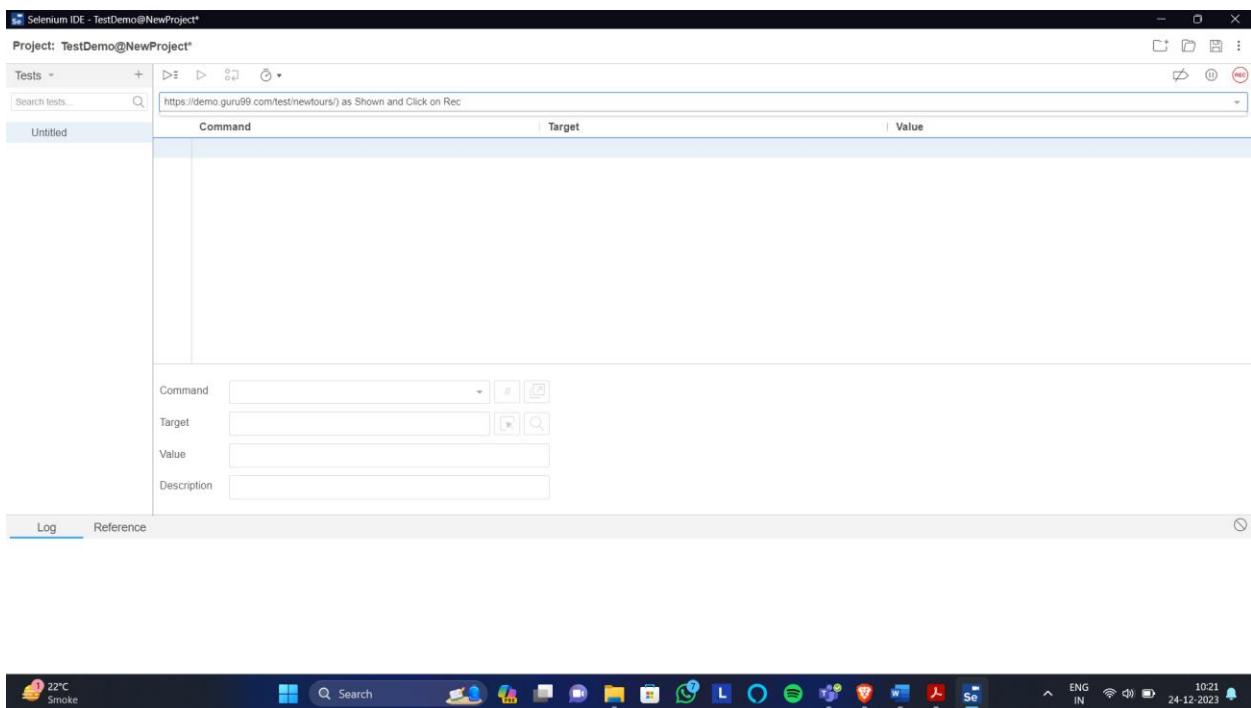
Step	Action	Time
2.	setWindowSize on 1411x753 OK	23:46:46
3.	mouseOver on css=td:nth-child(4) img OK	23:46:46
4.	click on name=userName OK	23:46:48
5.	type on name=userName with value sameer12 OK	23:46:48
6.	click on css=div:nth-child(5) OK	23:46:48
7.	click on name=submit OK	23:47:03
'TestCase4' completed successfully		23:47:04

3) Record and run test cases on demotour website for login page through Google Chrome.

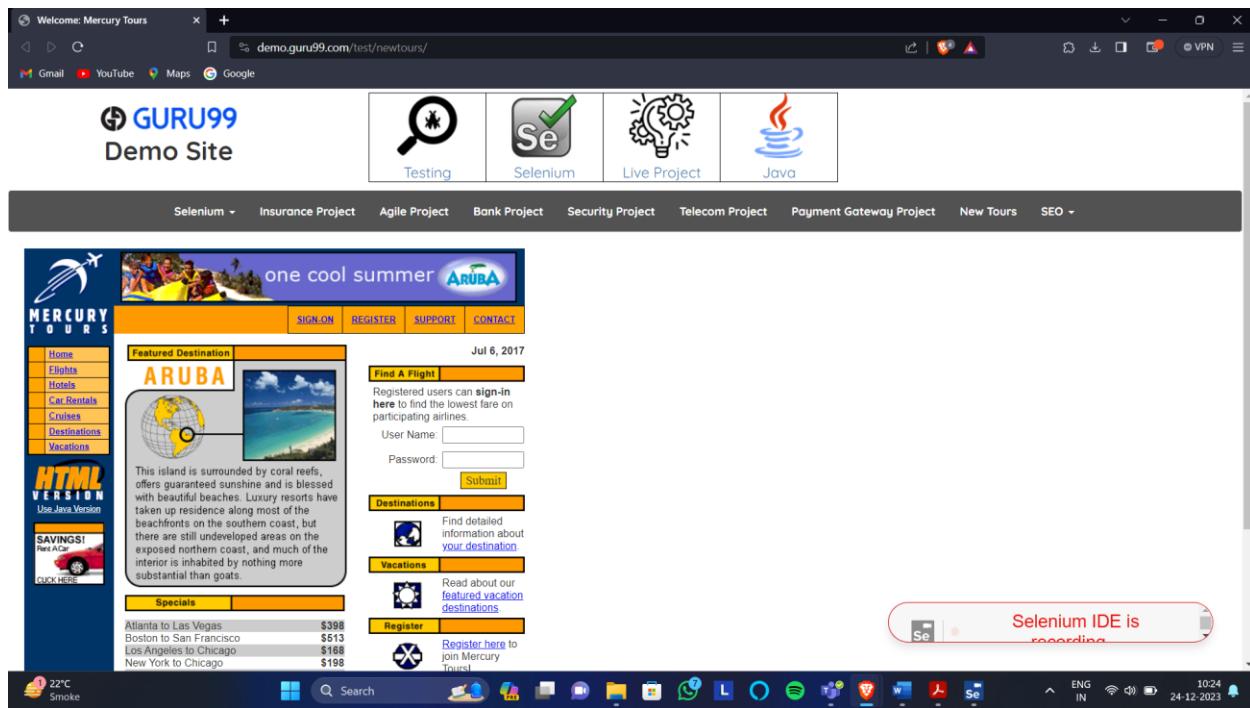
Step 1: Click on new Project and give the project name



Step 2: Paste the demotour website
Link(<https://demo.guru99.com/test/newtours/>) as Shown and
Click on Rec



Step 3: New window will open maximise that window



First Test Case: Both the username and password is incorrect

Command	Target	Value
open	https://demo.guru99.com/test/newtours/	
set window size	1552x832	
click	name=userName	
type	name=userName	Sumit@2001
click	name=password	
type	name=password	123456
click	name=submit	

Second Test Case: Username is correct but Password is incorrect

3	type	name=userName	username
6	type	name=password	Pass@123
7	click	name=submit	
8	click	name=userName	
9	type	name=userName	admin
10	type	name=password	12345
11	click	name=submit	

Third test Case: Username is incorrect but Password is correct

10	click	name=userName	
11	type	name=userName	wrong123
12	type	name=password	pass145
13	click	name=submit	

Fourth Test Case: Both Username and password is Correct

15	type	name=userName	admin
16	type	name=password	admin
17	send keys	name=password	\$(KEY_ENTER)

4)Record and run test cases on Registration form of any website

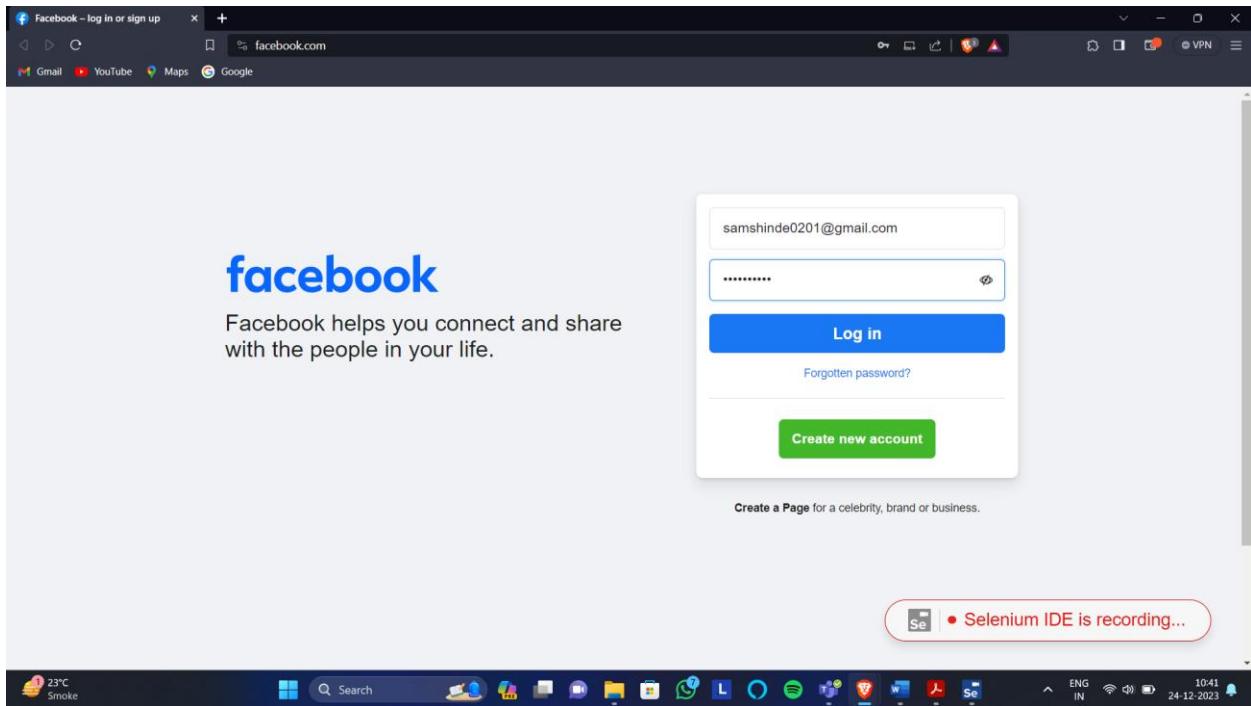
The screenshot shows a web browser window with the title 'Register: Mercury Tours'. The URL in the address bar is <https://demo.guru99.com/test/newtours/register.php>. The page itself is a registration form for 'Mercury TOURS'. It includes a sidebar with links like Home, Flights, Hotels, Car Rentals, Cruises, Destinations, and Vacations, along with options for 'HTML VERSION' and 'Use Java Version'. The main form has three main sections: 'Contact Information' (First Name: hruhikesh, Last Name: Zore, Phone: 9833424548, Email: rushizore99@gmail.com), 'Mailing Information' (Address: 1/18 jogeshwari east, City: mumbai, State/Province: Maharashtra, Postal Code: 400060, Country: INDIA), and 'User Information' (User Name: hruhik999, Password: [redacted], Confirm Password: [redacted]). A 'Submit' button is at the bottom. A red rounded rectangle on the right side contains the text 'Selenium IDE is recording...'. The footer of the page says '© 2005, Mercury Interactive (v. 011003-1.01-058)'.

Command	Target	Value
1 open		https://demo.guru99.com/test/newtours/register.php
2 set window size		1066x812
3 click	name=firstName	
4 type	name=firstName	hruhikesh
5 type	name=lastName	Zore
6 click	name=phone	
7 type	name=phone	9833424548
8 type	id=userName	rushizore99@gmail.com
9 click	name=address1	

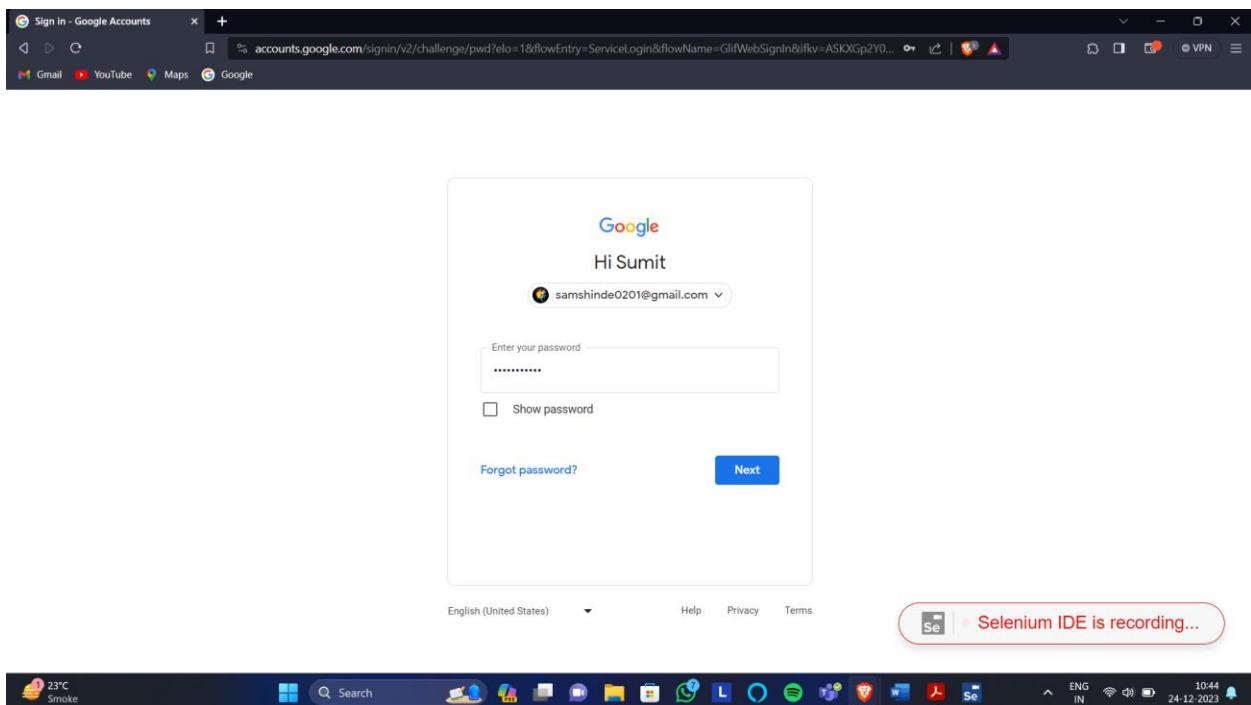
5)Record and run test cases on any website to check the validations of elements. (eg. IRCTC,

MSRTC, etc):

TestCase1: facebook Username and password incorrect



TestCase2: Gmail



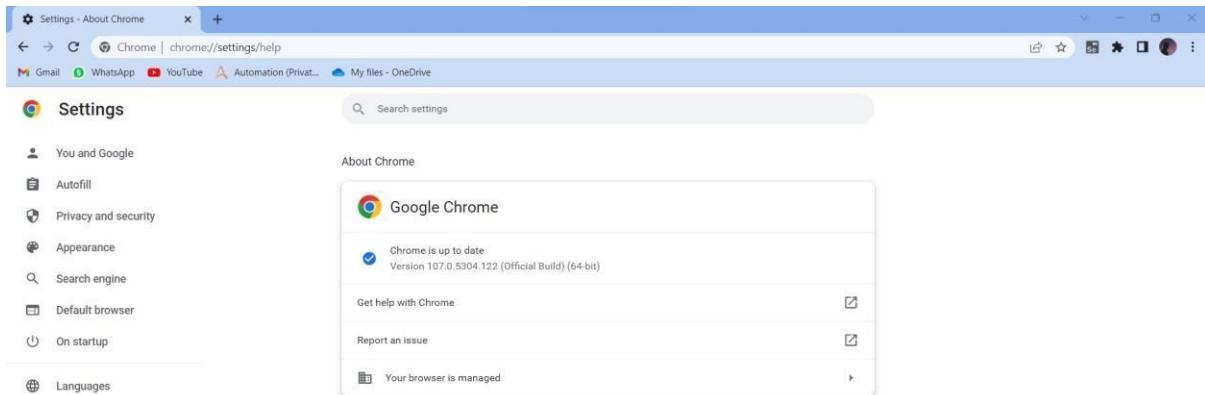
6) Implement Web Drivers on Chrome & Firefox Browsers.

WebDriver in

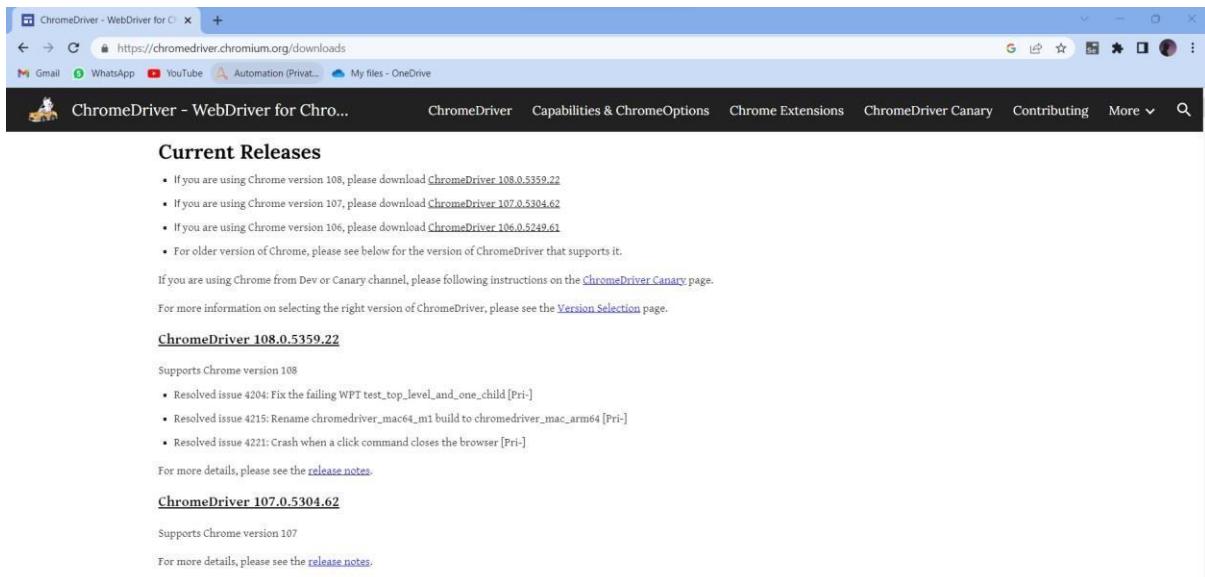
Chrome:

Installation Steps:

Step 1: Check the version of Chrome



Step2: Search for the Driver in google by giving command “selenium driver forchrome”



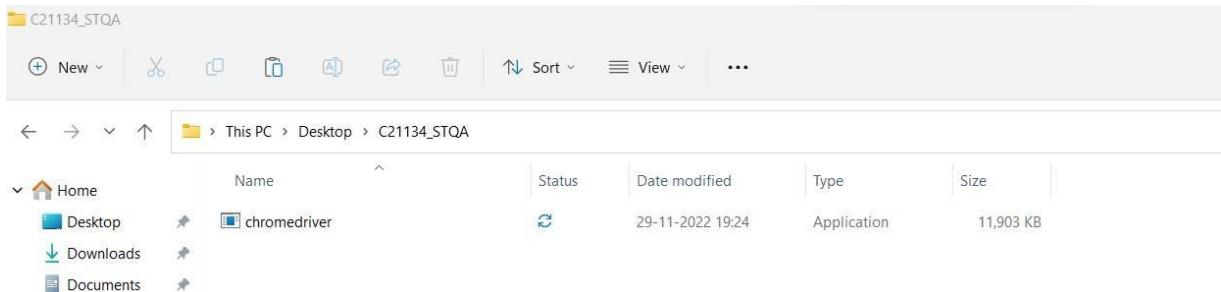
Step3: Click on Respective version and Download the zip file and Create the Separate folder of the name driver and extract the zip file in that folder

The screenshot shows a web browser window with the address bar displaying `https://chromedriver.storage.googleapis.com/index.html?path=107.0.5304.62/`. Below the address bar, there are several browser tabs and links. The main content of the page is titled "Index of /107.0.5304.62/" and lists the following files:

Name	Last modified	Size	ETag
Parent Directory		-	
chromedriver_linux64.zip	2022-10-25 12:20:56	7.26MB	90d3353f17fcbd755626d528e94a1d9a
chromedriver_mac64.zip	2022-10-25 12:20:57	8.41MB	652c969a3b8d47e7fa9518d90b411fba
chromedriver_mac_arm64.zip	2022-10-25 12:20:59	7.72MB	dba9920d41a8ec9fb847326ae0f68200
chromedriver_win32.zip	2022-10-25 12:21:00	6.46MB	a5040d2731fe174c9a7b026edb3fe271
notes.txt	2022-10-25 12:21:05	0.00MB	936b74dab32b11addaffb0a624d9894a

Index of /107.0.5304.62/

Name	Last modified	Size	ETag
Parent Directory		-	
chromedriver_linux64.zip	2022-10-25 12:20:56	7.26MB	90d3353f17fcbd755626d528e94a1d9a
chromedriver_mac64.zip	2022-10-25 12:20:57	8.41MB	652c969a3b8d47e7fa9518d90b411fba
chromedriver_mac_arm64.zip	2022-10-25 12:20:59	7.72MB	dba9920d41a8ec9fb847326ae0f68200
chromedriver_win32.zip	2022-10-25 12:21:00	6.46MB	a5040d2731fe174c9a7b026edb3fe271
notes.txt	2022-10-25 12:21:05	0.00MB	936b74dab32b11addaffb0a624d9894a



Step 4: Download the jar file by typing the following query “selenium jar files download”, Zip file will be downloaded extract the zip file in the Separate dedicated folder.

The screenshot shows the Selenium website with links to language bindings for various programming languages. Each language has a logo, a stable version link (4.6.0 November 4, 2022), a Changelog link, and an API Docs link.

Language	Stable Version	Changelog	API Docs
C#	4.6.0 (November 4, 2022)	Changelog	API Docs
Ruby	4.6.0 (November 4, 2022)	Changelog	API Docs
Java	4.6.0 (November 4, 2022)	Changelog	API Docs
Python	4.6.0 (November 4, 2022)	Changelog	API Docs
JavaScript	4.6.0 (November 4, 2022)	Changelog	API Docs

The screenshot shows a Windows File Explorer window displaying the contents of a folder named "selenium jar files" located on the desktop. The folder contains a subfolder "lib" and several executable JAR files. The table below provides a detailed view of the file list.

Name	Status	Date modified	Type	Size
CHANGELOG	29-11-2022 19:27	File	145 KB	
LICENSE	29-11-2022 19:27	File	12 KB	
NOTICE	29-11-2022 19:27	File	1 KB	
selenium-api-4.6.0	29-11-2022 19:27	Executable Jar File	216 KB	
selenium-api-4.6.0-sources	29-11-2022 19:27	Executable Jar File	169 KB	
selenium-chrome-driver-4.6.0	29-11-2022 19:27	Executable Jar File	15 KB	
selenium-chrome-driver-4.6.0-sources	29-11-2022 19:27	Executable Jar File	11 KB	
selenium-chromium-driver-4.6.0	29-11-2022 19:27	Executable Jar File	29 KB	
selenium-chromium-driver-4.6.0-sources	29-11-2022 19:27	Executable Jar File	21 KB	
selenium-devtools-v85-4.6.0	29-11-2022 19:27	Executable Jar File	1,111 KB	
selenium-devtools-v85-4.6.0-sources	29-11-2022 19:27	Executable Jar File	467 KB	

Step 5: Install Eclipse if not there

Step 6: Open Eclipse and create a java Project, give the project name, Create a package in that Project Give a package name, Create a Class give a name to the class

Screenshot of Eclipse IDE showing the creation of a Java package and class.

New Java Package:

- Java Package: C21134_Selenium_Test_Cases/src
- Name: Demo_Test
- Source folder: C21134_Selenium_Test_Cases/src

New Java Class:

- Java Class: Demo_Test
- Source folder: C21134_Selenium_Test_Cases/src
- Package: Demo_Test
- Name: Demo_Test
- Modifiers:
 - public (selected)
 - protected
 - private
 - final
 - abstract
 - static
 - none (selected)
 - sealed
 - non-sealed
 - final
- Superclass: java.lang.Object
- Interfaces: (empty)
- Method stubs:
 - public static void main(String[] args) (selected)
 - Constructors from superclass
 - Inherited abstract methods (selected)
- Comments: Generate comments (unchecked)

Buttons at the bottom: ? (Help), Finish, Cancel.

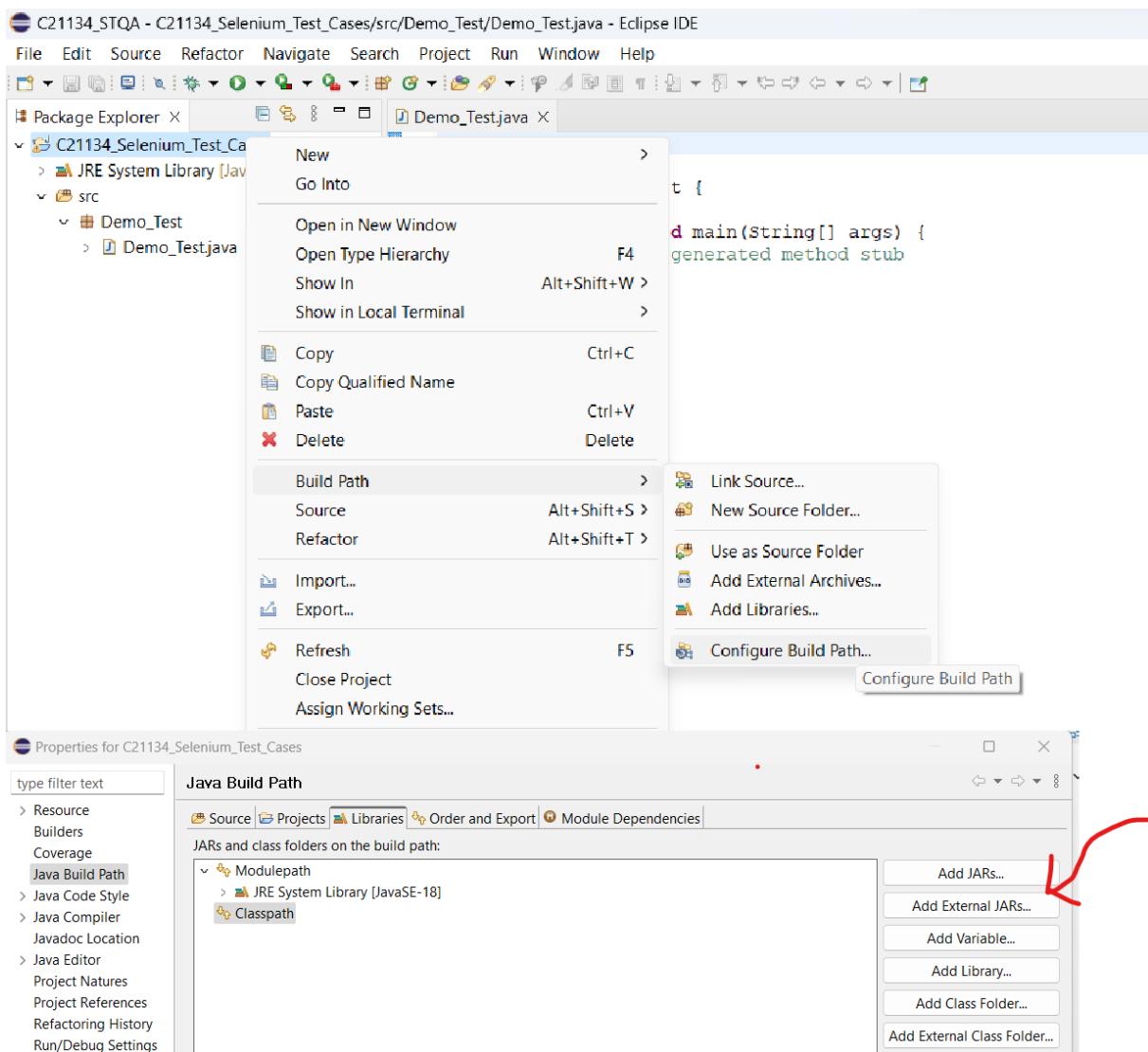
The screenshot shows the Eclipse IDE interface. The top menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The left sidebar is the Package Explorer, showing a project named 'C21134.Selenium_Test_Cases' with a 'src' folder containing a 'Demo_Test' package and a file 'Demo_Test.java'. The right side is the code editor for 'Demo_Test.java', displaying the following code:

```

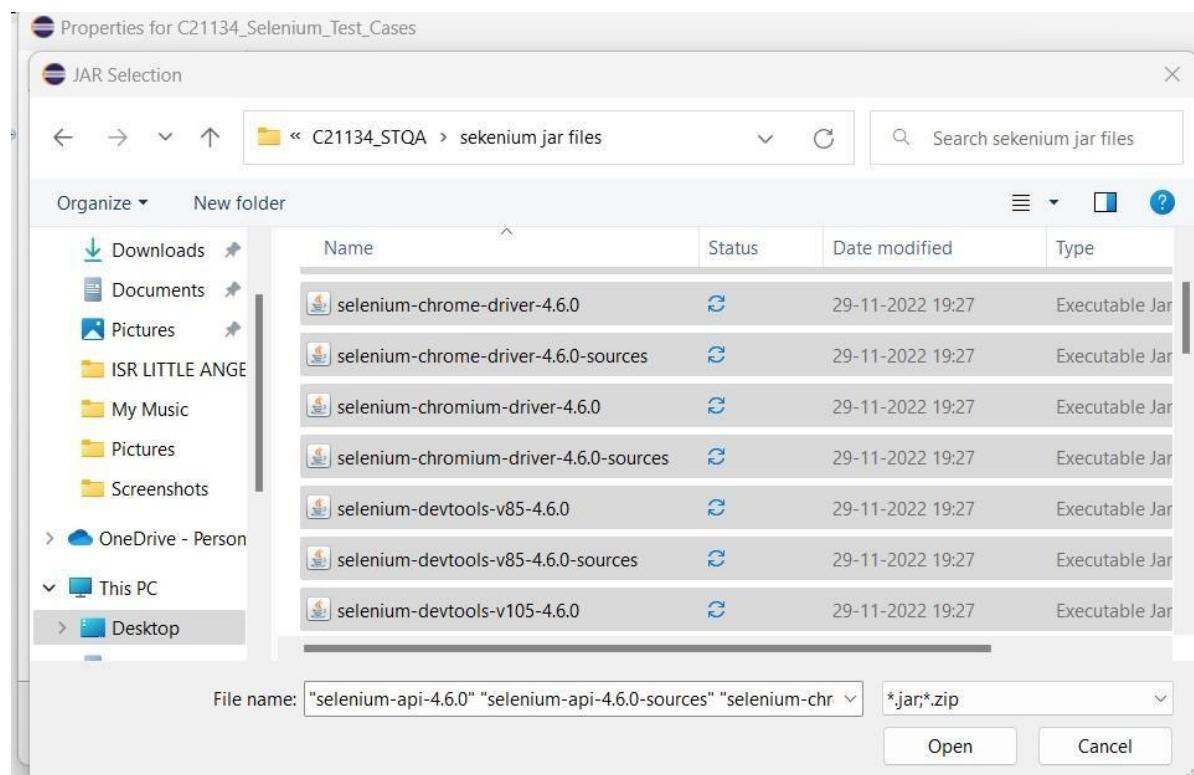
1 package Demo_Test;
2
3 public class Demo_Test {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7
8     }
9
10}
11

```

Step7: Import the jar file of selenium in eclipse, Right click on project goto properties, Goto java build path, Goto Libraries, Click on Classpath , Click on add External jar



Step 8: Select all the jar files inside of lib folder too



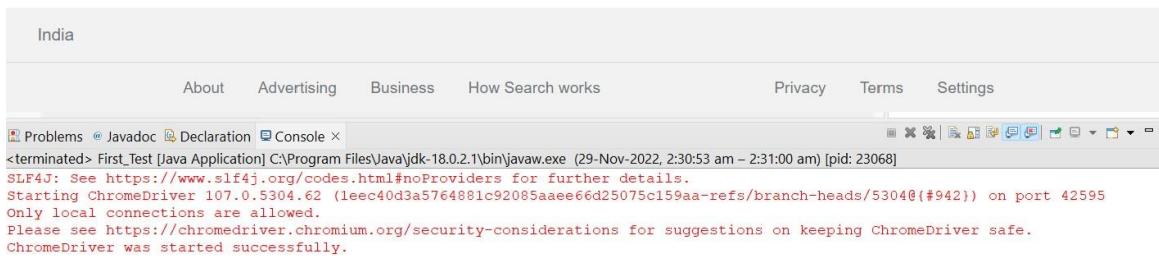
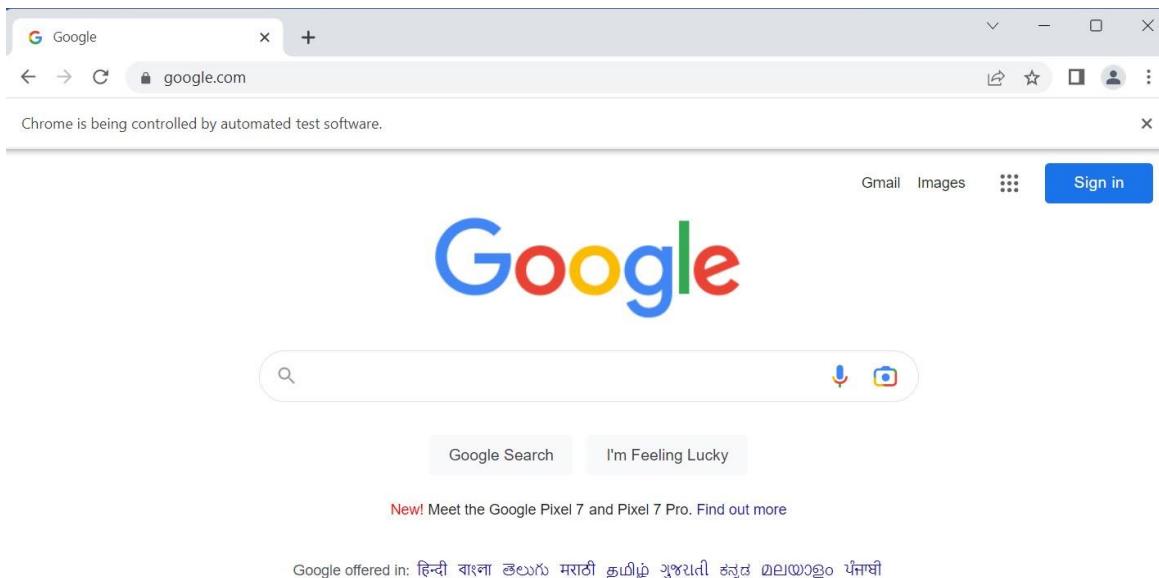
Step 9: Setup is ready know check the basic program for driverCode:

```
package TestCases;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.chrome.ChromeDriver;  
public class First_Test  
{  
    public static void main(String[] args)  
    {  
        System.setProperty("Webdriver.chrome.driver","C:\\\\Users\\\\rushi\\\\OneDrive\\\\Desktop\\\\C21134_STQA\\\\chromedriver.exe")  
    }  
}
```

}

}

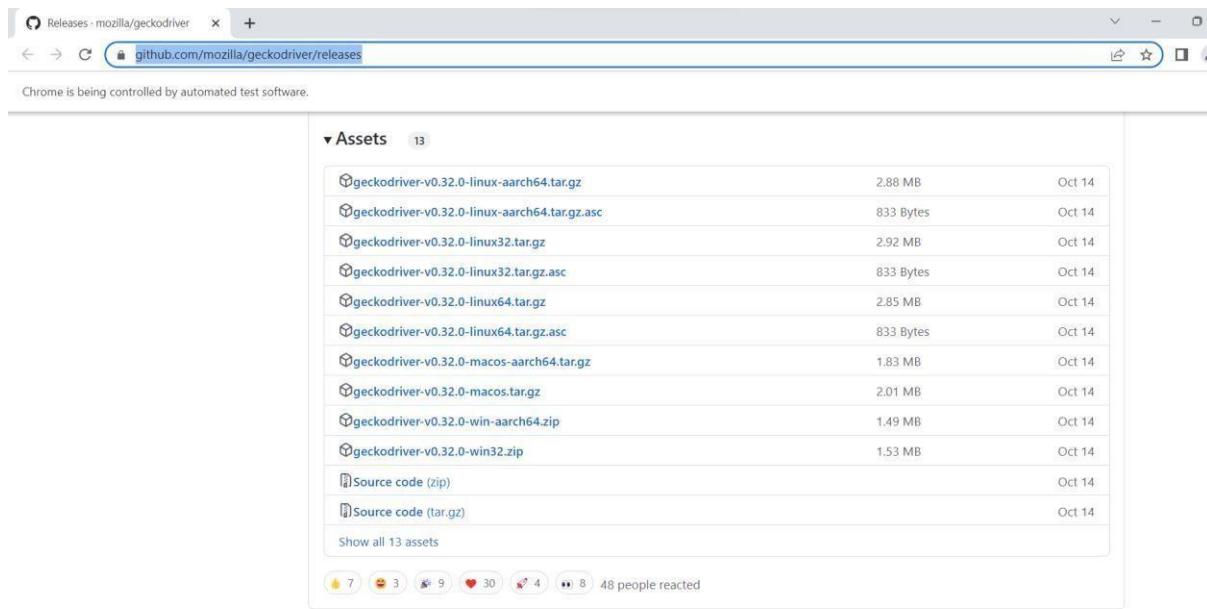
Output :



WebDriver in Firefox:

Step 1. the zip file of diver click on link

<https://github.com/mozilla/geckodriver/releases> and download it and Extract the zip file and save it to the Divers folder



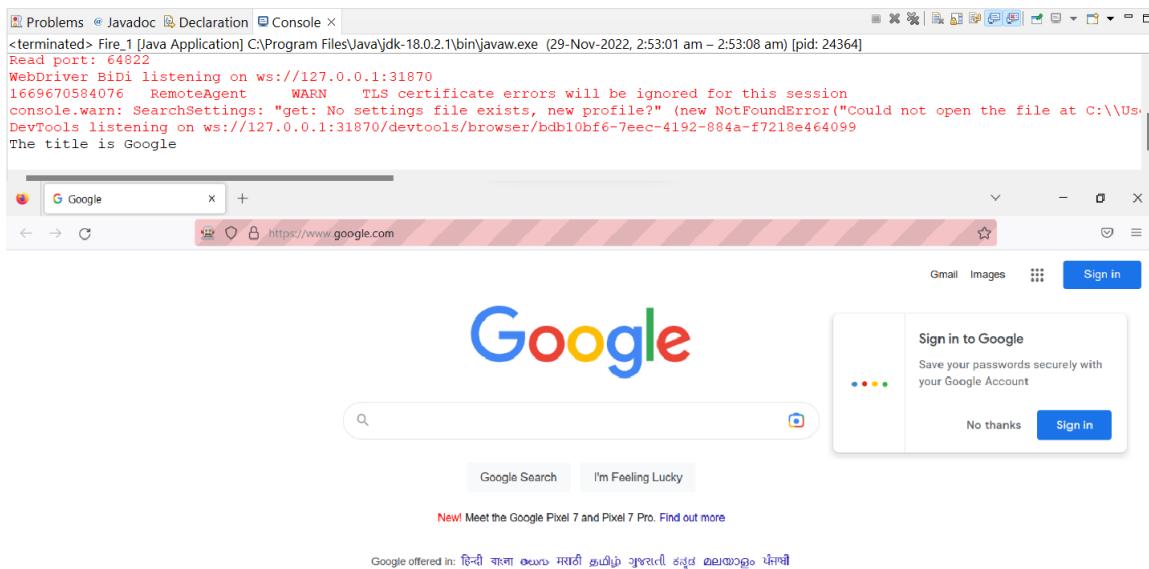
Repeat the same steps from step 4 to step 8

Step 9: Setup is ready know check the basic program for driver package TestCases;

```
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.firefox.FirefoxDriver;  
public class Fire_1  
{  
    public static void main(String[] args)  
    {  
        System.setProperty("Webdriver.gecko.driver",  
"C:\\STQA\\Drivers\\geckodriver.exe");  
        WebDriver driver = new FirefoxDriver();  
        driver.get("https://www.google.com/"); String title =  
        driver.getTitle(); System.out.println("The title is "+title);  
    }  
}
```

{

Output :



Demonstrate handling multiple frames in selenium

Code:

```
package TestCases;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class Multi_Frames
{
    public static void main(String[] args)
    {
        System.setProperty("Webdriver.chrome.driver","C:\\Users\\rushi\\OneDrive\\Desktop\\C21134_STQA\\chromedriver.exe");
```

```

WebDriver driver = new ChromeDriver();
driver.get("http://demo.guru99.com/test/guru99home/");

// navigates to the page consisting an iframe

driver.manage().window().maximize();
driver.switchTo().frame("a077aa5e");

System.out.println("We are switch to the iframe");
driver.findElement(By.xpath("html/body/a/img")).click();

//Clicks the iframe

System.out.println("*We are done*");

}
}

```

Output:

Implement Browser command and navigation Commands. Code:

```
package TestCases;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Navigation {

    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver","C:\\Users\\rushi\\OneDrive\\Desktop\\C21134_STQA
        \\chromedriver.exe");

        WebDriver driver = new ChromeDriver();

        //driver.get("https://www.google.com");
        driver.navigate().to("https://www.google.com");
        driver.findElement(By.linkText("Images")).click(); try {
            Thread.sleep(2000);
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        driver.navigate().back();

        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
```

```
System.out.println("back done    ");

driver.navigate().forward();

try {

Thread.sleep(1000);

} catch (InterruptedException e) {

// TODO Auto-generated catch block e.printStackTrace();

}

System.out.println("Forward done.      ");

driver.navigate().refresh();

try {

Thread.sleep(1000);

} catch (InterruptedException e) {

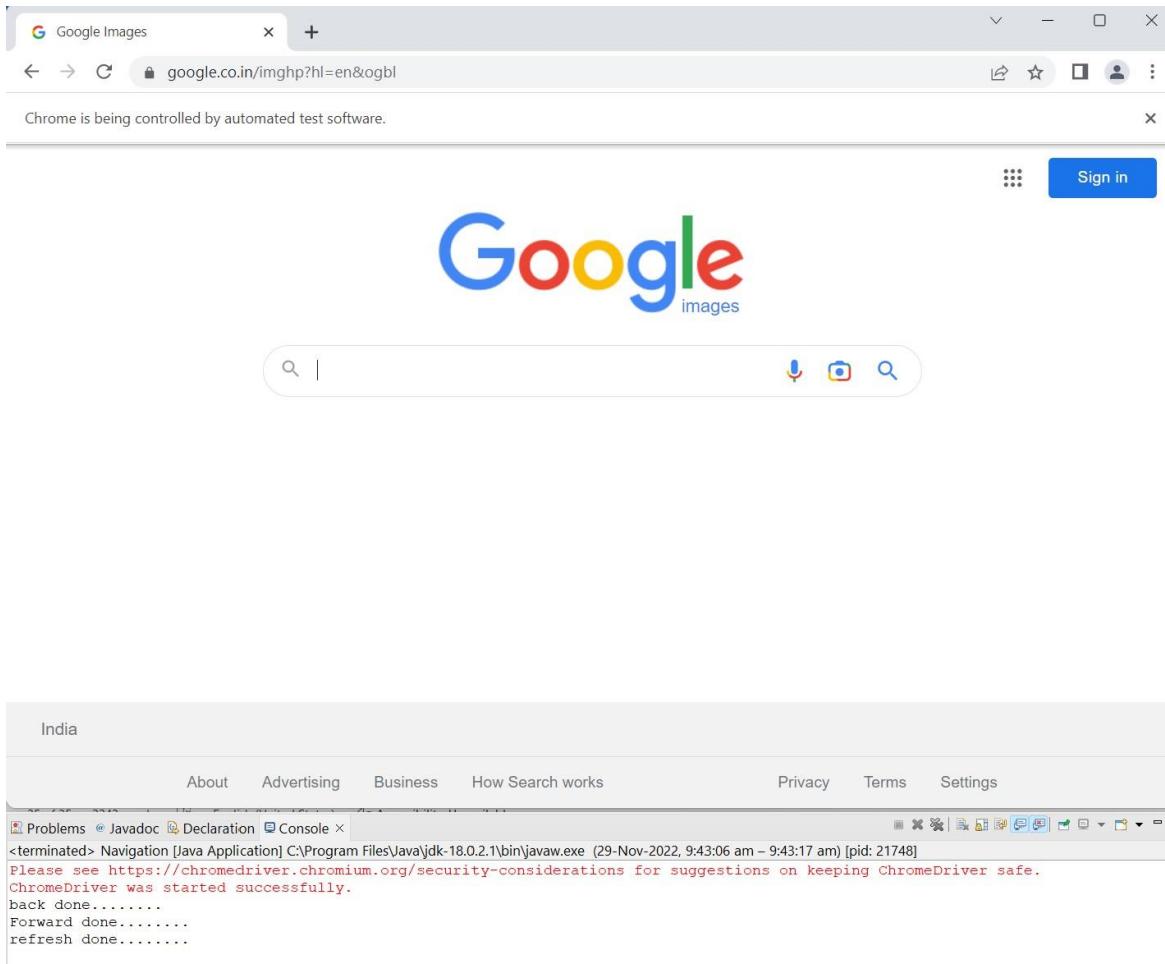
// TODO Auto-generated catch block e.printStackTrace();

}

System.out.println("refresh done. ");

}
```

Output:



Implement the find element command

Code:

```
package TestCases;  
  
import org.openqa.selenium.By;  
  
import org.openqa.selenium.WebDriver;  
  
import org.openqa.selenium.WebElement;  
  
import org.openqa.selenium.chrome.ChromeDriver;  
  
public class Find_Element {
```

```
public static void main(String[] args) {  
    System.setProperty("webdriver.chrome.driver","C:\\Users\\rushi\\On  
eDrive\\Desktop\\C21134_STQA  
\\chromedriver.exe");  
  
    WebDriver driver = new ChromeDriver();  
  
    //driver.get("https://www.google.com");  
    driver.navigate().to("https://www.facebook.com/"); WebElement el =  
    driver.findElement(By.id("email"));  
    el.sendKeys("hanchatesameer@gmail.com"); WebElement el2 =  
    driver.findElement(By.id("pass")); el2.sendKeys("123456");  
  
    WebElement el3 = driver.findElement(By.name("login")); el3.click();  
  
    System.out.println("Login button clicked");  
}  
}
```

Output:

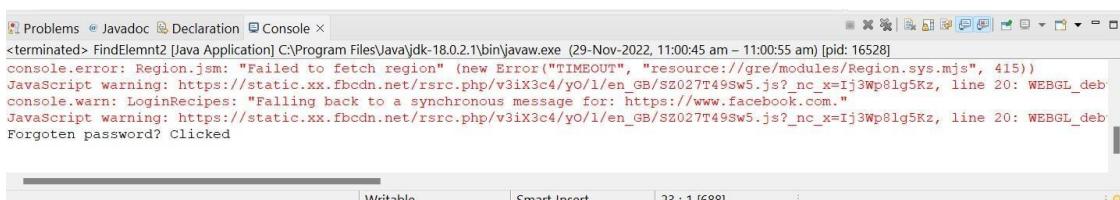
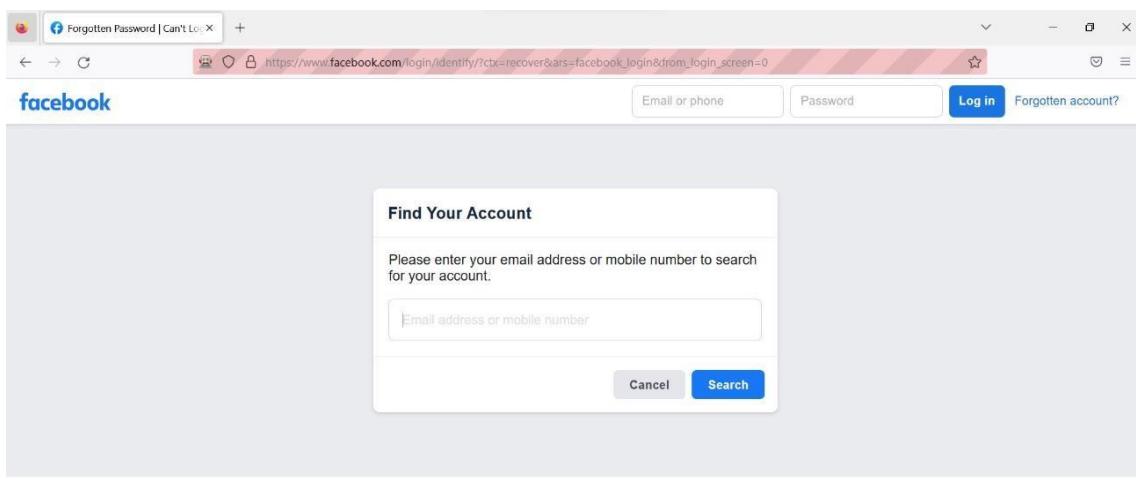


The screenshot shows a Java application window with the title 'Find_Element [Java Application]'. The console tab is active, displaying the following text:
<terminated> Find_Element [Java Application] C:\Program Files\Java\jdk-18.0.2\bin\javaw.exe (29-Nov-2022, 9:56:54 am – 9:56:58 am) [pid: 16612]
Starting ChromeDriver 107.0.5304.62 (1eec40d3a5764881c92085aaee66d25075c159aa-refs/branch-heads/5304@{#942}) on port 47712
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Login button clicked

Forgotten password? Link button testing By Find Element: Code:

```
package TestCases;  
import org.openqa.selenium.By;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.WebElement;  
import org.openqa.selenium.chrome.ChromeDriver;  
import org.openqa.selenium.firefox.FirefoxDriver;  
public class FindElemnt2 {  
    public static void main(String[] args) {  
        System.setProperty("Webdriver.chrome.driver","C:\\Users\\rushi\\On  
eDrive\\Desktop\\C21134_STQA  
\\chromedriver.exe");  
        WebDriver driver = new FirefoxDriver();  
        //driver.get("https://www.google.com");  
        driver.navigate().to("https://www.facebook.com/");  
        WebElement el = driver.findElement(By.linkText("Forgotten  
password?")); el.click();  
        System.out.println("Forgoten password? Clicked");  
    }  
}
```

Output:



To Get the URL and Title:

Code:

```
package TestCases;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class URL_Title {

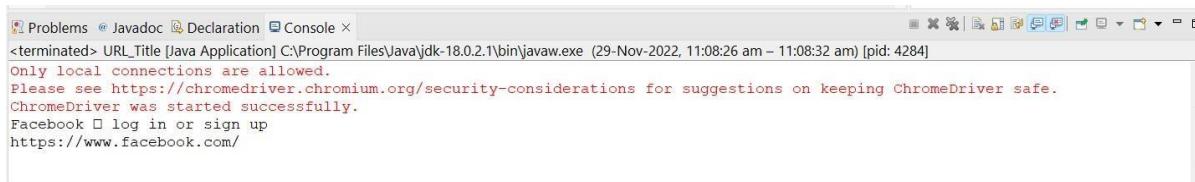
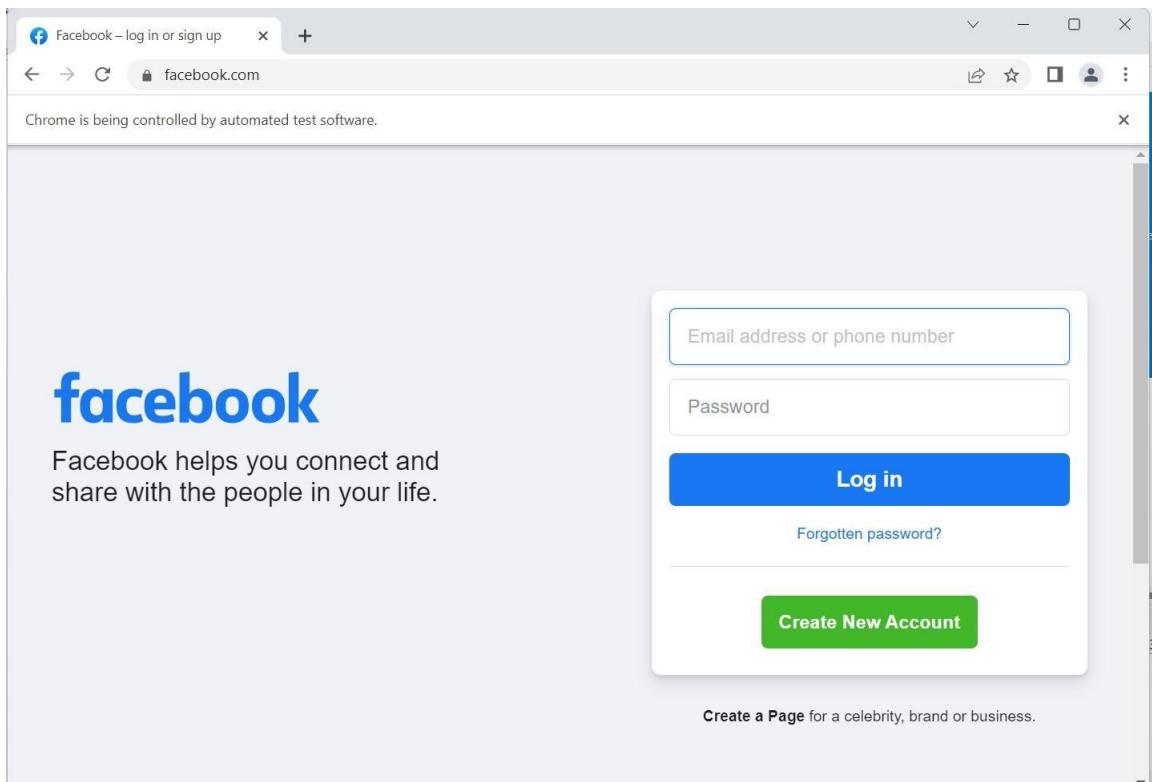
    public static void main(String[] args) {
        System.setProperty("Webdriver.chrome.driver", "C:\\Users\\rushi\\OneDrive\\Desktop\\C21134_STQA
\\chromedriver.exe");

        WebDriver driver = new ChromeDriver();
```

```
//driver.get("https://www.google.com");
driver.navigate().to("https://www.facebook.com/");
System.out.println(driver.getTitle());
System.out.println(driver.getCurrentUrl());

}
}
```

Output:



To Test Attributes and Tag name:

Code:

```
package TestCases;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Tag_Name {

    public static void main(String[] args) {
        System.setProperty("Webdriver.chrome.driver","C:\\Users\\rushi\\OneDrive\\Desktop\\C21134_STQA
        \\chromedriver.exe");

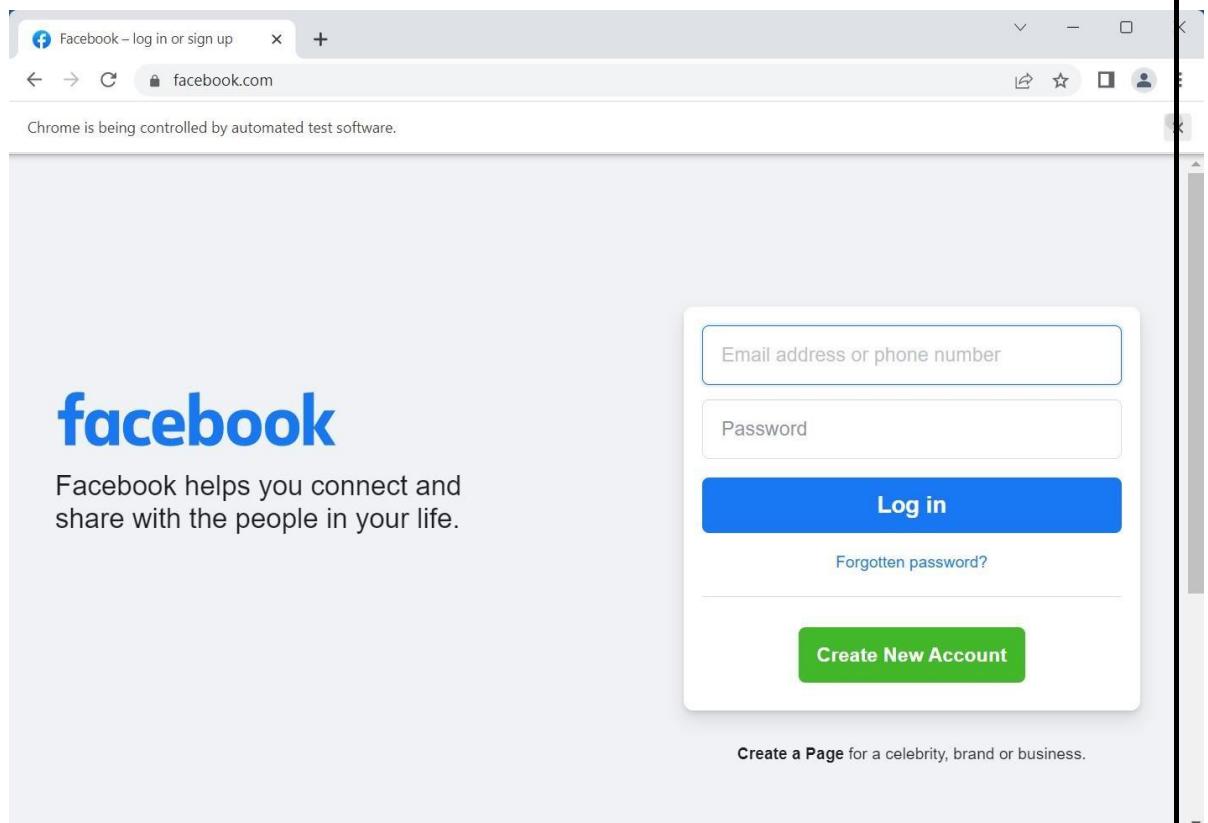
        WebDriver driver = new ChromeDriver();

        //driver.get("https://www.google.com");
        driver.navigate().to("https://www.facebook.com/"); String title =
        driver.getTitle(); System.out.println("Title is : "+title);

        WebElement el = driver.findElement(By.id("email"));
        System.out.println(el.getAttribute("class"));
        System.out.println(el.getTagName());

    }
}
```

Output:



```
<terminated> Tag_Name [Java Application] C:\Program Files\Java\jdk-18.0.2.1\bin\javaw.exe (29-Nov-2022, 11:16:10 am – 11:16:19 am) [pid: 24692]
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Title is : Facebook log in or sign up
inputtext _55rl _6luy
input
```

Demonstrate the Locator (id, CSS selector, path)

XPath Locator:

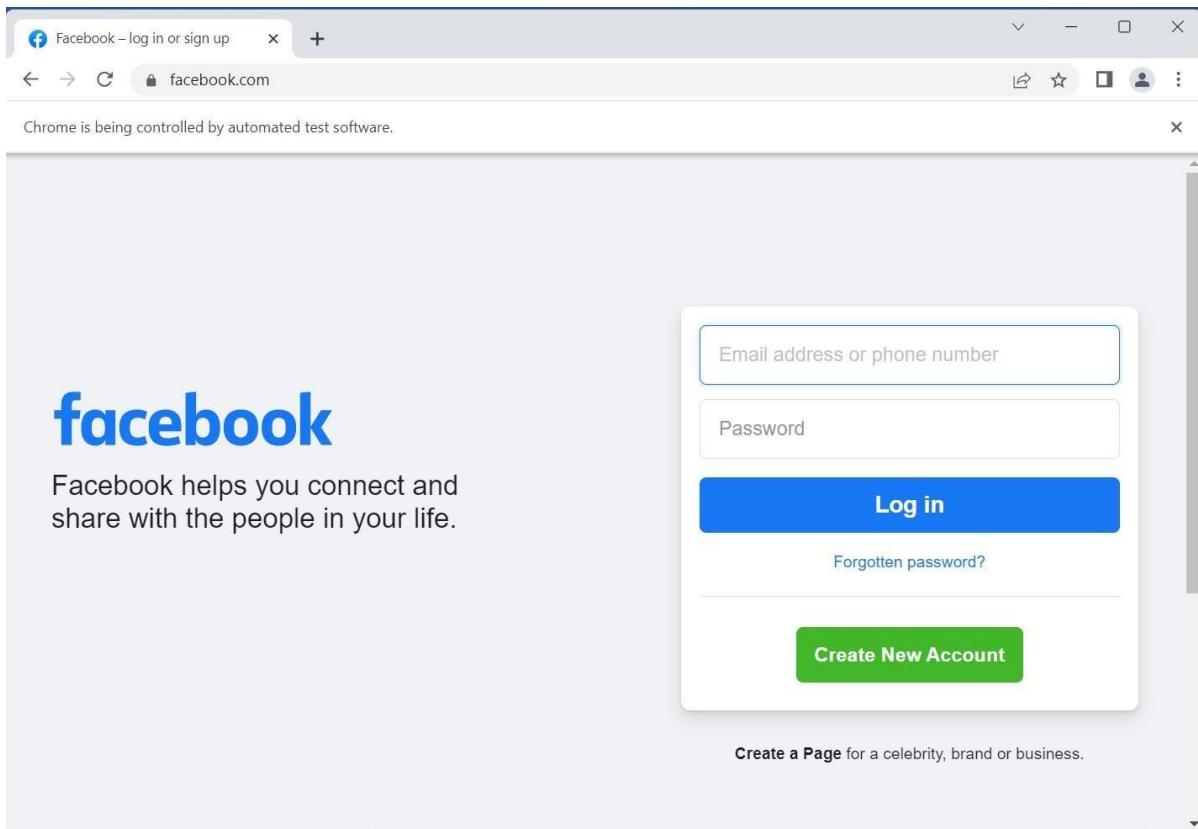
Code:

```
package TestCases;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
```

```
public class Locators {  
    public static void main(String[] args) {  
        System.setProperty("Webdriver.chrome.driver","C:\\Users\\rushi\\On  
eDrive\\Desktop\\C21134_STQA  
\\chromedriver.exe");  
  
        WebDriver driver = new ChromeDriver();  
  
        driver.get("https://www.facebook.com/");  
  
        WebElement img =  
        driver.findElement(By.xpath("//*[@id='email']]"));  
        System.out.println(img.isDisplayed());  
        System.out.println(img.isEnabled());  
        System.out.println(img.isSelected());  
  
    }  
}  
  
System.setProperty("Webdriver.chrome.driver","C:\\Users\\rushi\\On  
eDrive\\Desktop\\C21134_STQA  
\\chromedriver.exe");  
  
WebDriver driver = new ChromeDriver();  
  
driver.get("https://www.facebook.com/");  
  
WebElement img =  
driver.findElement(By.xpath("//*[@id='email']]"));  
System.out.println(img.isDisplayed());  
System.out.println(img.isEnabled());  
System.out.println(img.isSelected());  
  
}
```

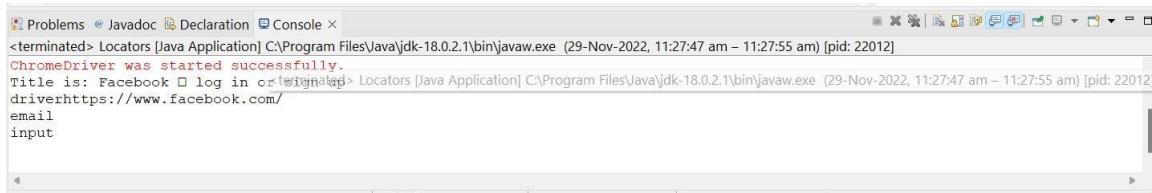
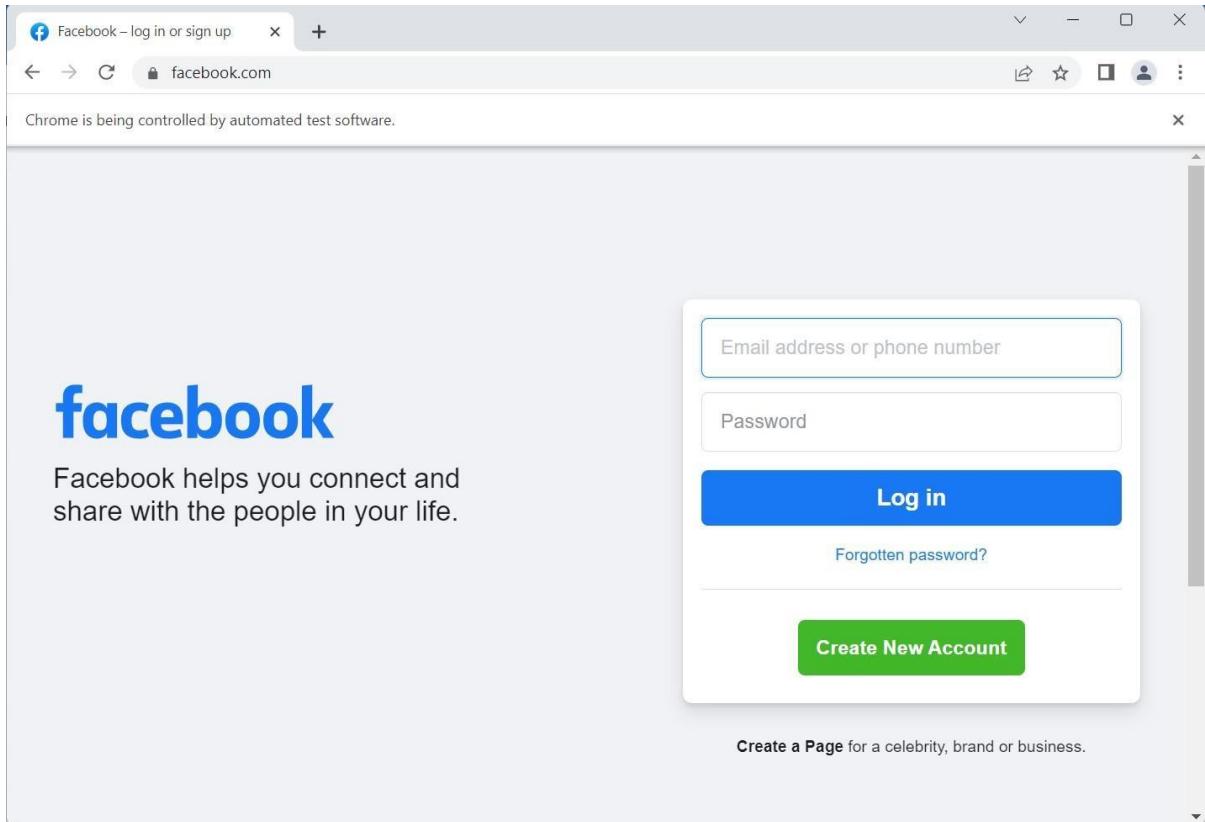
{

Output:**ID Locator:****Code:**

```
package TestCases;  
  
import org.openqa.selenium.By;  
  
import org.openqa.selenium.WebDriver;  
  
import org.openqa.selenium.WebElement;
```

```
import org.openqa.selenium.chrome.ChromeDriver;  
  
public class Locators {  
    public static void main(String[] args) {  
        System.setProperty("webdriver.chrome.driver","C:\\Users\\rushi\\OneDrive\\Desktop\\C21134_STQA  
        \\chromedriver.exe");  
        WebDriver driver = new ChromeDriver();  
        driver.get("https://www.facebook.com/");  
        String title = driver.getTitle();  
        System.out.println("Title is: " + title);  
        System.out.println("driver"+driver.getCurrentUrl());  
        WebElement unm= driver.findElement(By.id("email"));  
        System.out.println(unm.getAttribute("id"));  
        System.out.println(unm.getTagName());      //Html tag name is ID  
    }  
}
```

Output:



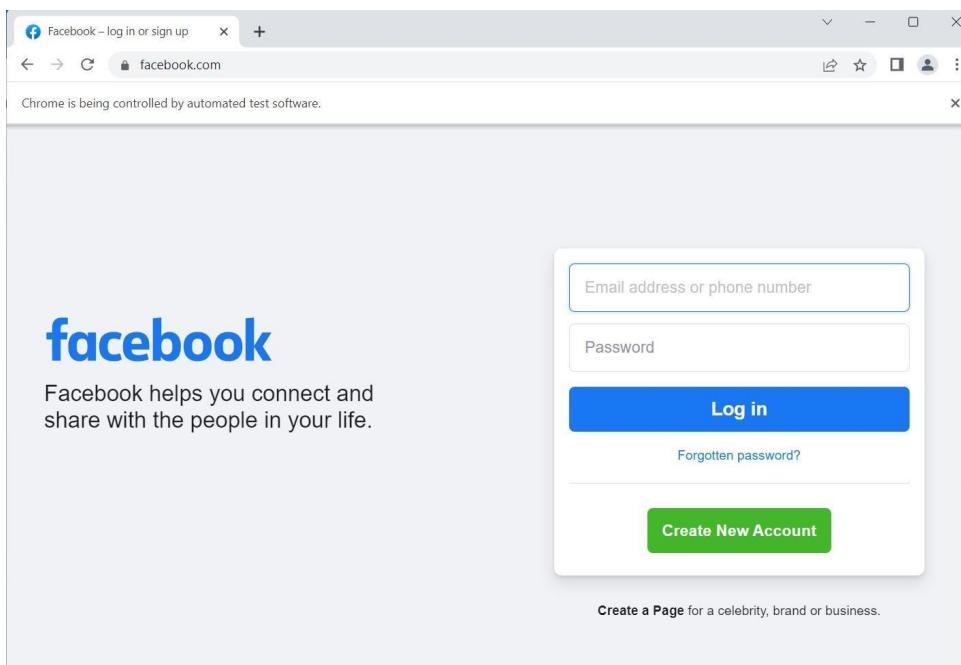
Locator By Using CSS Selector:

Code:

```
package TestCases;  
  
import org.openqa.selenium.By;  
  
import org.openqa.selenium.WebDriver;  
  
import org.openqa.selenium.WebElement;  
  
import org.openqa.selenium.chrome.ChromeDriver;
```

```
public class Locators {  
    public static void main(String[] args) {  
        System.setProperty("Webdriver.chrome.driver","C:\\Users\\rushi\\On  
eDrive\\Desktop\\C21134_STQA  
\\chromedriver.exe");  
  
        WebDriver driver = new ChromeDriver();  
        driver.get("https://www.facebook.com/");  
  
        WebElement el =  
        driver.findElement(By.cssSelector("input[name='email']"));  
        System.out.println(el.getTagName());  
  
    }  
}
```

Output:





The screenshot shows the Eclipse IDE's Console view. The title bar says 'Locators [Java Application]'. The log output is as follows:

```
<terminated> Locators [Java Application] C:\Program Files\Java\jdk-18.0.2.1\bin\javaw.exe (29-Nov-2022, 11:44:29 am - 11:44:36 am) [pid: 18296]
Starting ChromeDriver 107.0.5304.62 (leec40d3a5764881c92085aaee66d25075c159aa-refs/branch-heads/5304@{#942}) on port 26241
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
```

Demonstrate different types of alerts

Code:

```
package TestCases;

import org.openqa.selenium.Alert;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
public class Alerts {

    public static void main(String[] args) {
        System.setProperty("Webdriver.chrome.driver","C:\\Users\\rushi\\OneDrive\\Desktop\\C21134_STQA
        \\chromedriver.exe");

        WebDriver driver = new FirefoxDriver();
        //driver.get("https://www.google.com");
        driver.navigate().to("https://demo.guru99.com/test/delete_customer.p
```

```
hp"); driver.findElement(By.name("cusid")).sendKeys("35");
driver.findElement(By.name("submit")).click();

Alert alt = driver.switchTo().alert();

String alert_msg = driver.switchTo().alert().getText();
System.out.println(alert_msg);

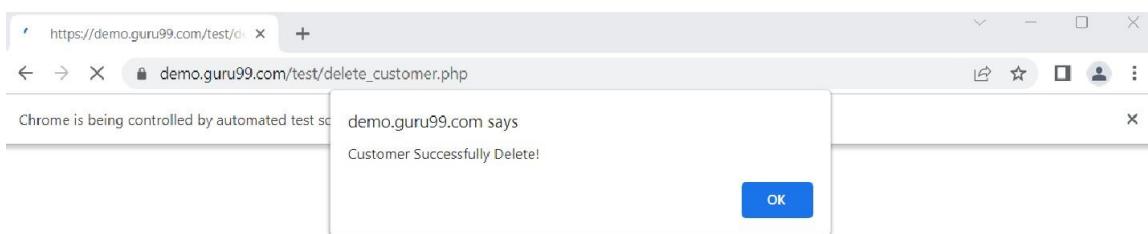
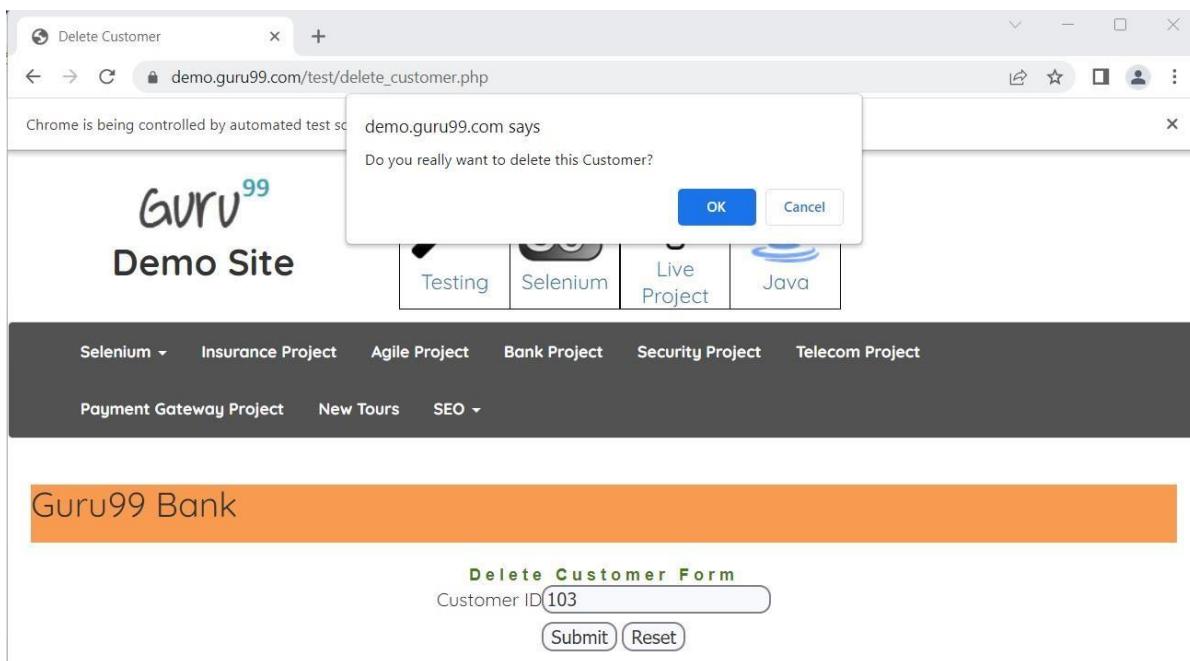
try {
    Thread.sleep(500);
} catch (InterruptedException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}

alt.accept();

System.out.println("Alert Box done");

}
```

Output:



11] Demonstrate:

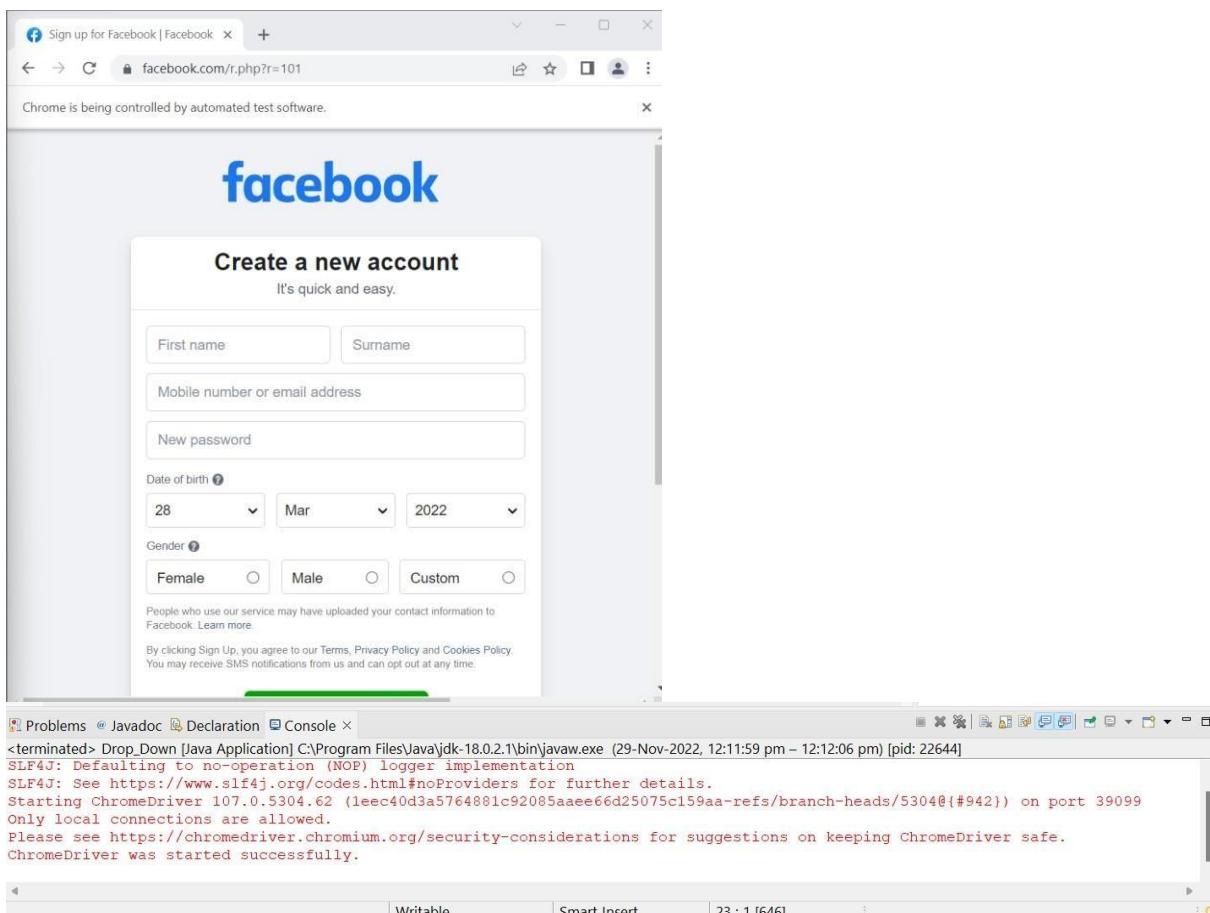
Handling Drop Down

Code:

```
package TestCases;  
  
import org.openqa.selenium.By;  
  
import org.openqa.selenium.WebDriver;
```

```
import org.openqa.selenium.chrome.ChromeDriver;  
import org.openqa.selenium.support.ui.Select;  
  
public class Drop_Down {  
  
    public static void main(String[] args) {  
        System.setProperty("webdriver.chrome.driver","C:\\Users\\rushi\\OneDrive\\Desktop\\C21134_STQA  
        \\chromedriver.exe");  
  
        WebDriver driver = new ChromeDriver();  
  
        //driver.get("https://www.google.com");  
        driver.navigate().to("https://www.facebook.com/r.php?r=101");  
        Select month= new Select(driver.findElement(By.id("month")));  
  
        //month.deselectByVisibleText("Oct"); month.selectByIndex(2);  
    }  
}
```

Output:



List Boxes

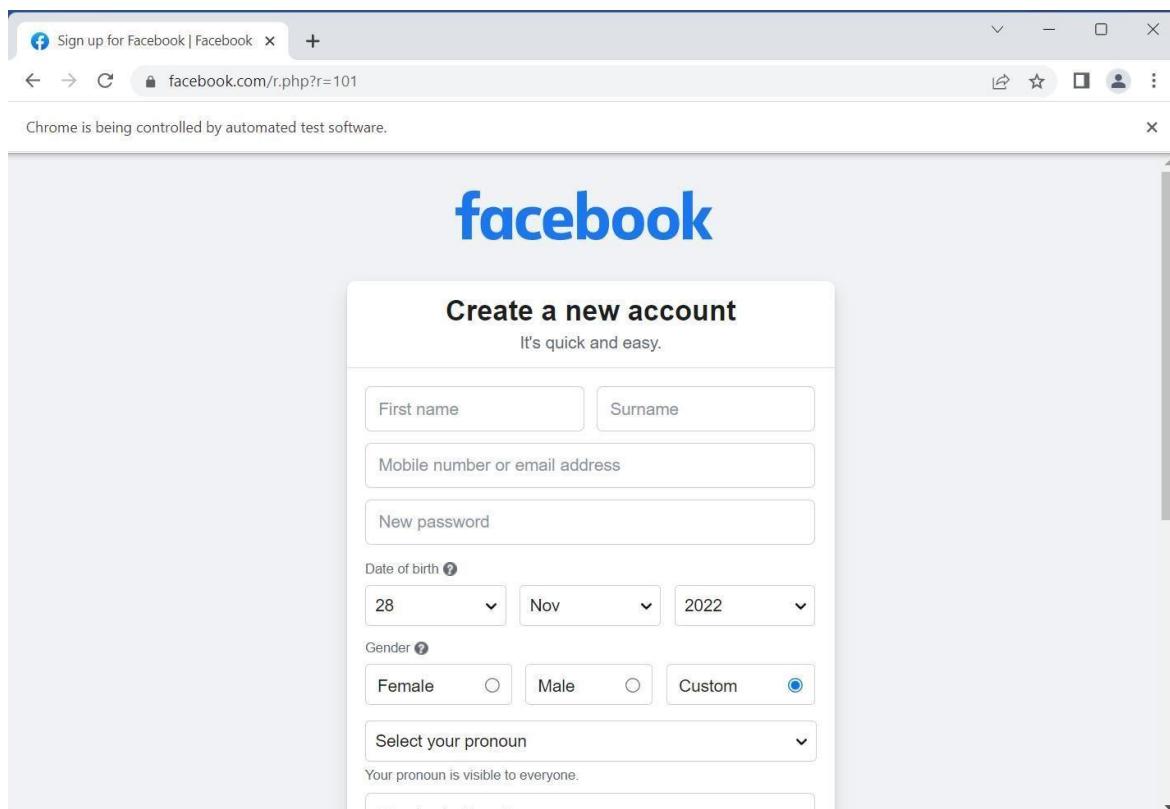
Code:

```
package TestCases;  
  
import java.util.List;  
  
import org.openqa.selenium.By;  
  
import org.openqa.selenium.WebDriver;  
  
import org.openqa.selenium.WebElement;  
  
import org.openqa.selenium.chrome.ChromeDriver;  
  
import org.openqa.selenium.support.ui.Select;
```

```
public class Drop_Down {  
    public static void main(String[] args) {  
        System.setProperty("Webdriver.chrome.driver","C:\\Users\\rushi\\OneDrive\\Desktop\\C21134_STQA  
        \\chromedriver.exe");  
        WebDriver driver = new ChromeDriver();  
        //driver.get("https://www.google.com");  
        driver.navigate().to("https://www.facebook.com/r.php?r=101"); Select  
        month= new Select(driver.findElement(By.id("month"))); List  
        <WebElement> gender = driver.findElements(By.name("sex")); int  
        cnt = gender.size();  
  
        System.out.println(cnt);
```

```
for(int i=0;i<=cnt-1;i++)  
{  
String text = gender.get(i).getText(); System.out.println(text);  
gender.get(i).click();  
try {  
    Thread.sleep(200);  
} catch (InterruptedException e) {  
    // TODO Auto-generated catch block  
    e.printStackTrace();  
}  
}  
}  
}  
}
```

Output:





The screenshot shows the Eclipse IDE interface with a 'Console' tab selected. The output window displays the following text:

```
<terminated> Drop_Down [Java Application] C:\Program Files\Java\jdk-18.0.2.1\bin\javaw.exe (29-Nov-2022, 12:49:40 pm – 12:49:48 pm) [pid: 14676]
ChromeDriver was started successfully.
3
```

Command Button

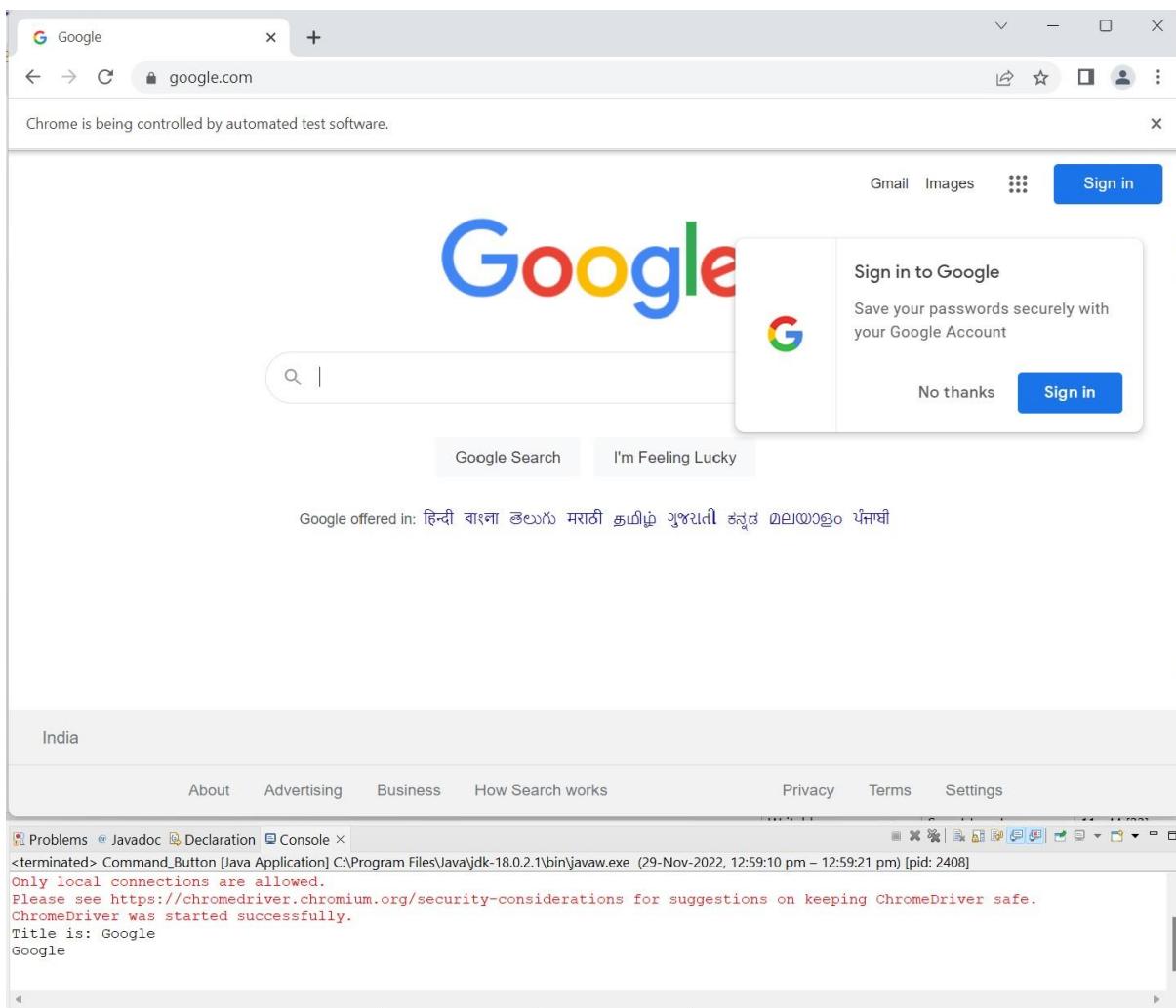
Get Title:

```
package TestCase;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class Command_Button {

    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver","C:\\\\Users\\\\rushi\\\\OneDrive\\\\Desktop\\\\C21134_STQA
        \\\\chromedriver.exe");

        WebDriver driver = new ChromeDriver();
        driver.get("https://www.google.com/"); String title = driver.getTitle();
        System.out.println("Title is: "+title);
        System.out.println(driver.getTitle());
    }
}
```



Radio buttons & text boxes

First, we Create the html file:

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>
```

```
</head>

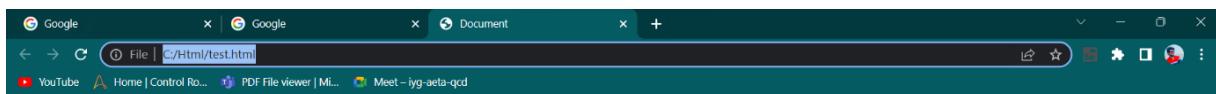
<body>

<h1>Html page</h1>

<input type="radio" name="course" value="MCA">MCA<br><br>
<input type="radio" name="course" value="MMS">MMS<br><br>
<input type="radio" name="course" value="BCA">BCA<br><br>

</body>

</html>
```



Code:

```
package TestCases;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Radio_Button {

    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver","C:\\Users\\rushi\\OneDrive\\Desktop\\C21134_STQA
        \\chromedriver.exe");
    }
}
```

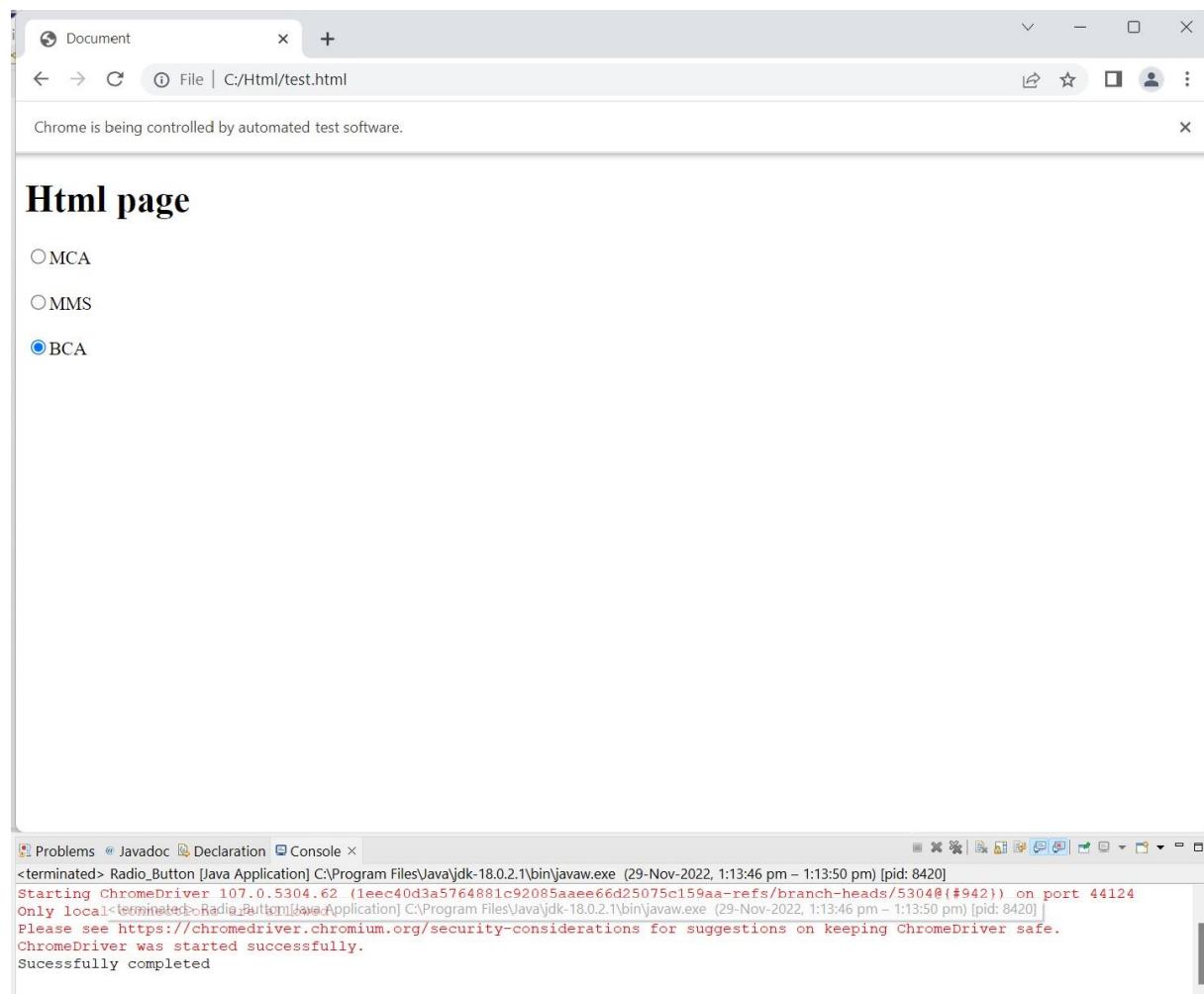
```
WebDriver driver = new ChromeDriver();

//driver.get("https://www.google.com");
driver.navigate().to("file:///C:/Html/test.html");
driver.findElement(By.xpath("/html/body/input[3]")).click();
System.out.println("Sucessfully completed");

}
```

```
}
```

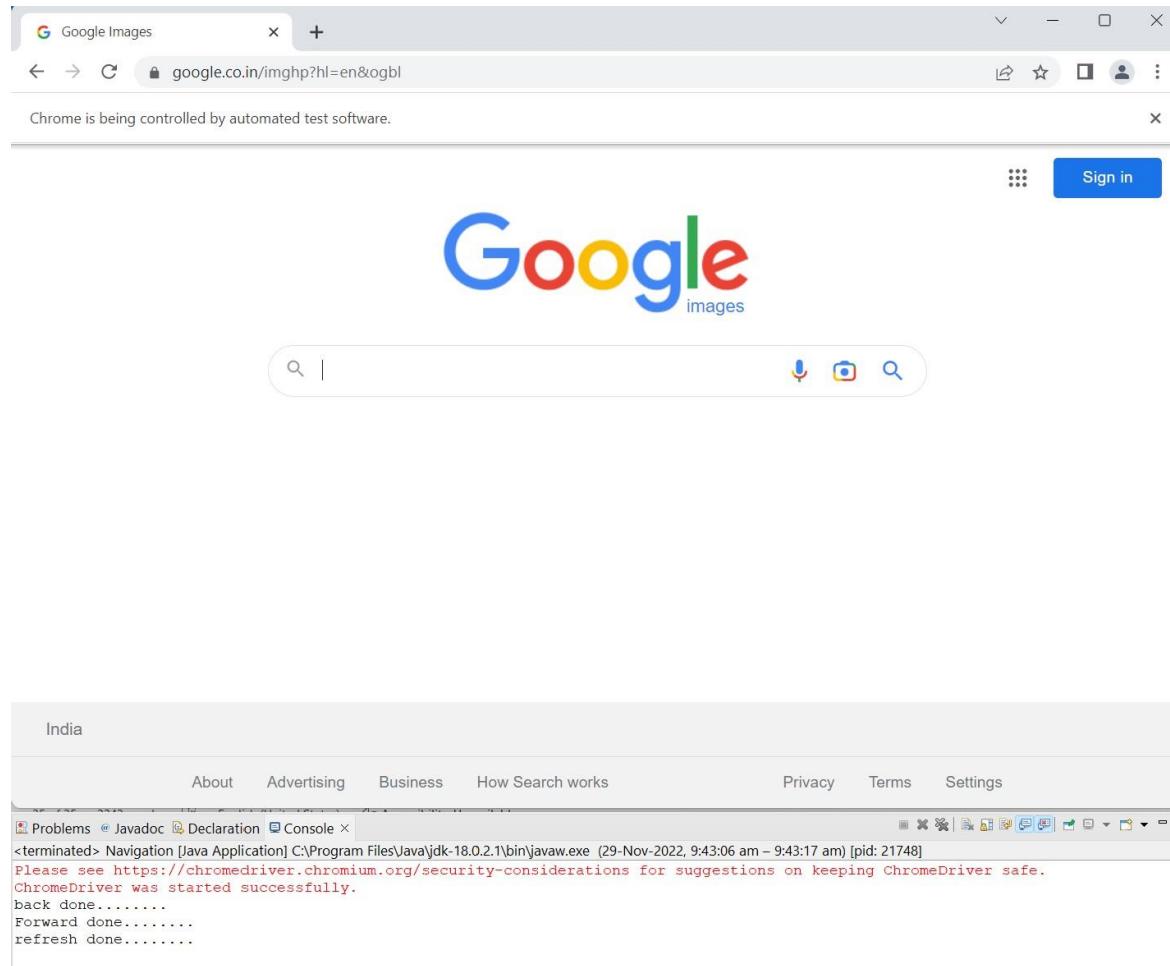
Output:



Waits command in selenium

Thread.sleep() method use for wait command for the process. which we can see the process very easily.

Code:



Installation of TestNG, Running TestNG:

TestNg

TestNG is an advance framework designed in a way to leverage the benefits by both the developers and testers.

TestNG is an automation testing framework in which NG stands for “Next Generation”. TestNG is inspired by [JUnit](#) which uses the annotations (@). TestNG overcomes the disadvantages of JUnit and is designed to make [end-to-end testing](#) easy.

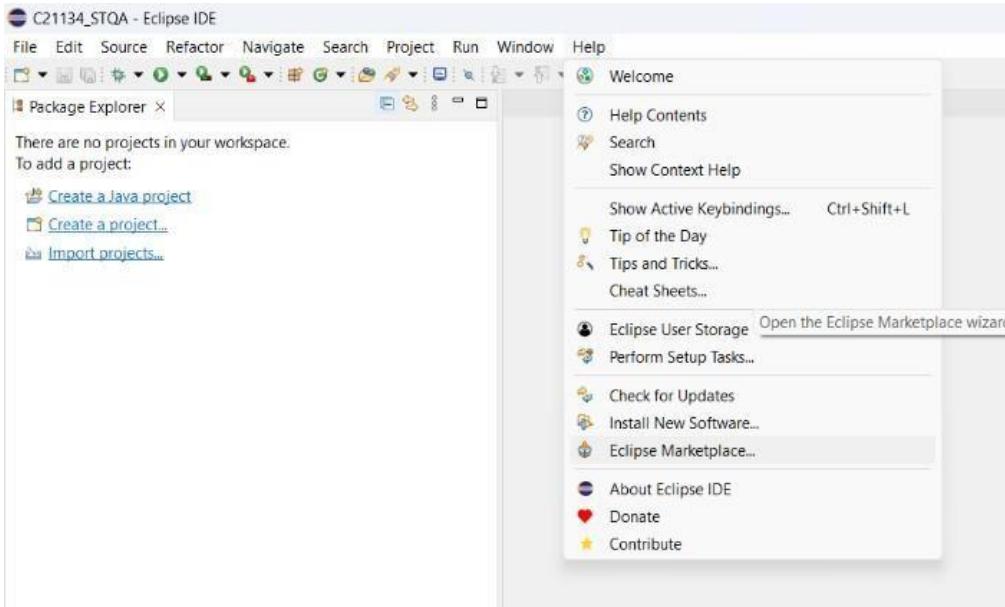
Using TestNG, you can generate a proper report, and you can easily come to know how many test cases are passed, failed, and skipped. You can execute the failed test cases separately.

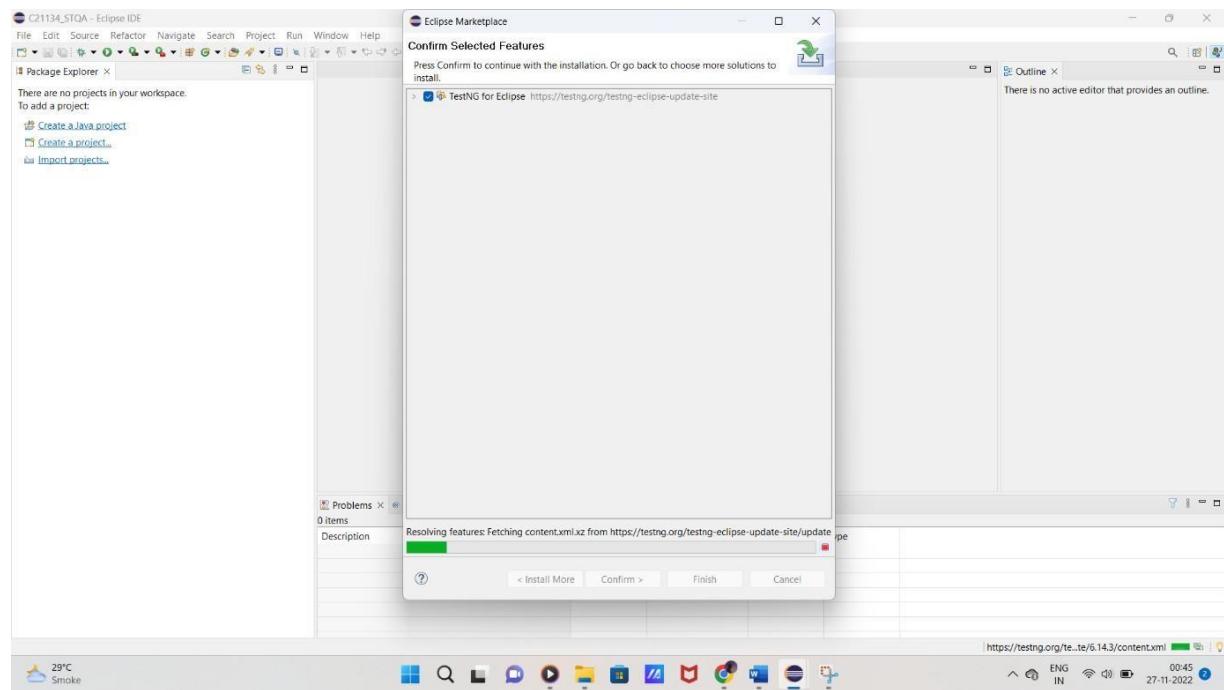
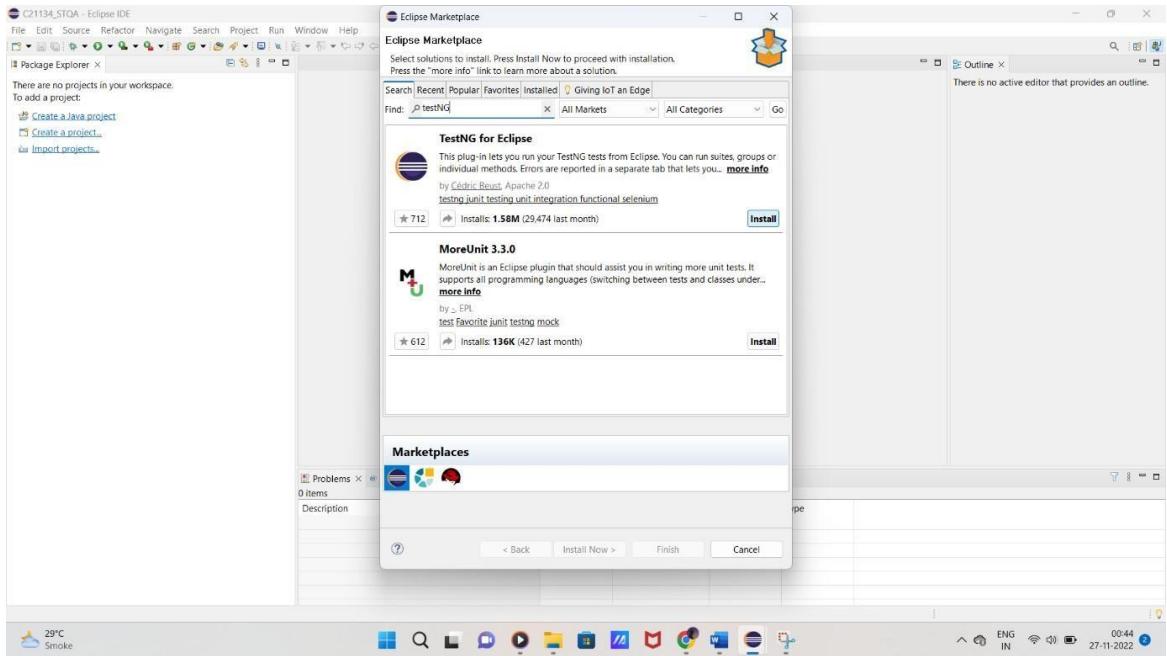
For example:

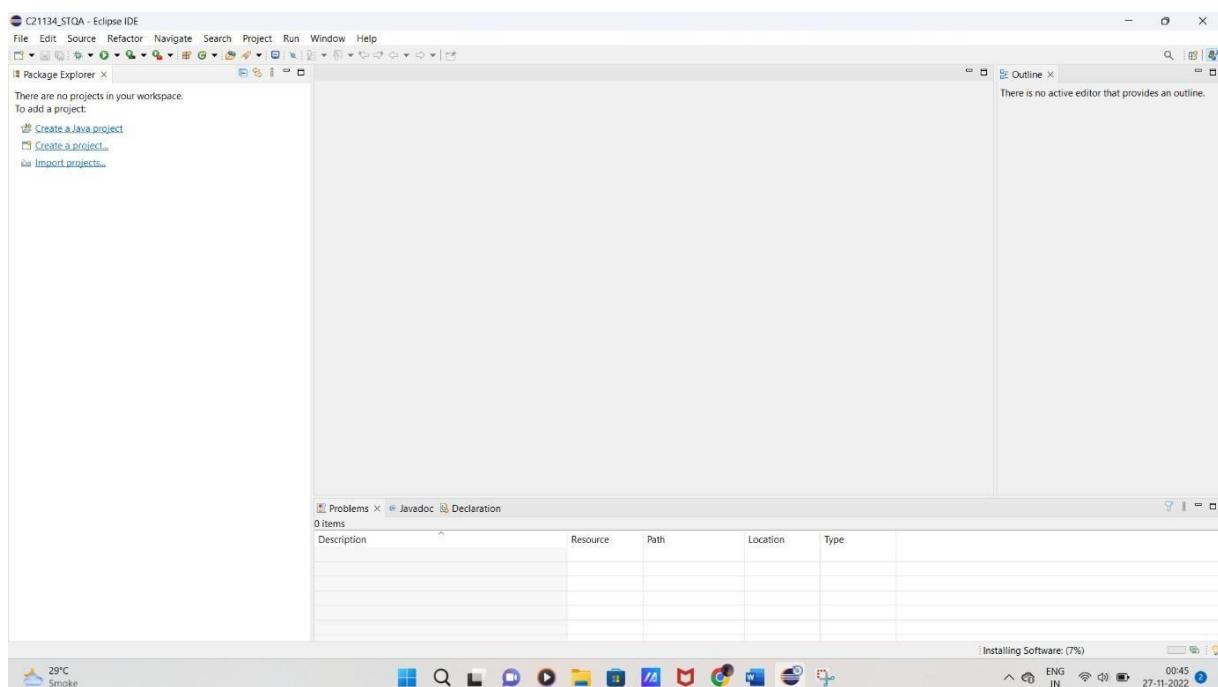
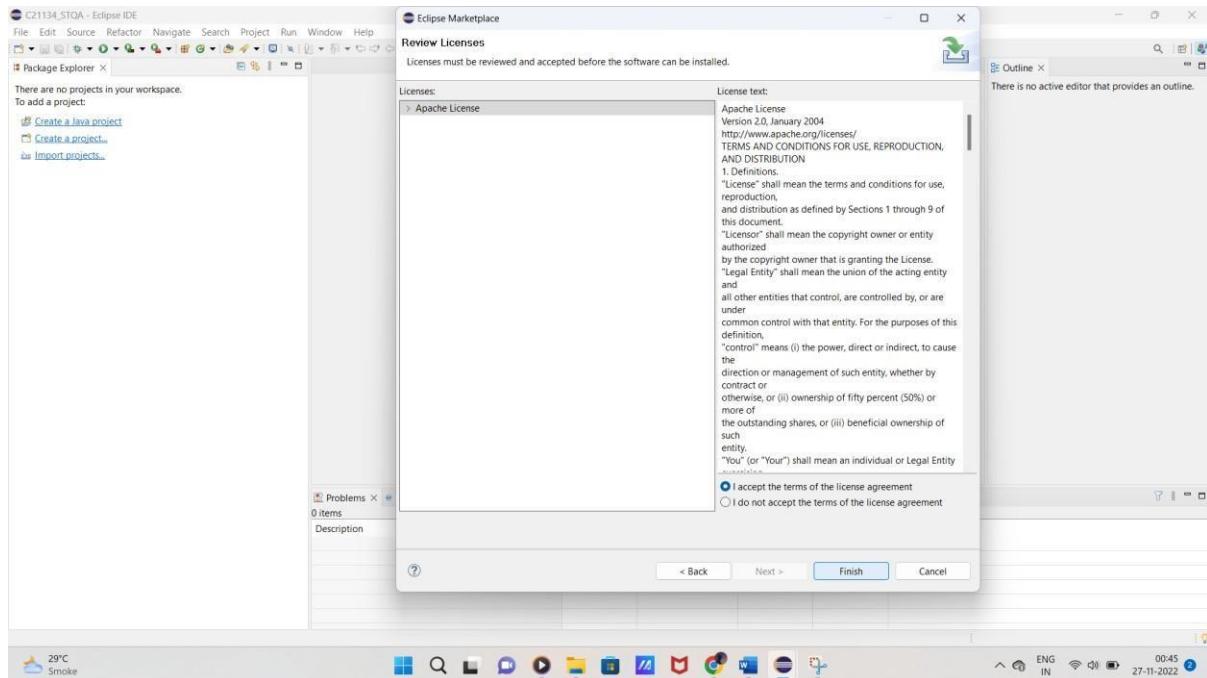
Suppose, you have five test cases, one method is written for each test case (Assume that the program is written using the main method without using testNG). When you run this program first, three methods are executed successfully, and the fourth method is failed. Then correct the errors present in the fourth method, now you want to run only fourth method because first three methods are anyway executed successfully. This is not possible without using TestNG.

The TestNG in Selenium provides an option, i.e., testng-failed.xml file in test-output folder. If you want to run only failed test cases means you run this XML file. It will execute only failed test cases.

Installation



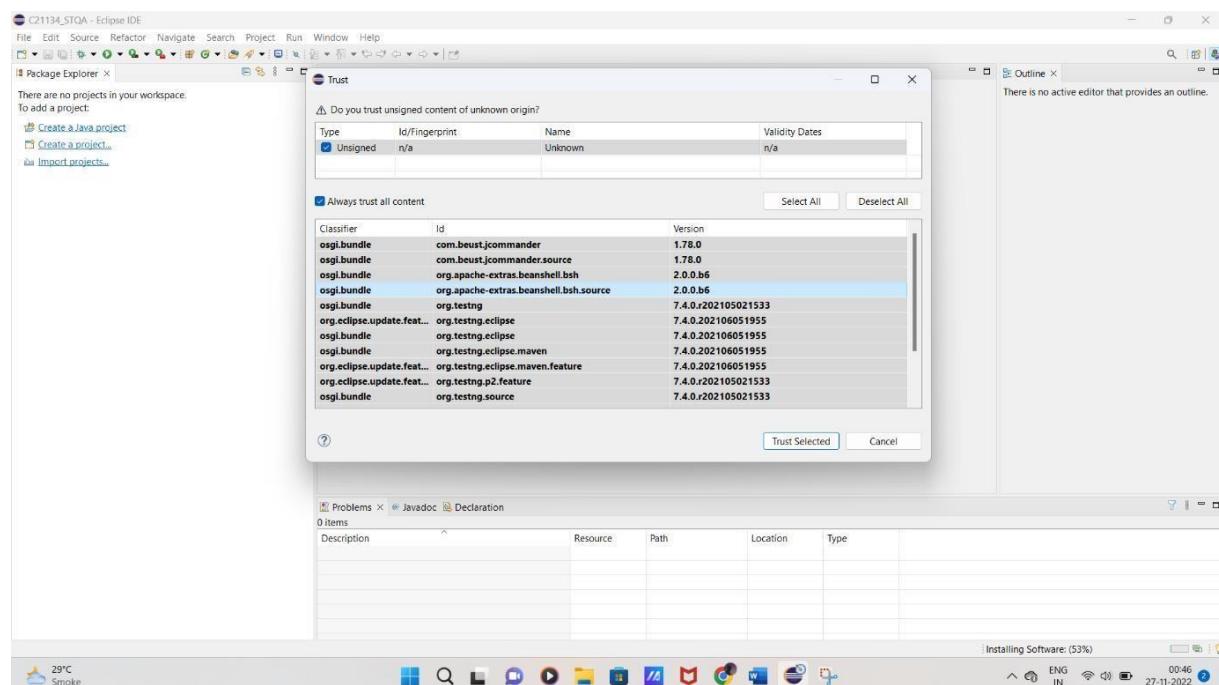




Step 5) After accept the licence agreement, wait for installation.
ignore security warning if occurs

If you encounter a Security warning, just click “Install Anyway”.

Or select all trust issues

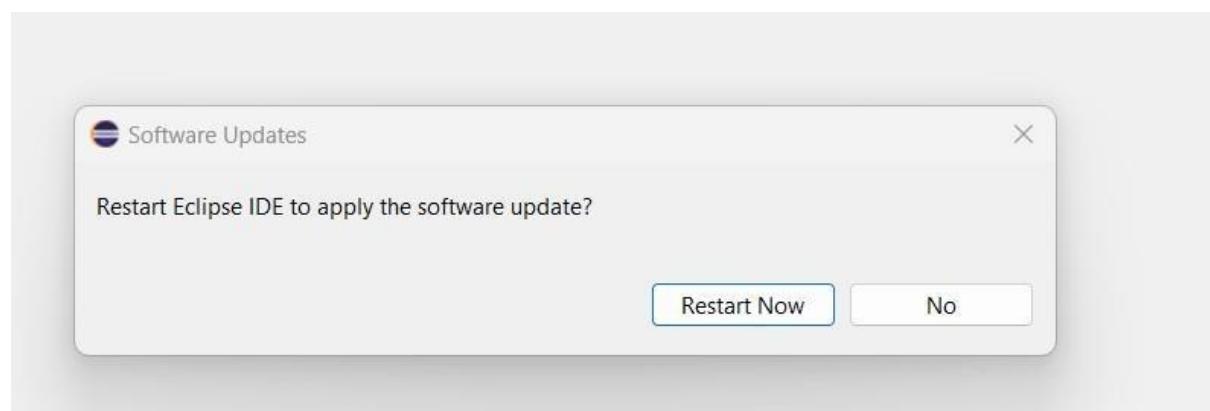


Finish the installation and restart your system

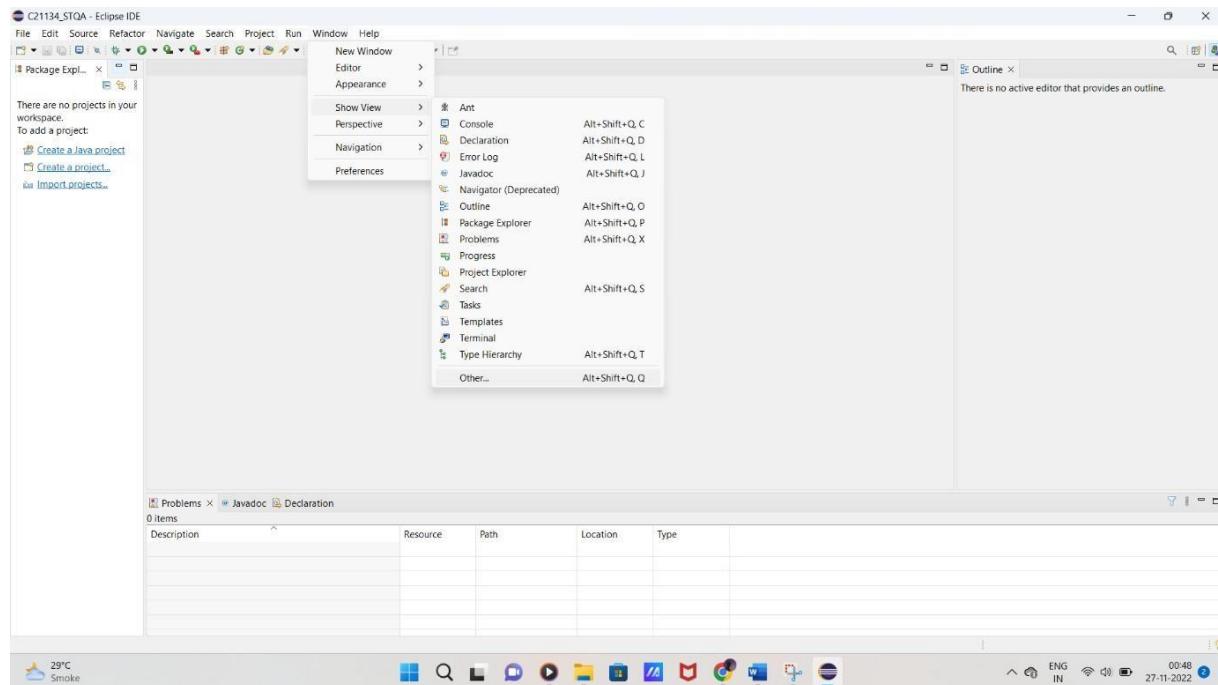
Wait for the TestNG install in Eclipse to finish. When Eclipse prompts you for a restart, click “Restart now.”

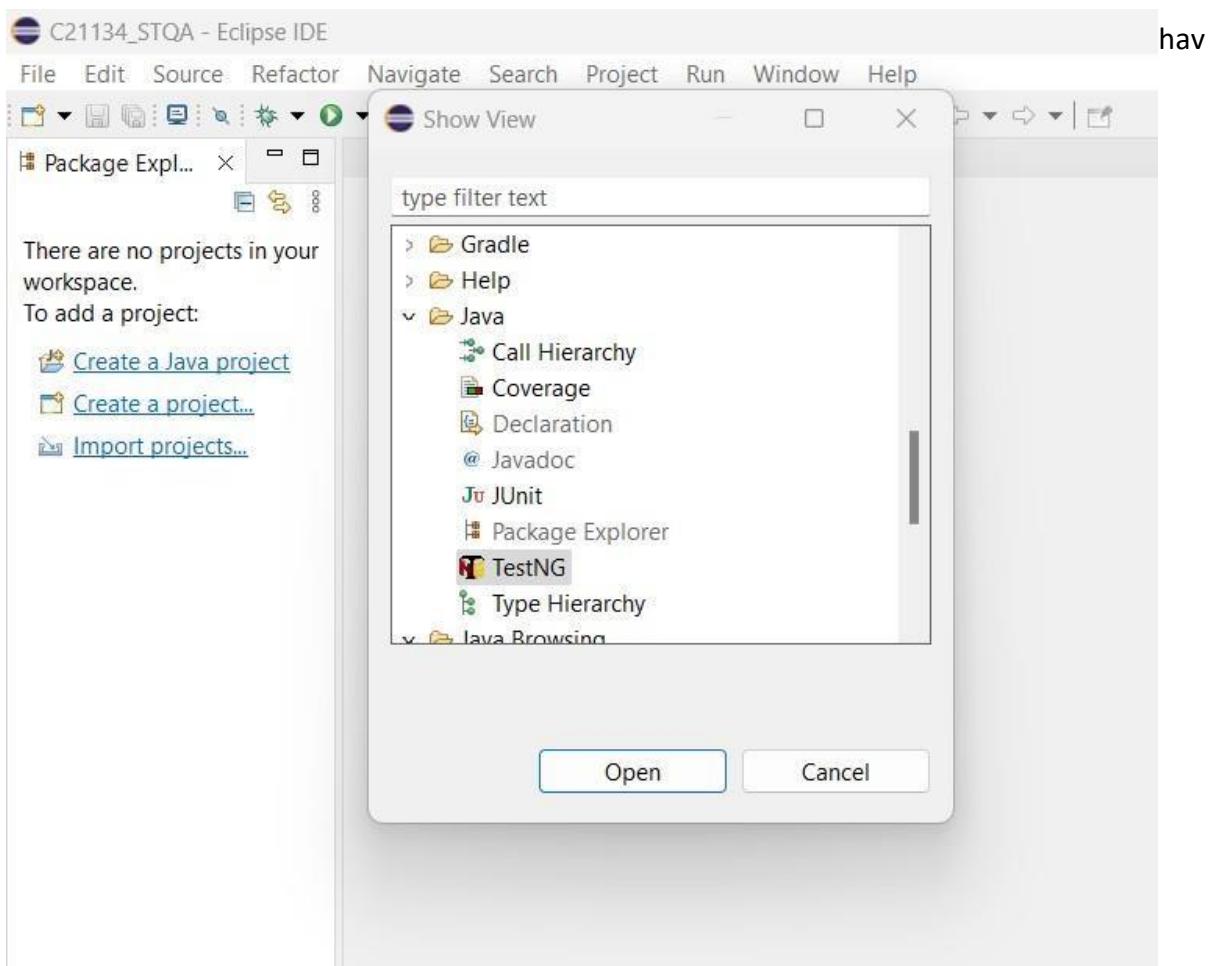
Verify if the installation is done properly

After the restart, verify if TestNG for Eclipse was indeed successfully



installed. Click Window > Show View > Other.





Click on open and that's it..Now see the changes in console.. It has opened Test NG Terminal also



Running TestNG Annotations:

TestNG1.java:

```
package TestNGFrameworkDemo;  
  
import org.openqa.selenium.By;  
  
import org.openqa.selenium.Keys;  
  
import org.openqa.selenium.WebDriver;
```

```
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.Test;
public class TestNG1 {
    public String baseUrl = "https://www.google.com/";
    String driverPath = "E:\\C21134
Hrushikesh\\chromedriver\\chromedriver.exe";
    public WebDriver driver;
    @Test
    public void f() throws InterruptedException {
        System.out.println("Launching chrome browser");
        System.setProperty("Webdriver.chrome.driver", driverPath); driver =
new ChromeDriver();
        driver.get(baseUrl);
        driver.findElement(By.name("q")).sendKeys("DES NMITD",
Keys.ENTER); Thread.sleep(2000);
    }
}
```

TestNG2.java:

```
package TestNGFrameworkDemo; import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.Test;
public class TestNG2 { @Test
```

```
public void TestGoogle() throws InterruptedException {  
    System.setProperty("webdriver.chrome.driver", "E:\\C21134  
Hrushikesh\\chromedriver\\chromedriver.exe");  
  
    WebDriver driver = new ChromeDriver();  
    driver.get("https://www.google.co.in/");  
    driver.findElement(By.name("q")).sendKeys("DES's  
NMITD", Keys.ENTER); Thread.sleep(2000);  
  
}
```

@Test

```
public void TestFacebook() throws InterruptedException {  
    System.setProperty("webdriver.chrome.driver", "E:\\C21134  
Hrushikesh\\chromedriver\\chromedriver.exe");  
  
    WebDriver driver = new ChromeDriver();  
    driver.get("https://www.facebook.com/");  
    driver.findElement(By.name("email")).sendKeys("DES's  
NMITD", Keys.ENTER); Thread.sleep(2000);  
}  
}
```

Output:

Facebook – log in or sign up

facebook.com/login/?privacy_mutation_token=eyJ0eXAiOiowLCJcmVhdGlvbi90aW1ljo... ↗ ☆ ☐ ⋮

Chrome is being controlled by automated test software.

Is this your account?

 facebook

Facebook helps you share with the people

NNST Program on IoT by DES's NMITD
DES's NMITD • Not you?

We couldn't find an account that matches what you entered, but we've found one that closely matches.

Yes, Continue

Phone number

Log in

Forgot password?

New Account

Create a Page for a celebrity, brand or business.

C22052

STQA Lab

Jui Jadhav

Google search results for "DES's NMITD". The search bar shows "DES's NMITD". Below the search bar, there are links for All, News, Maps, Images, Videos, and More. The "Tools" button is also visible. The search results include a snippet for "Deccan Education Society's Navinchandra ...". It shows the logo of Deccan Education Society's Navinchandra, the text "Educational institution in Mumbai, Maharashtra", and two buttons: "Overview" and "Course admissions". To the left of the snippet, there is a link to the website "https://www.nmitd.edu.in". Below the snippet, there are sections for "About", "Address", "Phone", and a "Claim this knowledge panel" button. On the right side of the snippet, there are sections for "Apply Now", "Placements", and "Student Login".

Google DES's NMITD - Google Search

google.co.in/search?q=DES%27s+NMITD&source=hp&ei=La10Y729AsGRoATI_Zb4Cg&iflsig=AJiK0e8AAAA...

Chrome is being controlled by automated test software.

DES's NMITD

All News Maps Images Videos More Tools

About 2,350 results (0.48 seconds)

Deccan Education Society's Navinchandra ...

Educational institution in Mumbai, Maharashtra

Overview Course admissions

<https://www.nmitd.edu.in>

nmitd

Being a part of Navinchandra Mehta Institute was the best decision. This institute not only helped to gain theoretical knowledge but also empowered me to face ...

Apply Now

Expert Faculty, Learning Environment, Better Placements ...

Placements

NMITD belongs to Deccan Education Society, Pune, which ...

Student Login

About

nmitd.edu.in

Address: 2RCJ+GHH, DES Mumbai Camp Off-, College St, Chandrakant, Dhuru Wadi, West, Dadar, Mumbai, Maharashtra 400028

Phone: 022 6276 4583

Claim this knowledge panel

Facebook – log in or sign up

facebook.com/login/?privacy_mutation_token=eyJ0eXAiOiowLCJjcmVhdGlvbI90aW1lloxNjY4NTkwODg1LCJjYWsc...

Chrome is being controlled by automated test software.

Facebook

Facebook helps you connect and share with the people in your life.

DES's NMITD

Password

Log in

Forgotten password?

Create New Account

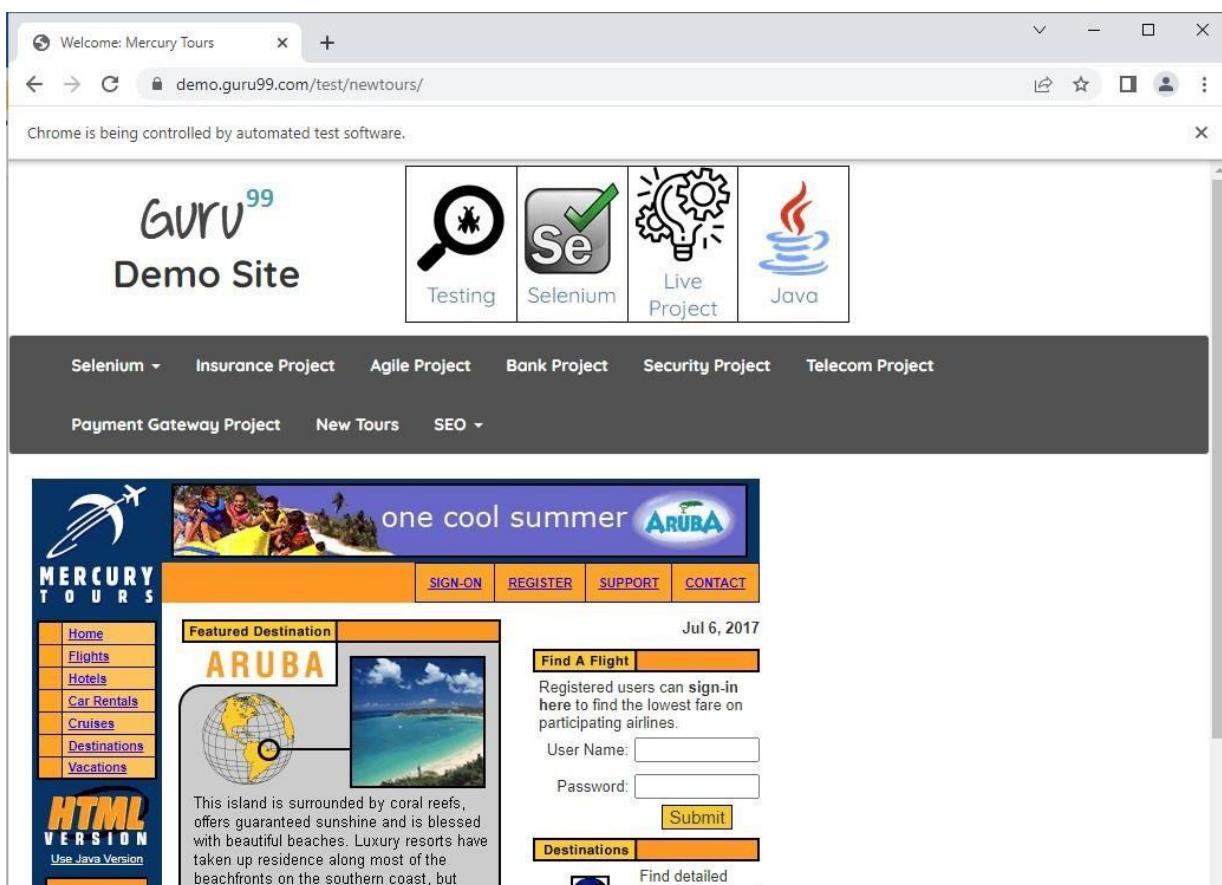
package TestNGFrameworkDemo;

```
import org.testng.annotations.Test;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.chrome.ChromeDriver;  
import org.testng.annotations.BeforeTest;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.chrome.ChromeDriver;  
import org.testng.Assert;  
import org.testng.annotations.AfterTest;  
public class TestNG3 {  
    String actualTitle,expectedTitle; @BeforeTest  
    public void launchBrowser() {  
        System.out.println("launching Google chrome browser");  
        System.setProperty("webdriver.chrome.driver", "E:\\C21134  
        Hrushikesh\\chromedriver\\chromedriver.exe");  
        WebDriver driver = new ChromeDriver();  
        driver.get("https://demo.guru99.com/test/newtours/");  
    }  
    @Test  
    public void verifyHomepageTitle() {  
  
        String expectedTitle = "Welcome: Mercury Tours"; String ActualTitle  
        = driver.getTitle(); Assert.assertEquals(actualTitle, expectedTitle);  
    }  
}
```

@AfterTest

```
public void terminateBrowser() { driver.close();  
}  
}
```

Output:



Demonstrate Validation testing

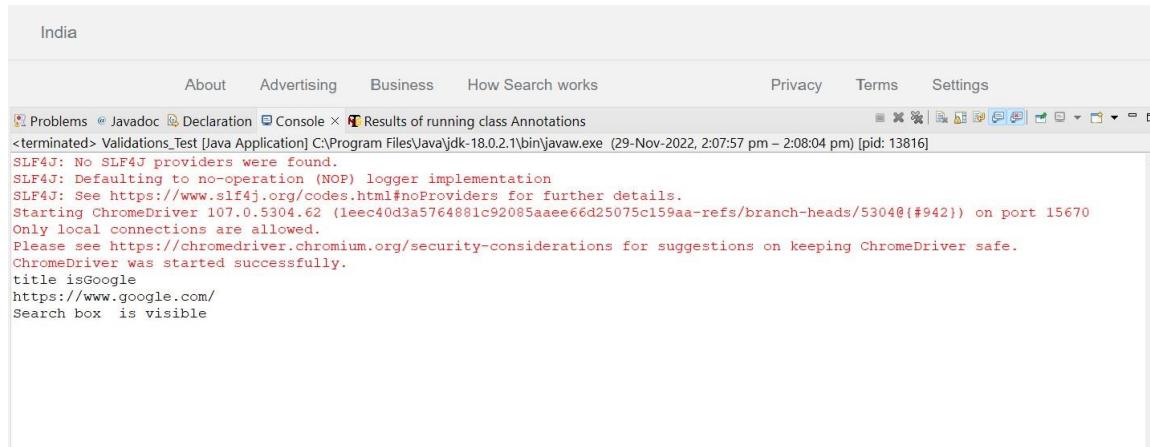
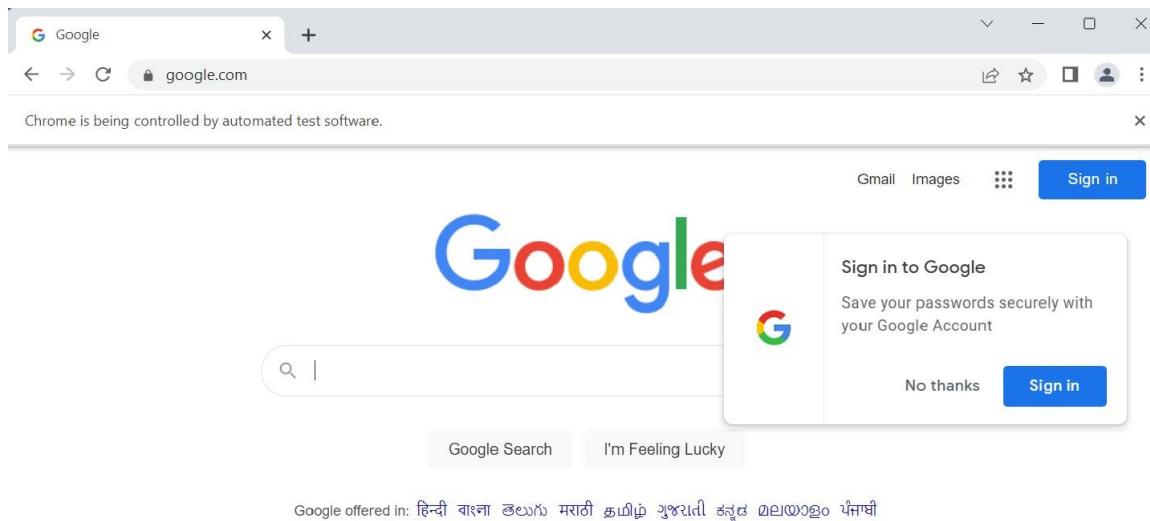
Code:

```
package TestCases;
```

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class Validations_Test {
    public static void main(String []args)
    {
        System.setProperty("Webdriver.chrome.driver","C:\\STQA\\Drivers\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.google.com/");
        String title = driver.getTitle(); // Check the validation of title
        System.out.println("title is" +title);
        System.out.println(driver.getCurrentUrl()); //Check the validation of URL
        // # Check the validation of the Search box is Visible are not.
        boolean status =
        driver.findElement(By.xpath("/html/body/div[1]/div[3]/form/div[1]/div[1]/div[1]")).isDisplayed();
        if(status){
            System.out.println("Search box is visible");
```

```
} else {  
}  
}  
}  
}
```

Output:



Perform regression testing

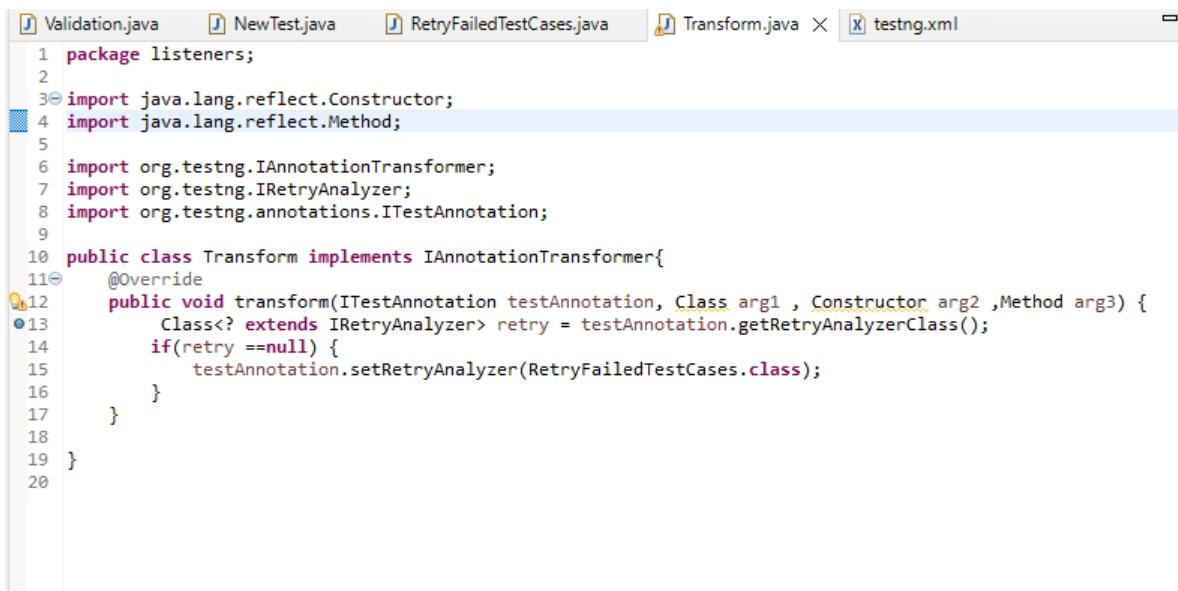
Retrying failed test cases using listeners in TestNG.

Code:



The screenshot shows an IDE interface with several tabs at the top: Validation.java, NewTest.java, RetryFailedTestCases.java (selected), Transform.java, and testng.xml. The code in the editor is as follows:

```
1 package listeners;
2
3 import org.testng.IRetryAnalyzer;
4
5 public class RetryFailedTestCases implements IRetryAnalyzer {
6     int noOfRetries =0;
7     int maximumRetries = 3;
8
9
10    @Override
11    public boolean retry(ITestResult result) {
12
13        // TODO Auto-generated method stub
14        if(noOfRetries < maximumRetries) {
15            System.out.println(result.getName()+" no of retry is" + (noOfRetries+1));
16            noOfRetries++;
17            return true;
18        }
19        return false;
20    }
21
22 }
```



The screenshot shows an IDE interface with several tabs at the top: Validation.java, NewTest.java, RetryFailedTestCases.java, Transform.java (selected), and testng.xml. The code in the editor is as follows:

```
1 package listeners;
2
3 import java.lang.reflect.Constructor;
4 import java.lang.reflect.Method;
5
6 import org.testng.IAnnotationTransformer;
7 import org.testng.IRetryAnalyzer;
8 import org.testng.annotations.ITestAnnotation;
9
10 public class Transform implements IAnnotationTransformer{
11    @Override
12    public void transform(ITestAnnotation testAnnotation, Class arg1 , Constructor arg2 ,Method arg3) {
13        Class<? extends IRetryAnalyzer> retry = testAnnotation.getRetryAnalyzerClass();
14        if(retry ==null) {
15            testAnnotation.setRetryAnalyzer(RetryFailedTestCases.class);
16        }
17    }
18
19 }
```

The screenshot shows the Eclipse IDE interface. On the left, the Project Explorer view displays several Java projects and files under the 'REGRESSION' project. In the center, the Validation.java code editor shows the following Java code:

```
1 package TEST;
2
3 import org.testng.Assert;
4
5 public class NewTest {
6     @Test
7     public void validationRetry() {
8         Assert.assertTrue(false);
9     }
10 }
11
12 }
```

The screenshot shows the Eclipse IDE interface. On the left, the Project Explorer view displays the same 'REGRESSION' project structure. In the center, the testng.xml code editor shows the following XML configuration:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
3 <suite name="Suite">
4     <listeners>
5         <listener class-name ="Listeners.Transform"/>
6     </listeners>
7     <test thread-count="5" name="Testing">
8         <classes>
9             <class name="TEST.NewTest"/>
10        </classes>
11    </test> <!-- Testing -->
12 </suite> <!-- Suite -->
13 
```

Output:

The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The output window displays the results of running the TestNG suite:

```
<terminated> REGRESSION_testng.xml [TestNG] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe (28-Nov-2022)
[RemoteTestNG] detected TestNG version 7.4.0

=====
Suite
Total tests run: 1, Passes: 0, Failures: 1, Skips: 0
=====
```

The screenshot shows the Eclipse IDE interface with the Test View open. The title bar includes 'Problems', '@ Javadoc', 'Declaration', 'Console', 'Results of running suite', 'Tasks', and other icons. Below the title bar, a status bar displays 'Passed: 0', 'Failed: 1', 'Skipped: 0', 'Tests: 1/1 Methods: 1 (202 ms)'. The main area shows a tree view of test results under 'Suite (0/1/0/0) (0.009 s)'. The 'Failed Tests' tab is selected, showing one failure: 'validationRetry' under 'TEST.NewTest'. A detailed stack trace for this failure is provided:

```
java.lang.AssertionError: expected [true] but found
at org.testng.Assert.fail(Assert.java:99)
at org.testng.Assert.failNotEquals(Assert.java:1037)
at org.testng.Assert.assertTrue(Assert.java:45)
at org.testng.Assert.assertTrue(Assert.java:55)
at TEST.NewTest.validationRetry(NewTest.java:9)
at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
at java.base/java.util.ArrayList.forEach(ArrayList.java:151)
at org.testng.remote.AbstractRemoteTestNG.run(AbstractRemoteTestNG.java:115)
```