NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

Deccan Education Society’s

Navinchandra Mehta Institute of Technology and Development

CERTIFICATE

This is to certify that Mr. **Vaishnavi Shahaji Patil** of M.C.A. Semester III with Roll No. **C23090** has completed **\_All\_** practicals of MCAL35 **Software Testing** under my supervision in this college during the year 2024- 2025.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CO | R1  (Attendance) | R2  (Performance during lab session) | R3  (Innovation in problem solving technique) | R4  (Mock Viva) | R5  (Variation in implementation of learnt topics on projects) |
| CO1 |  |  |  |  |  |
| CO2 |  |  |  |  |  |
| CO3 |  |  |  |  |  |
| CO4 |  |  |  |  |  |

Practical-in-charge Head of Department MCA Department (NMITD)

pg. 1

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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   1. 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   1. 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.** | 1. Prepare test cases on ATM Machin using Unit and System Testing 2. Prepare a test cases on Login Page of Gmail using Unit and System Testing 3. Prepare a test cases on Calculator using Unit and System Testing |  | **CO1** |
| **Automation Testing** | | | | |
| **4.** | **Introduction to Selenium**   1. Write down a steps and process of Selenium IDE Installation on any 1 browser.(eg.Firefox) 2. Record and run test cases on demotour website for login page through Mozila Firefox. 3. Record and run test cases on demotour website for login page through Google Chrome. 4. Record and run test cases on Registration form of any website 5. 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 : |  | **CO2** |  |

pg. 2

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | * Handling Drop Down, * List Boxes * Command Button, * Radio buttons & text boxes. * Waits command in selenium |  |  |  |
| **12** | Installation of TestNg , running testNg and TestNg annotations |  | **CO3** |  |
| **13** | Demonstrate Validation testing |  | **CO3** |  |
| **14** | Perform regression testing |  | **CO4** |

pg. 3

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

# PRACTICAL NO.1: Implementation of Black Box Testing

Prepare test cases for below given applications using Boundary Value Analysis and Equivalence Class Partitioning:

1. BMI Calculator:

Input: Person’s Weight and Height

Output: Body Mass Index

The range of Weight: 3 kg to 300 kg

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| S.No | Action | Inputs | Expected Output | Actual Output | Test Browser | Test Result | Test Comments |
| 1 | Putting the data inside the Weight and Height Column | Weight: 84 kg,  Height: 182 cm | Result: 25.4 | Result: 25.4 | nan | PASS | Valid BMI value |
| 2 | Putting the data inside the Weight Column \*maximum weight accpetance is 140KG | Weight: 175 kg,  Height: 170 cm | Message  :Enter the weight between 20KG to 140KG | Message: Enter the weight between 20KG to 140KG | nan | PASS | Showing Error when |
| 3 | Putting the data inside the Height Column \*maximum height accpetance is 220CM | Weight: 60 kg,  Height: 220 cm | Message: Enter the height between 90CM  to 220CM | Message: Enter the height between 90CM  to 220CM | nan | PASS | Showing message for missing values |
| 4 | Clicking the 'Calculate' button when Height and Weight Text Input box is empty | Weight: EMPTY kg, Height: EMPTY cm | Message: The values cannot be empty. | Message: The values cannot be empty. | nan | PASS | Showing message for missing values |
| 5 | Clicking the 'Calculate' button when Height Text Input box is empty | Weight: 70 kg, Height: EMPTY cm | Message: Height Box cannot be Empty | Message: Height Box cannot be Empty | nan | PASS | Showing message for missing values |

pg. 4

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 6 | Clicking the 'Calculate' button when Weight Text Input box is empty | Weight: EMPTY kg, Height: 180cm | Message: weight Box cannot be Empty | Message: weight Box cannot be Empty | nan | PASS | Showing message for missing values |
| 7 | Putting the data inside the Weight Column \*minimum weight accpetance is 30KG | Weight: 16 kg,  Height: 170 cm | Message: weight should be more than 30KG | Result: 6.2 | nan | FAIL | Not showing message for Calculating the BMI without showing the message for entering weight between 30KG to 120KG |
| 8 | Putting the data inside the Height Column \*minimum height accpetance is 120CM | Weight=35, Height=100cm | Message: weight should be more than 150CM | Result: 35 | nan | FAIL | Not showing the message Calculating the BMI without showing the message for entering height between 150CM to 260CM |
| 9 | Putting the data inside the Weight and Height Text input Box's and Clicking the 'Calculate' Button | Weight=4KG , Height=26CM | Message: You have entered the Invalid Data to Calculate | Result: 5000 | nan | FAIL | Not showing the message for Calculating the BMI without showing the message for entering height between 150CM to 260CM and Weight between 20KG to 140KG |
| 10 | Putting the data inside the Weight and Height Text input Box's and Clicking the 'Calculate' Button | Weight=200KG, Height=360CM | Message: You have entered the Invalid Data to Calculate | Result: 90 | nan | FAIL | Not showing the message for Calculating the BMI without showing the message for entering height between 150CM to  260CM and Weight |

pg. 5

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  | between 30KG to 120KG |
| 11 | Putting the data inside the Height Column \*maximum height acceptance is 260CM | Weight=70KG, Height=190.75CM | Result: 19.2 | Result: 19.2 | nan | PASS | Valid BMI value |
| 12 | Putting the data inside the Weight Column \*maximum weight acceptance is 120KG | Weight=80.58 KG, Height=185CM | Result: 23.5 | Result: 23.5 | nan | PASS | Valid BMI value |
| 13 | Putting the data inside the Weight and Height Text input Box's and Clicking the 'Calculate' Button | Weight=50.65KG, Height=180.36CM | Result: 15.6 | Result: 15.6 | nan | PASS | Valid BMI value |
| 14 | Click the 'Rest' Button with valid data inside the Weight and Height Input Text Box | Weight: 50 kg,  Height: 170 cm | The Input Box for Height and Weight becomes to empty | The data inside the input box for height and weight remains same. | nan | FAIL | The Input Box for Height and Weight does not changes to empty |
| 15 | Click the 'Rest' Button with valid data inside the Weight Input Text Box | Weight: 50 kg,  Height: 0 CM | The Input Box for Weight becomes to empty | The data inside the input box for weight remains same. | nan | FAIL | The Input Box for Weight does not changes to empty |
| 16 | Click the 'Rest' Button with valid data inside the Height Input Text Box | Weight: 0 kg,  Height: 160 CM | The Input Box for Height becomes to empty | The data inside the input box for height remains same. | nan | FAIL | The Input Box for Height does not changes to empty |

pg. 6

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 17 | Inputting non- numeric characters in the Weight and Height fields. | Weight: "Sumit" kg, Height: "Rai" cm | Message: Please enter numeric values for Weight and Height. | Message: Please enter numeric values for Weight and Height. | nan | PASS | Showing the Message: Please enter numeric values for Weight and Height. |
| 18 | Inputting values that are exactly on the boundary limits. | Weight: 30 kg,  Height: 150 cm | Result: 13.3 | Result: 13.3 | nan | PASS | Valid BMI value |
| 19 | Putting negative values in the Weight input box. | Weight: -50 kg,  Height: 170 cm | Message: Please provide positive weight value. | Message: Please provide positive height value. | nan | PASS | Showing the message "Please provide positive height value." |
| 20 | Putting negative values in the Height input box. | Weight: 50 kg,  Height: -170 cm | Message: Please provide positive height value. | Message: Please provide positive height value. | nan | PASS | Showing the message "Please provide positive weight value." |
| 21 | Putting negative values in the Weight and Height columns. | Weight: -50 kg,  Height: -170 cm | Message: "Please provide positive height value.  Please provide positive weight value." | Message: "Please provide positive height value.  Please provide positive weight value." | nan | PASS | Showing the Message: "Please provide positive height value.  Please provide positive weight value." |

1. 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.

pg. 7

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

Creating the equivalence class partitioning below:

1. Equilateral Triangle: All three sides are equal (e.g., a = b = c).
2. Isosceles Triangle: Exactly two sides are equal (e.g., a = b ≠ c).
3. Scalene Triangle: All three sides are different (e.g., a ≠ b ≠ c).
4. Not a Triangle: The sum of any two sides must be greater than the third side:
   * a + b ≤ c
   * a + c ≤ b
   * b + c ≤ a

Invalid Equivalence Classes

1. Negative Sides: Any side is negative (e.g., a < 0, b < 0, or c < 0).
2. Zero Sides: Any side is zero (e.g., a = 0, b = 0, or c = 0).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Input three sides | Expected Output | Actual Output | Test Comments | Class Type |
| TC1 | 3, 4, 5 | Scalene Triangle | Scalene Triangle | Passed: All sides are different and satisfy the triangle inequality theorem.. | Valid - Scalene |
| TC2 | 5,5,5 | Equilateral Triangle | Equilateral Triangle | Passed: All Three sides are equal. | Valid - Equilateral |
| TC3 | 4, 4, 6 | Isosceles Triangle | Isosceles Triangle | Passed: Two sides are equal. | Valid - Isosceles |
| TC4 | 0,0,0 | Not a Triangle | Equilateral Triangle | Failed: Two sides are equal. | Invalid - Equilateral |
| TC5 | |  | | --- | | -3, -4, -5 |  |  | | --- | |  | | Not a Triangle | Scalene Triangle | FAIL: Negative lengths are not valid for triangle sides. | Invalid - Scalene |
| TC6 | 2, 1, 3 | Not a Triangle | Scalene Triangle | FAIL The sum of two sides is equal to the third side; it is not a valid triangle. | Invalid - Isosceles |

pg. 8

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Input three sides | Expected Output | Actual Output | Test Comments | Class Type |
| TC7 | 3, 3, 0 | Not a Triangle | Isosceles Triangle | FAIL: Expected Not a Triangle but got Isosceles. Logic error detected. | Invalid - Zero |
| TC8 | 3, 4, 5 | Scalene Triangle | Scalene Triangle | Passed: All sides are different. | Valid - Scalene |
| TC9 | 7, 10, 5 | Scalene Triangle | Scalene Triangle | Passed: All sides are different and satisfy the triangle inequality theorem. | Valid - Scalene |
| TC10 | -1, 2, 2 | Not a Triangle | Not a Triangle | Passed: Negative side is invalid. | Invalid - Negative |
| TC11 | 0, 1, 1 | Not a Triangle | Not a Triangle | Passed: Zero side is invalid. | Invalid - Zero |
| TC12 | 2, 2, 4 | Not a Triangle | Not a Triangle | Passed: Sum of two sides equals the third. | Invalid - Boundary |
| TC13 | 1, 1, 2 | Not a Triangle | Not a Triangle | Passed: Sum of two sides equals the third. | Valid - Not a Triangle |

pg. 9

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

# PRACTICAL NO.2: Implementation of White Box Testing.

Data Flow Analysis, Control Flow Analysis, Cyclomatic Complexity

# Data Flow Analysis

s+Title: To implement Data Flow Testing Problem statement :

S0: void quadratic\_ eq (float a, b, c , Boolean Is\_Comp) S1: { float discrim = b\*b-4\*a\*c

S2 : float r1, r2; S3 : {

S4 : If discrim < 0.0 S5 : Is\_Comp = true; S6: else

S7 : Is\_Comp = false; S8 : EndIf;

S9 : If not : Is\_Comp

S10 : r1=(-b + Sqrt (discrim ) ) / ( 2.0 \* a ) ; S11 : r2=(-b + Sqrt (discrim ) ) / ( 2.0 \* a ) ; S12 : EndIf;

S13 : End quadratic\_ eq } S14 : }

For the above function, draw the following tables and write the test cases:

* 1. Occurrence of variables and their categories.
  2. du pairs and their type
  3. All c-uses
  4. All p-uses

pg. 10

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

Theory:

Data flow testing is a technique based on identifying the paths in the program through program's control flow to explore the sequences of events. This is structural testing.

This testing strategy is designed in a way in which we can categorize each and every variable as per their occurrences in the code. Depending upon the status (such as define, created , use , etc ) of the variable , categorize are assigned to the particular variable or simply it disclose the proper and improper use of data very efficiently.

The main focus of Data flow testing is to execute sub paths of the program from the point where each variable is defined to the point where it has been referenced for any use of that variable.

Data Anomalies :

* d - It is referred as initialized , defined, created

It represents occurrences of a variable who has given a new value

such as input given by the user, assignment , input provided through file , etc.

* u - It is referred as used variables i.e. variable has not given any new value. There are two ways to show used variables

C- Computational use (C-use) : It includes, variable occurrences in the right hand side of an assignment statement or an output statement.

P - Predicate use (P-use) : this variable occurs in the predicate portion of a decision statement.

such as if ... else, while ... do , etc.

du path (define ==> use) : This represents a sub path from variable definition to use .

As mentioned above with the help of different categorize of variables ,Data flow testing technique is help us to know about the following things:

1. To identify the variables which are declared but not used within the program
2. To identify the variables which are used but never declared
3. To identify the variables which are declared several times
4. To identify the variables which are deallocated before it is used Based on this, test conditions are designed on four groups of coverage :
   * All definition

pg. 11

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

* + All c - use variables
  + All p - use variables
  + All du-paths

To test a given code , follow below given steps :

1. Initially give numbering to the program.
2. list out the variables
3. list out the occurrences of each and every variable and after listing assign a category to each variable.
4. Then identify use i.e. p-use and c -use variables and du-path too.
5. Based on this define the test cases depend on the coverage.

Solution:

Based on the given pseudo code in problem statement implement above mentioned steps to perform Data Flow Testing.

Step 1: Initially give numbering to the program.

0: void quadratic\_ eq (float a, b, c , Boolean Is\_Comp) 1: { float discrim = b\*b-4\*a\*c

2 : float r1, r2; 3 : {

4 : If discrim < 0.0 5 : Is\_Comp = true; 6: else

1. : Is\_Comp = false;
2. : EndIf;
3. : If not : Is\_Comp

10 : r1=(-b + Sqrt (discrim ) ) / ( 2.0 \* a ) ;

11 : r2=(-b + Sqrt (discrim ) ) / ( 2.0 \* a ) ; 12 : EndIf;

pg. 12

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

1. : End quadratic\_ eq } 14 : }

step 2: list out the variables

* + a, b, c
  + discrim
  + Is\_Comp
  + r1, r2

step 3: list out the occurrences of each and every variable and after listing assign a category to each variable.

As shown below table shows variable categories,

for ex. variable a, b, c are defined in line no 0 and they are used as computation ( c-use) variable at line no 1. same way at line no 1 , new value is assigned to variable discrim and at line no 4 , it is used as p-use variable.

|  |  |  |  |
| --- | --- | --- | --- |
| Statement | Variable Category | | |
| Number Variable | | c-use | p-use |
|  | Definition | variables | variables |
| 0 | a , b, c |  |  |
| 1 | discrim | a , b, c |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  | discrim |
| 5 | Is\_Comp |  |  |
| 6 |  |  |  |
| 7 | Is\_Comp |  |  |

pg. 13

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

|  |  |  |  |
| --- | --- | --- | --- |
| 8 |  |  |  |
| 9 |  |  | Is\_Comp |
| 10 | r1 | a,b, discrim |  |
| 11 | r2 | a,b, discrim |  |
| 12 |  |  |  |
| 13 |  |  |  |
| 14 |  |  |  |

Figure 15: Table showing various categories

step 4: Then identify use i.e. p-use and c -use variables and du-path too.

This table represents the pair of each variables define and use pair. such as variable discrim is as p-use variable at line no 4 but it is of c-use variable at line no 10 and 11.

|  |  |  |
| --- | --- | --- |
| du-pair (definition-use pair) | Variable | |
| Start line --> end line | c-use | p-use |
| 0 --> 1 | a,b,c |  |
| 0 -->10 | a,b |  |
| 0 --> 11 | a,b |  |
| 1 --> 4 |  | discrim |
| 1 --> 10 | discrim |  |
| 1 -->11 | discrim |  |
| 5 --> 9 |  | Is\_Comp |
| 7 --> 9 |  | Is\_Comp |
| 10 --> 14 | r1 |  |
| 11 --> 14 | r2 |  |

step 5 : Based on this , define the test cases depend on the coverage.

pg. 14

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

dc-use pairs

variables

Sub\_ path in the code

Test case

Inputs

a b c

Is\_Comp r1 r2

0 --> 1

a,b,c

0-1

TC1 1 1 1 T - -

0 --> 10

a,b

0-1-2-3-4-6-7-8-

9-10

TC2 1 2 1 F

-1 -1

0 -->11

a,b

0-1-2-3-4-6-7-8-

9-10-11

TC3 1 2 1 F

-1 -1

1 -->10

discrim

1-2-3-4-6-7-8-9-

10

TC4 1 2 1 F

-1 -1

1--> 11

discrim

1-2-3-4-6-7-8-9-

10-11

TC5 1 2 1 F

-1 -1

5 --> 14

Is\_comp

5-8-9-1-13-14

TC6 1 1 1 T - -

7 –14

Is\_comp

7-8-9-10-11-12-

13-14

TC7 1 2 1 F

-1 -1

10 -->

1. r1

10-11-12-13-14

TC8 1 2 1 F

-1 -1

11 -->

14 r2

11-12-13-14

TC9 1 2 1 F

-1 -1

dp- use pairs

variables

Sub\_ path in the code

Test case

Inputs

Is\_comp r1 r2

1-(4-

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | a | b | c |  | | |
| discrim | 1-2-3-4-5 | TC10 | 1 | 1 | 1 | T | - | - |
| discrim | 1-2-3-4-6 | TC11 | 1 | 2 | 1 | F | -1 | -1 |
| Is\_complex | 5-8-9-12 | TC12 | 1 | 1 | 1 | T | - | - |
| Is\_complex | 7-8-9-10 | TC13 | 1 | 2 | 1 | F | -1 | -1 |

5)

1-(4-

6)

5-(9-

12)

7-(9-

10)

pg. 15

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

Conclusion :

This structural based Data Flow testing technique gives the better scenario of variable categories and their different use with various sub path which helps to the tester to validate the proper and improper use of data.

# Control Flow Graph and Cyclomatic Complexity

Check the following code fragment for calculating cyclomatic Complexity: insertion\_procedure (int a[], int p [], int N)

{

int i,j,k;

for (i=0; i<=N; i++) p[i] = i; for (i=2; i<=N; i++)

{

k = p[i]; j = 1;

while (a[p[j-1]] > a[k]) {p[j] = p[j-1]; j--} p[j] = k;

}

}

# Control flow graph:

Control flow testing represents one of the important structural testing technique of testing. this technique uses control structure of a program or module to develop the test cases of the application. and this control structure can be represented by control flow graph of the program

The control flow graph (G) =(N,E) of an application which consist of a set of nodes (N) and set of edges (E)

Here cyclomatic complexity V(G) is defined for each module of an application to be e-n +2 where e = number of edges,

n= number of nodes ,in the control flow graph

pg. 16

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

Cyclomatic complexity also known as V(G) , where V refers to the cyclomatic number in graph theory and "G" indicates complexity is a function of the graph.

"Cyclomatic referred as the number of basic cycles which are connected through undirected graph which also provides number of independent paths through strongly connected graphs.

This strongly connected graphs is one in which each node can be reached from any other node

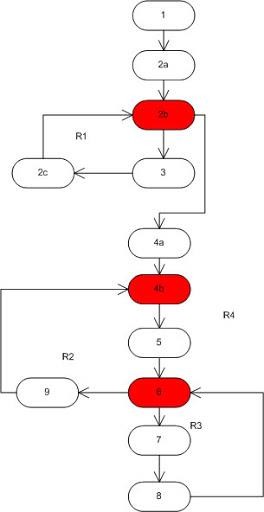
, when the virtual edge is connecting the exit node to entry node . Finally the cyclomatic complexity for specified program control flow graph is elaborate the cyclomatic number formula by adding one to represent the presence of virtual edge. This defines the cyclomatic complexity equal to the number of independent paths through the standard control flow graph model.

* first and foremost start numbering the statement insertion\_procedure (int a[], int p [], int N)

|  |  |  |
| --- | --- | --- |
| { |  | |
| **(1)** |  | Int i,j,k; |
| **(2)** |  | for (**(2a)**i=0; **(2b)**i<=N; **(2c)**i++) |
| **(3)** |  | p[i] = i; |
| **(4)** |  | for (**(4a)**i=2; **(4b)**i<=N; **(4c)**i++) |
|  | { |  |
| **(5)** |  | k=p[i];j=1; |
| **(6)** |  | while (a[p[j-1]] > a[k]) { |
| **(7)** |  | p[j] = p[j-1]; |
| **(8)** |  | j-- |
|  |  | } |
| **(9)** |  | p[j] = k; |
|  | } |  |

pg. 17

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB



Now, to calculate cyclomatic complexity : use one of three methods

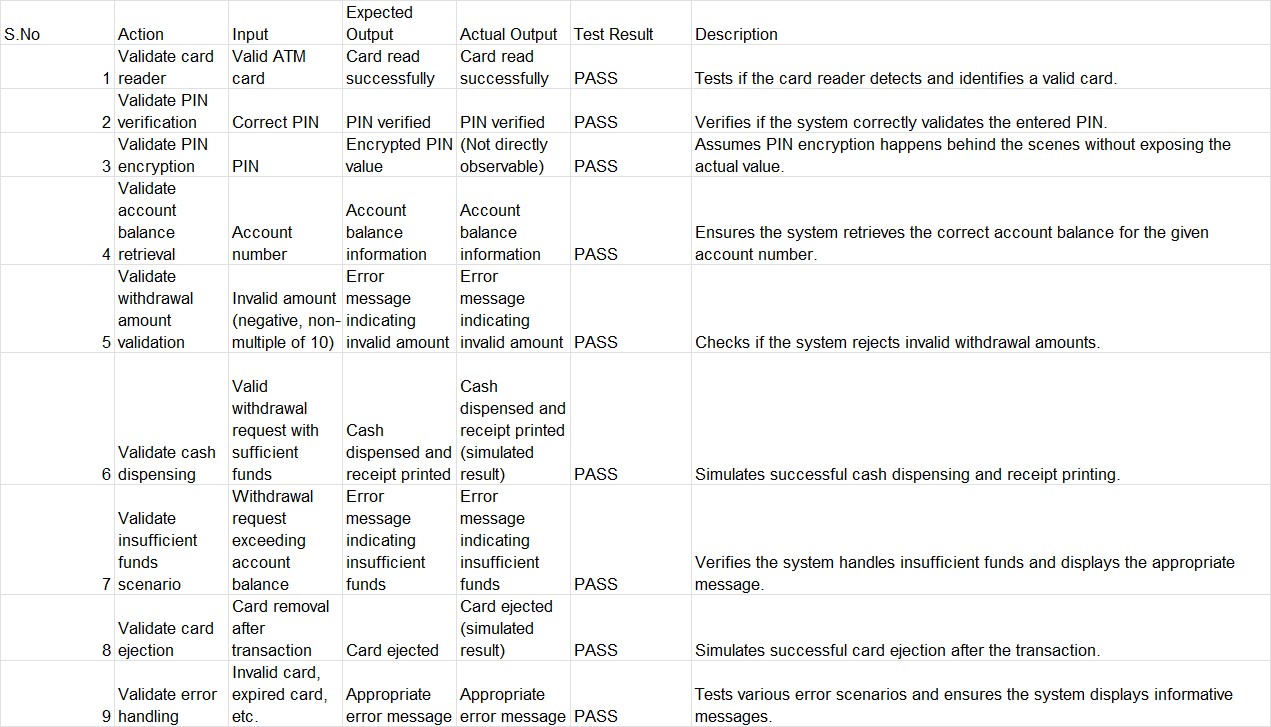
1. Count the number of regions on the graph: 4
2. No. of predicates (red on graph) + 1 : 3 + 1 = 4
3. No of edges – no. of nodes + 2: 14 – 12 + 2 = 4

pg. 18

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

**PRACTICAL NO.3:** Prepare test cases for below given applications using Boundary Value Analysis and Equivalence Class Partitioning:

1. Prepare test cases on ATM Machin using Unit and System Testing Unit Testing



System Testing:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S.  No | Action | Scenario | Expected Result | Actual Result | Test Result | Description |
| 10 | Login and withdrawal | User inserts valid card, enters correct PIN, selects withdrawal, and enters a valid amount. | Successful withdrawal and receipt printed | Successful withdrawal and receipt printed | PASS | Tests the complete withdrawal flow with valid inputs. |
| 11 | Login and invalid PIN | User inserts valid card, enters incorrect PIN, and retries. | Card blocked after exceeding PIN attempts | Card blocked after exceeding PIN attempts | PASS | Simulates incorrect PIN attempts and card blocking behavior. |

pg. 19

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | Login and insufficient funds | User inserts valid card, enters correct PIN, selects withdrawal, and enters an amount exceeding balance. | Insufficient funds message displayed | Insufficient funds message displayed | PASS | Checks withdrawal behavior with insufficient funds. |
| 13 | Network connectivity | User attempts a transaction during a network outage. | Error message indicating network unavailable | Error message indicating network unavailable | PASS | Tests system behavior during network issues. |
| 14 | Cash dispenser malfunction | User attempts a withdrawal, but the cash dispenser fails. | Error message indicating dispenser issue | Error message indicating dispenser issue | PASS | Simulates cash dispenser malfunction and error handling. |
| 15 | Login and cancel transaction | User inserts valid card, enters correct PIN, selects withdrawal, but cancels the transaction before entering amount. | Transaction canceled and card ejected | Transaction canceled and card ejected | PASS | Tests cancellation of a withdrawal transaction. |
| 16 | Card reader malfunction | User inserts damaged or unreadable card. | Error message indicating card reading failure | No response from machine | FAIL | Checks system response to unreadable card. |
| 17 | PIN entry timeout | User inserts valid card but delays entering PIN  beyond | Transaction canceled and card ejected | Transaction timeout not triggered | FAIL | Tests timeout mechanism for delayed PIN entry. |

pg. 20

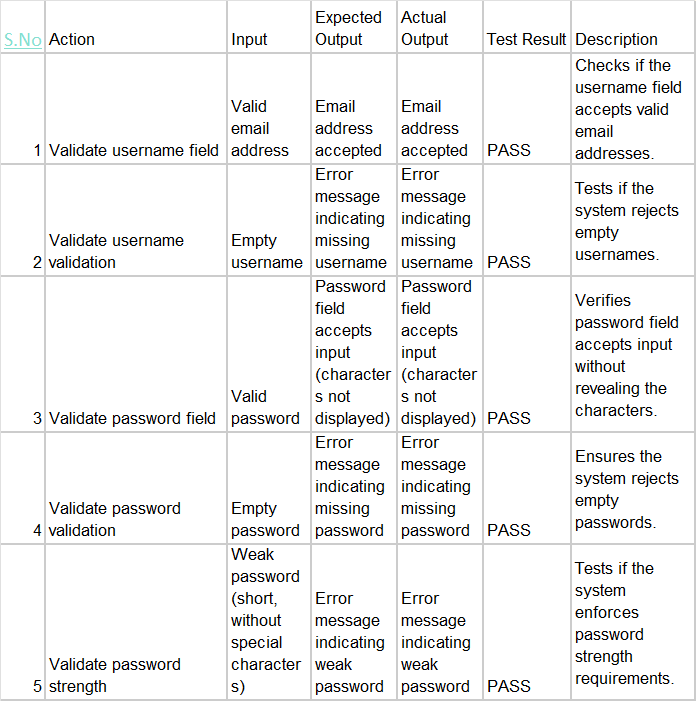
NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | allowed time limit. |  |  |  |  |
| 18 | Incorrect account selection | User inserts valid card, enters correct PIN, selects invalid account type (e.g., savings instead of checking). | Error message for invalid account type | Withdrawal processed from default account | FAIL | Checks system response to invalid account selection. |
| 19 | Double withdrawal attempt | User inserts valid card, enters correct PIN, performs withdrawal, and immediately attempts another withdrawal exceeding available balance. | Insufficient funds message displayed | Withdrawal allowed, overdraft triggered | FAIL | Tests account balance check after initial withdrawal. |
| 20 | Currency denomination issue | User requests a specific currency denomination that is unavailable in the ATM. | Message indicating unavailable denomination | Machine dispensed mixed denominations instead | FAIL | Tests system handling of unavailable currency denominations. |

pg. 21

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

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



pg. 22

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

System Testing:

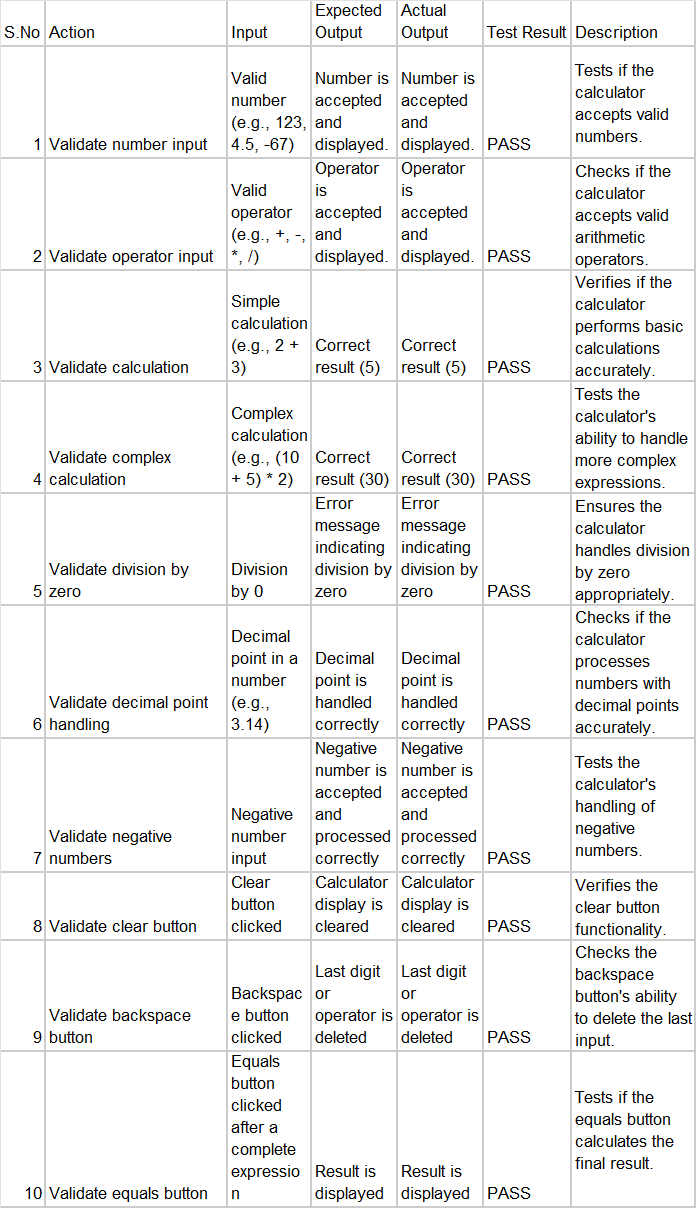


1. Prepare a test cases on Calculator using Unit and System Testing

pg. 23

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

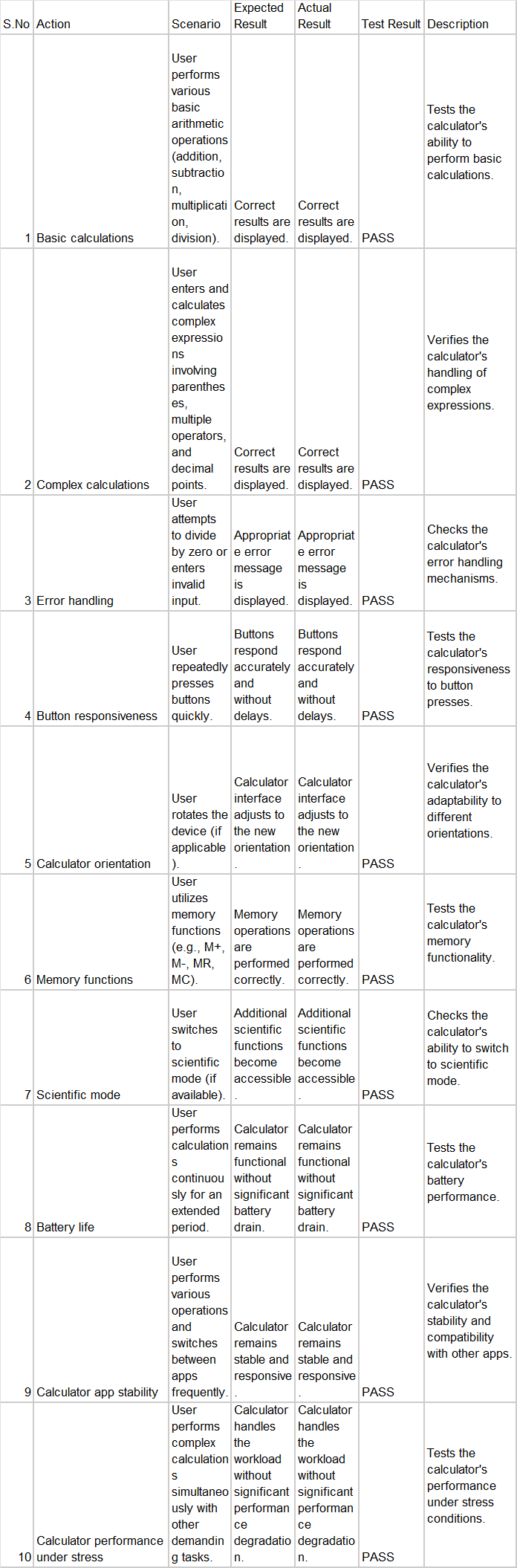
Unit Testing:



System Testing:

pg. 24

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB



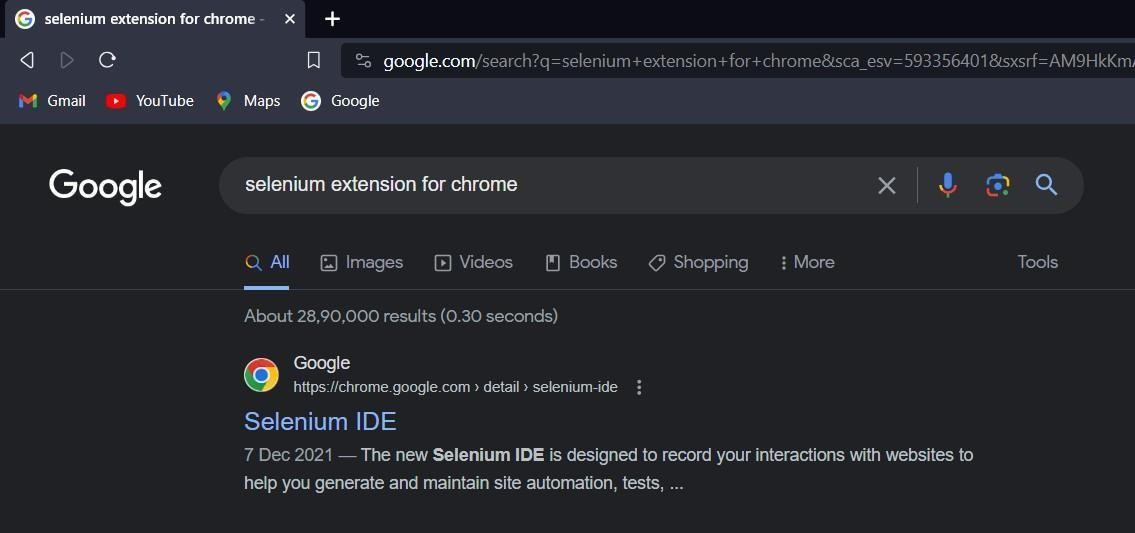
pg. 25

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

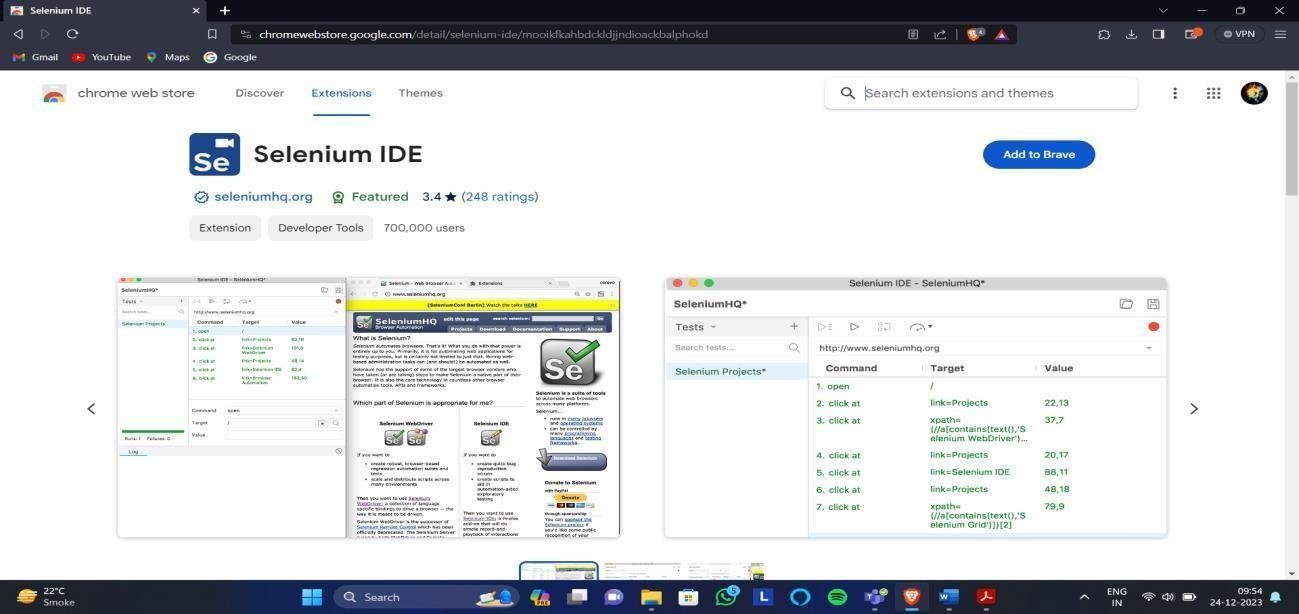
**PRACTICAL NO.4:** Introduction to Selenium

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

Step 1: Search on Google chrome for “selenium extension for chrome”



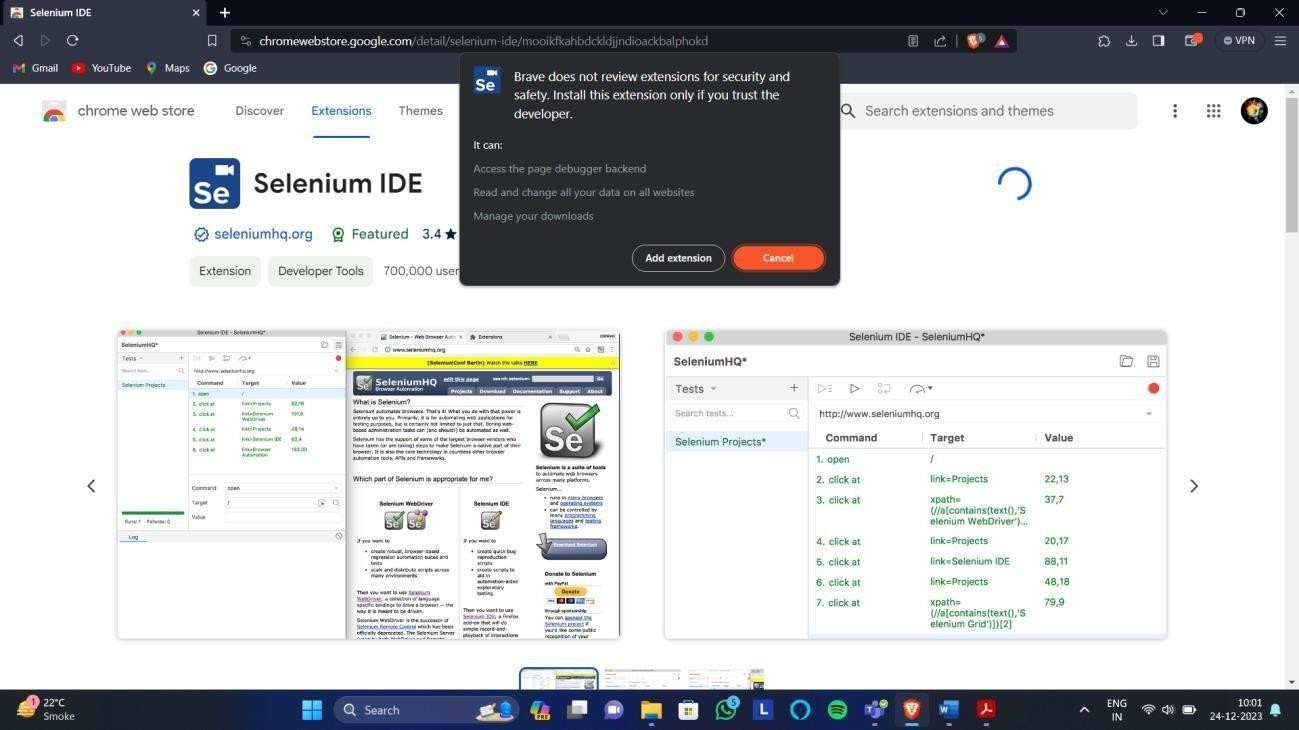
Step 2: click on add to chrome:



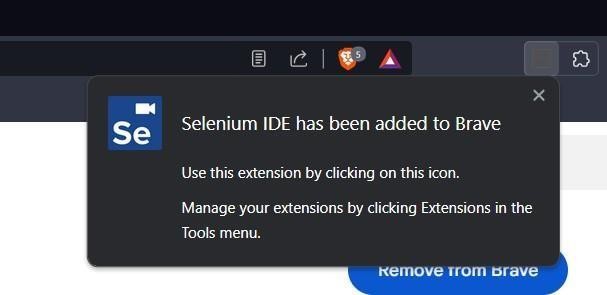
pg. 26

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

Step 3: Click on Add Extension



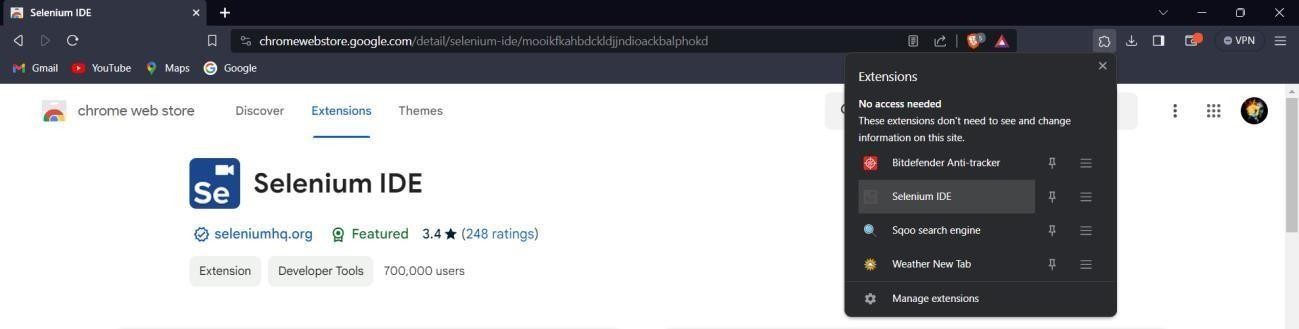
Step 4: Installation Done and the following popup will show up



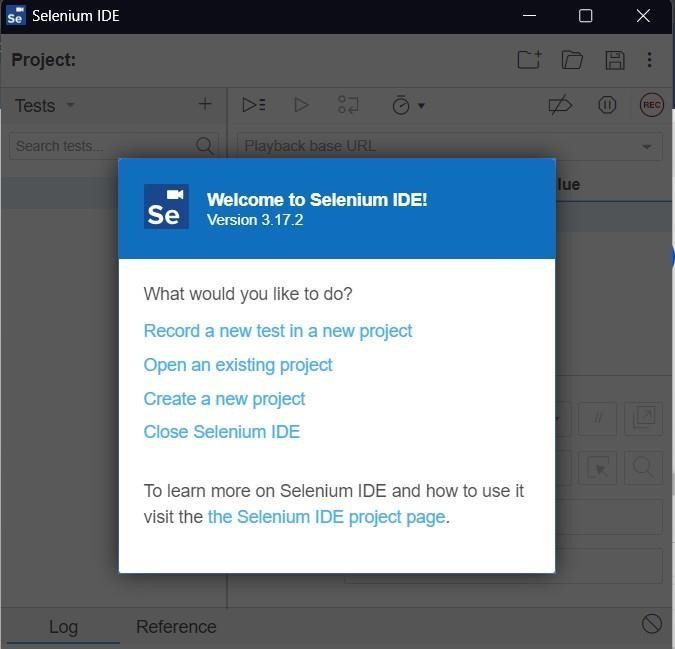
Step 5: To open Selenium in Chrome, Click the following

pg. 27

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB



Step 6: Selenium Window will open

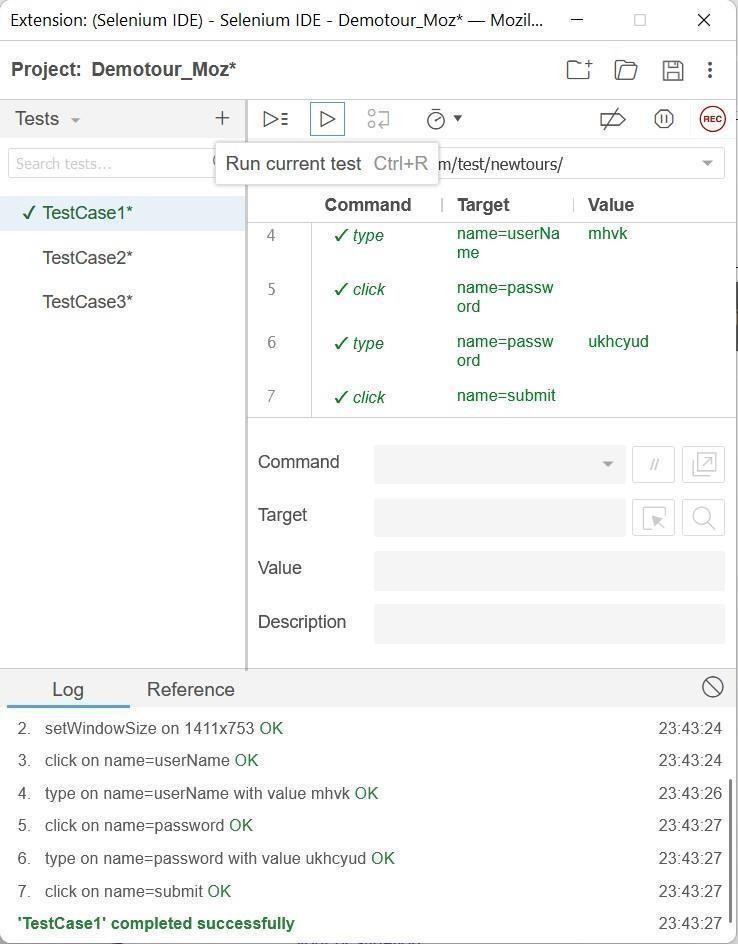


1. Record and run test cases on demo tour website for login page through Mozilla Firefox.

Case 1: Both username and password are incorrect

pg. 28

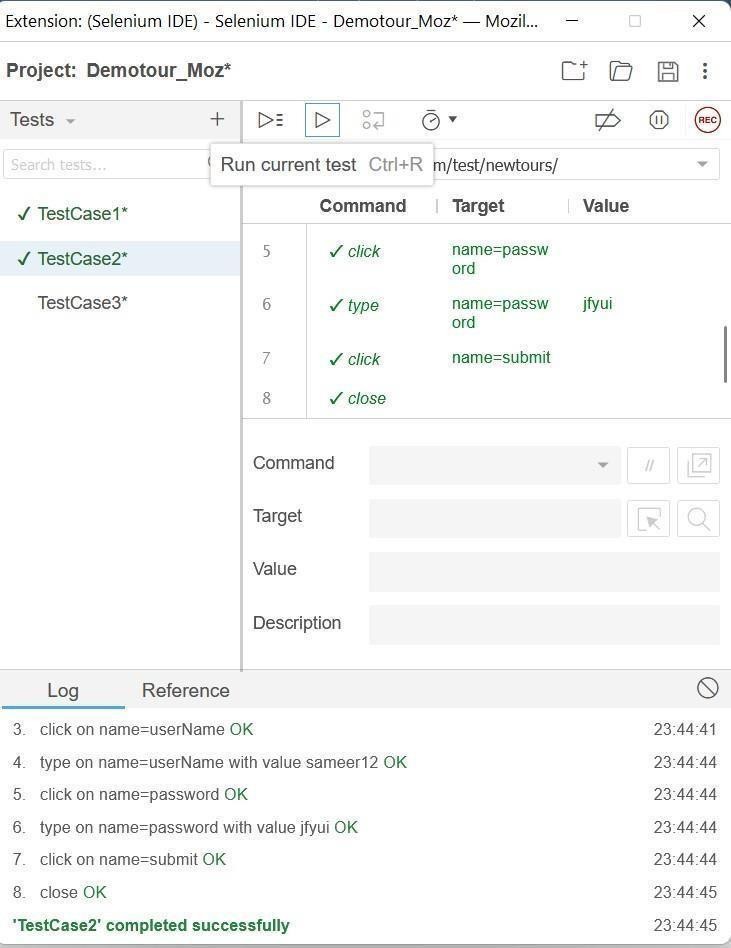
NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB



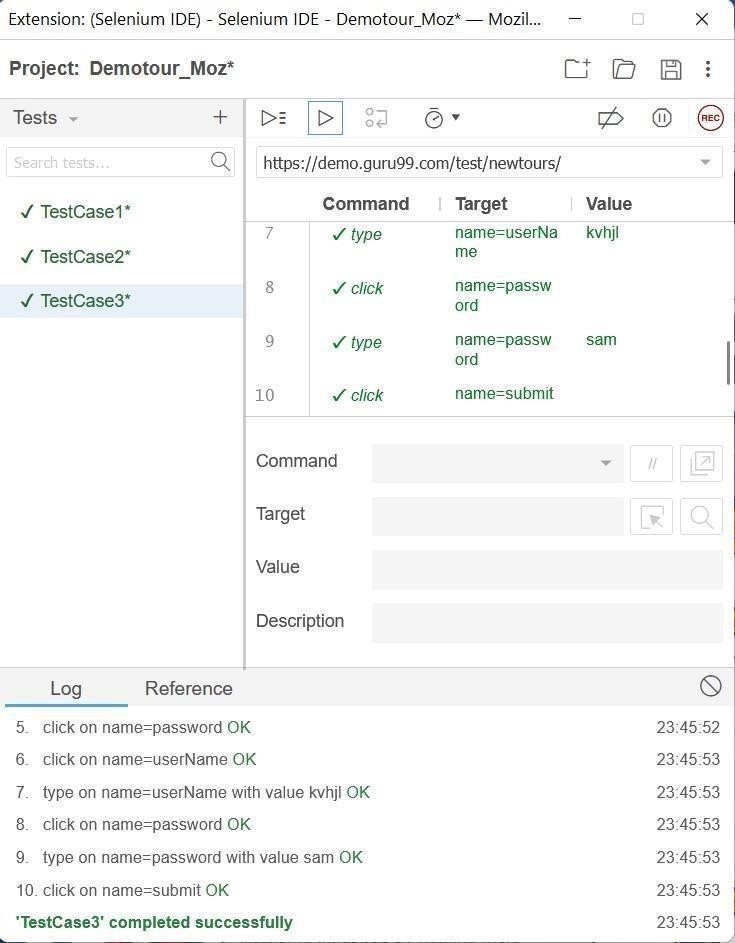
Case 2: Username is correct but password is incorrect

pg. 29

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB



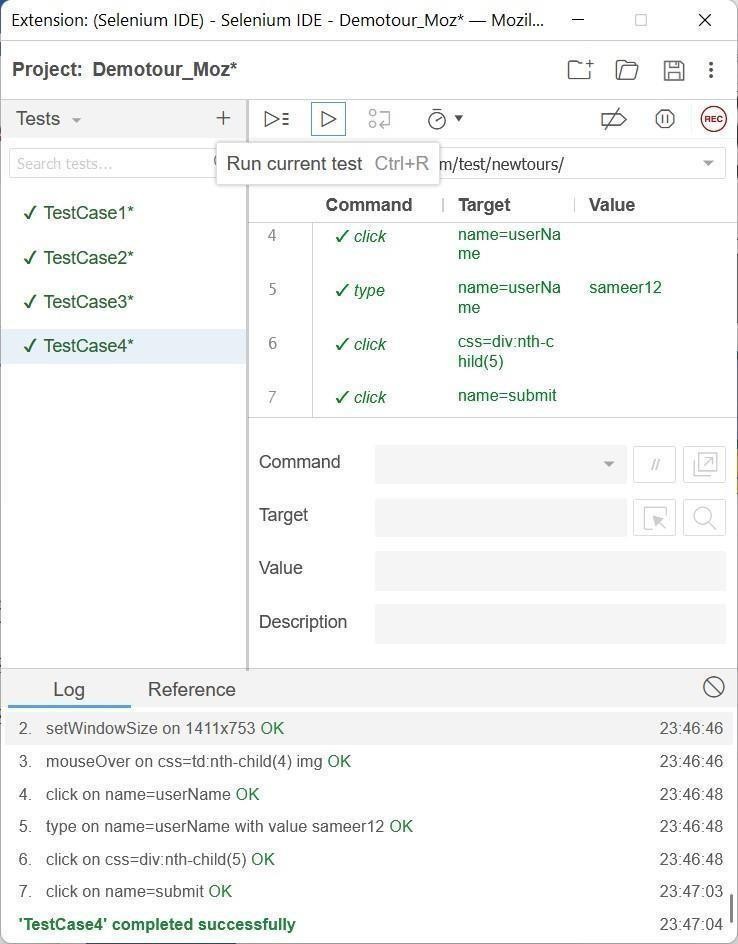
Case 3: Username is incorrect but password is correct



Case 4: Both Username and Password are correct

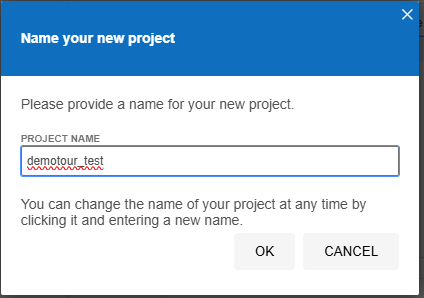
pg. 30

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB



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

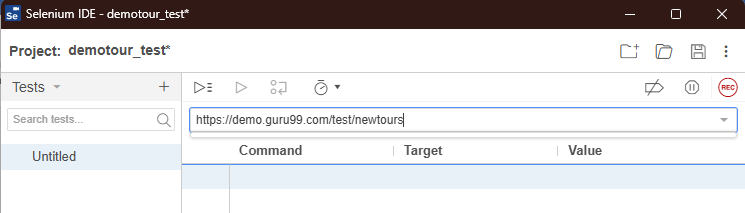
Step 1: Click on new Project and give the project name



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

pg. 31

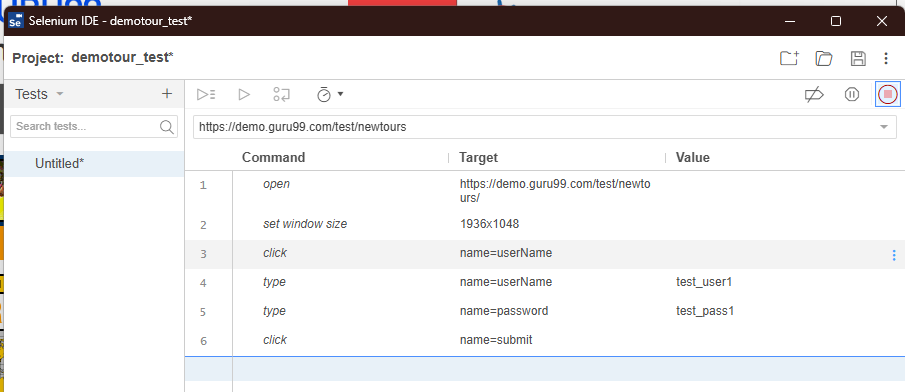
NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB



Step 3: New window will open maximise that window



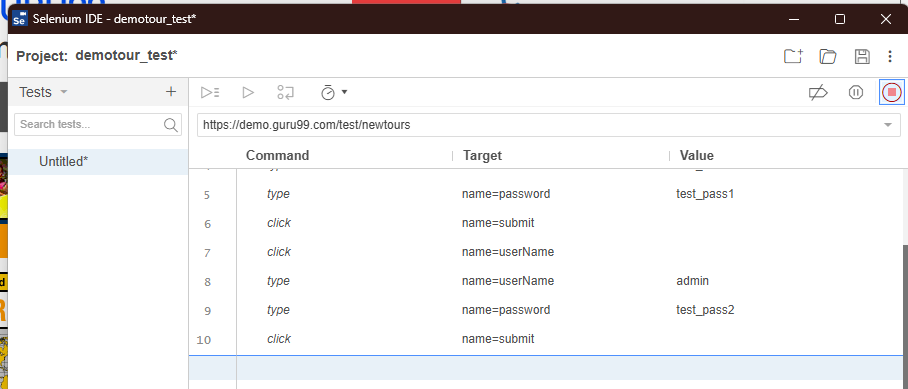
First Test Case: Both the username and password is incorrect



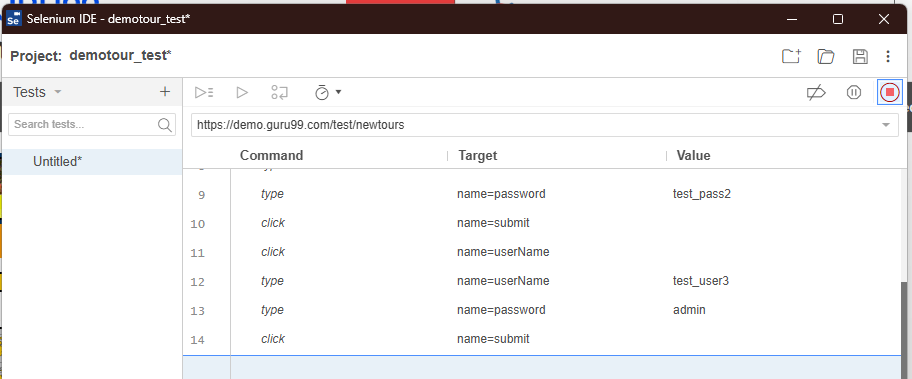
Second Test Case: Username is correct but Password is incorrect

pg. 32

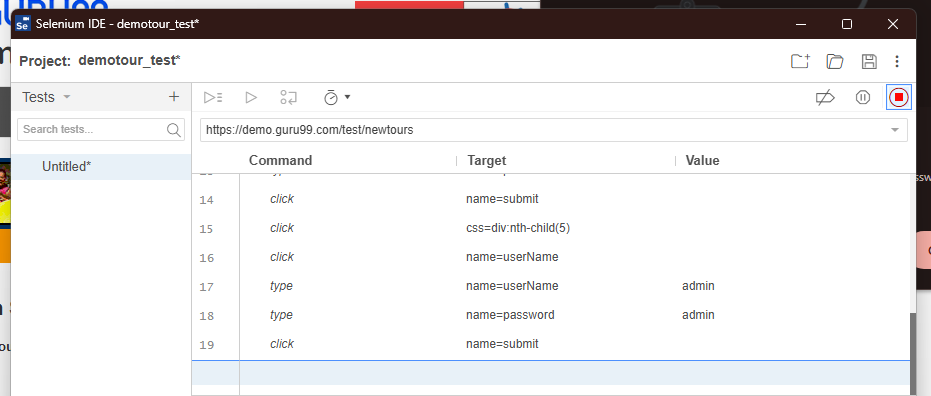
NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB



Third test Case: Username is incorrect but Password is correct



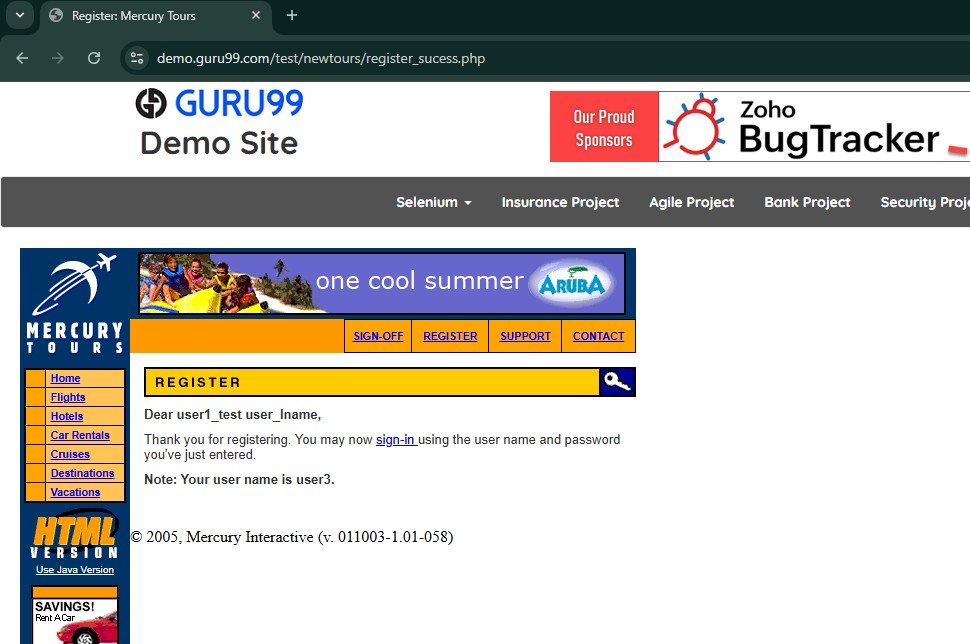
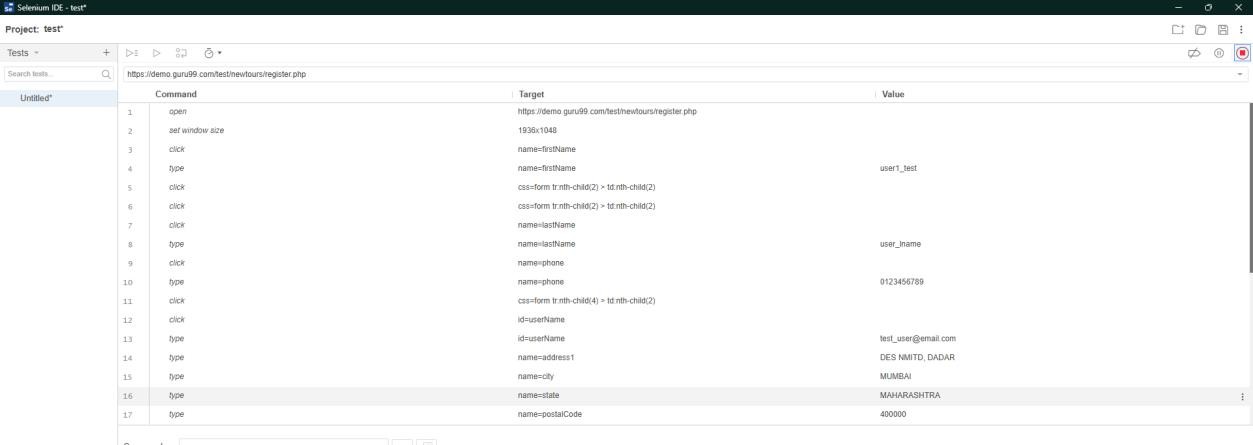
Fourth Test Case: Both Username and password is Correct



pg. 33

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

1. Record and run test cases on Registration form of any website

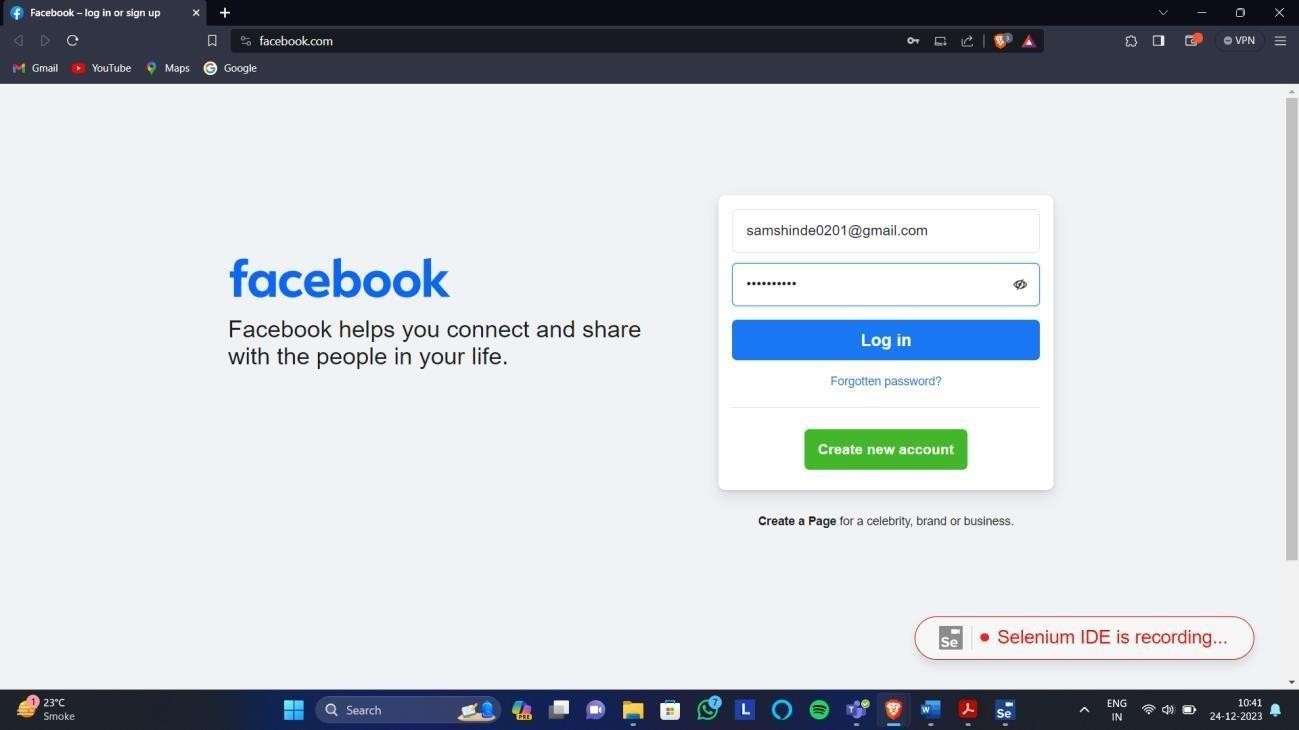


1. Record and run test cases on any website to check the validations of elements. (eg. IRCTC , MSRTC , etc )

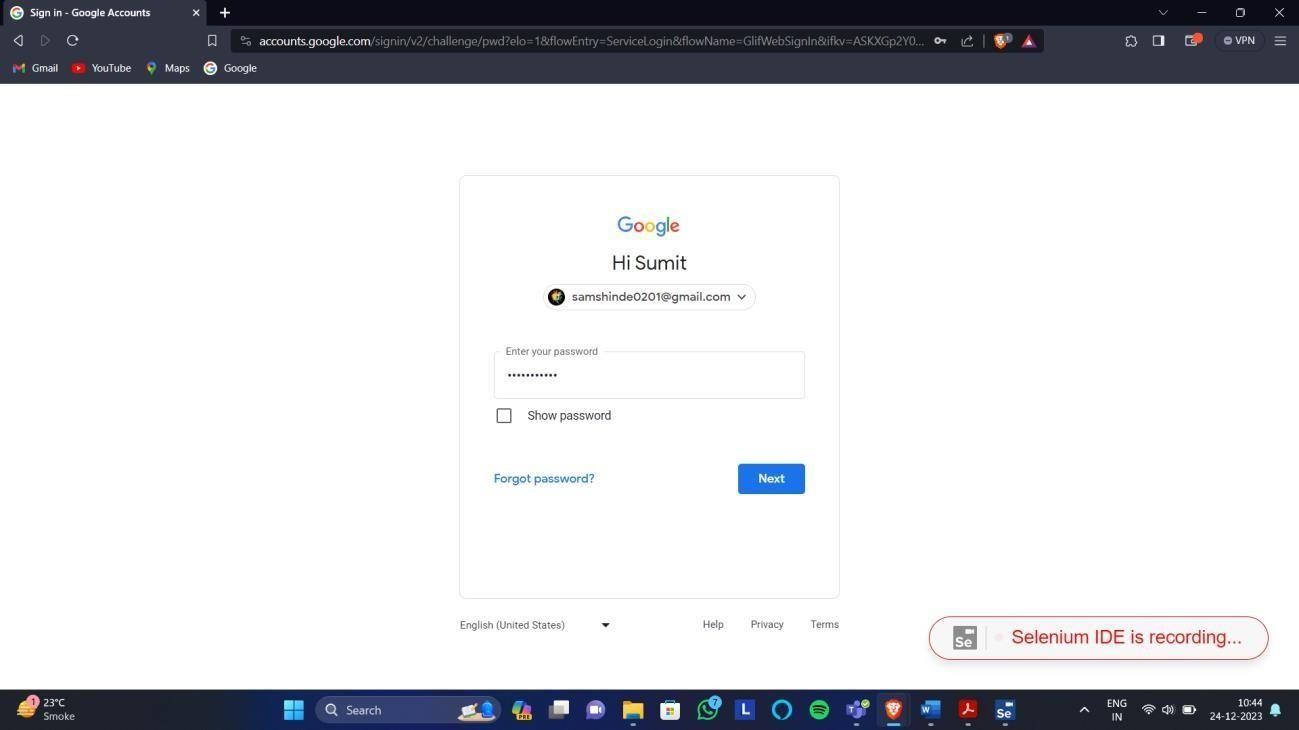
pg. 34

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

TestCase1: facebook Username and password incorrect



TestCase2: Gmail login



pg. 35

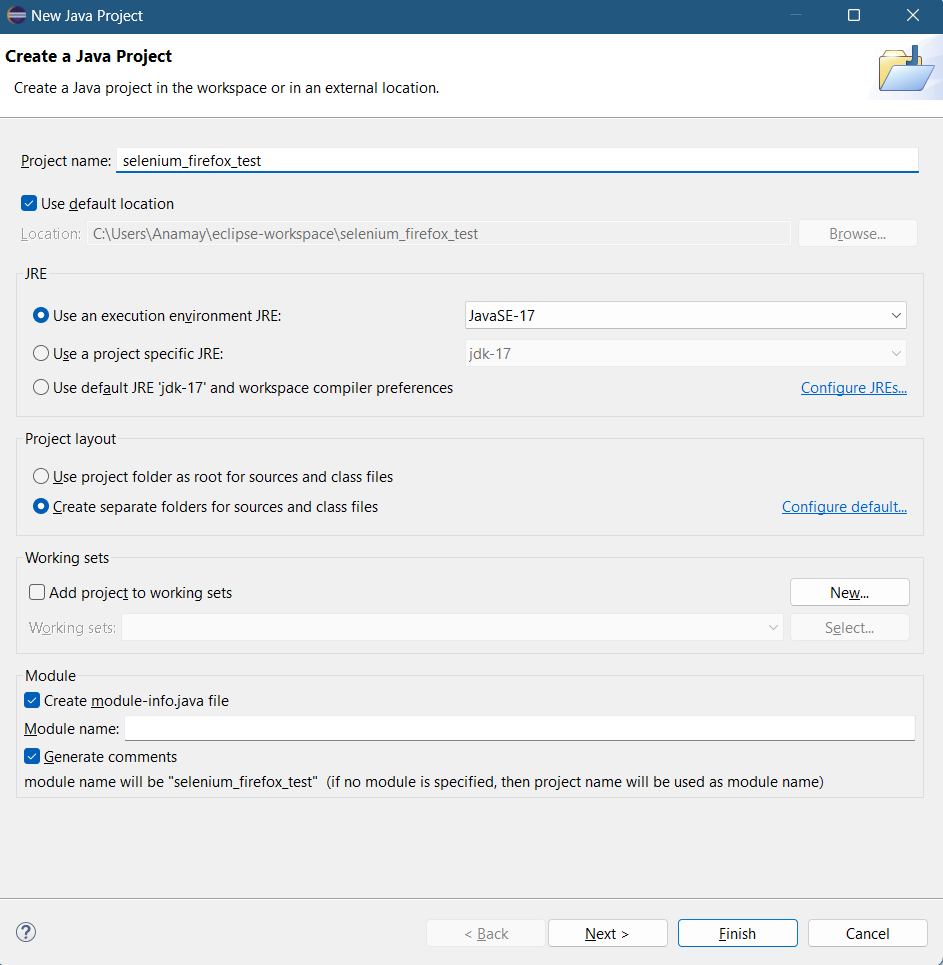
NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

**PRACTICAL NO.5:** Implement Web Drivers on Chrome & Firefox Browsers. Installation steps of Web Drivers on Firefox Browsers.

Step 1: Download the Selenium drivers ZIP file for Java from this link  [https://github.com/SeleniumHQ/selenium/releases/download/selenium-4.24.0/selenium-](https://github.com/SeleniumHQ/selenium/releases/download/selenium-4.24.0/selenium-java-4.24.0.zip) [java-4.24.0.zip](https://github.com/SeleniumHQ/selenium/releases/download/selenium-4.24.0/selenium-java-4.24.0.zip). Extract the contents of the ZIP file and save them in a known location.

Step 2: Download the GeckoDriver from this link  [https://sourceforge.net/projects/geckodriver.mirror/files/v0.35.0/geckodriver-v0.35.0-](https://sourceforge.net/projects/geckodriver.mirror/files/v0.35.0/geckodriver-v0.35.0-win64.zip/download) [win64.zip/download](https://sourceforge.net/projects/geckodriver.mirror/files/v0.35.0/geckodriver-v0.35.0-win64.zip/download). This file (geckodriver-v0.35.0-win64.zip) is for 64-bit Windows and might not be available on the official GeckoDriver GitHub page. Extract the downloaded ZIP file and save the GeckoDriver in a known location.

Step 3: Open Eclipse IDE for Java Developers (Version: 2022-09 (4.25.0) or similar). Create a new Java project in Eclipse.

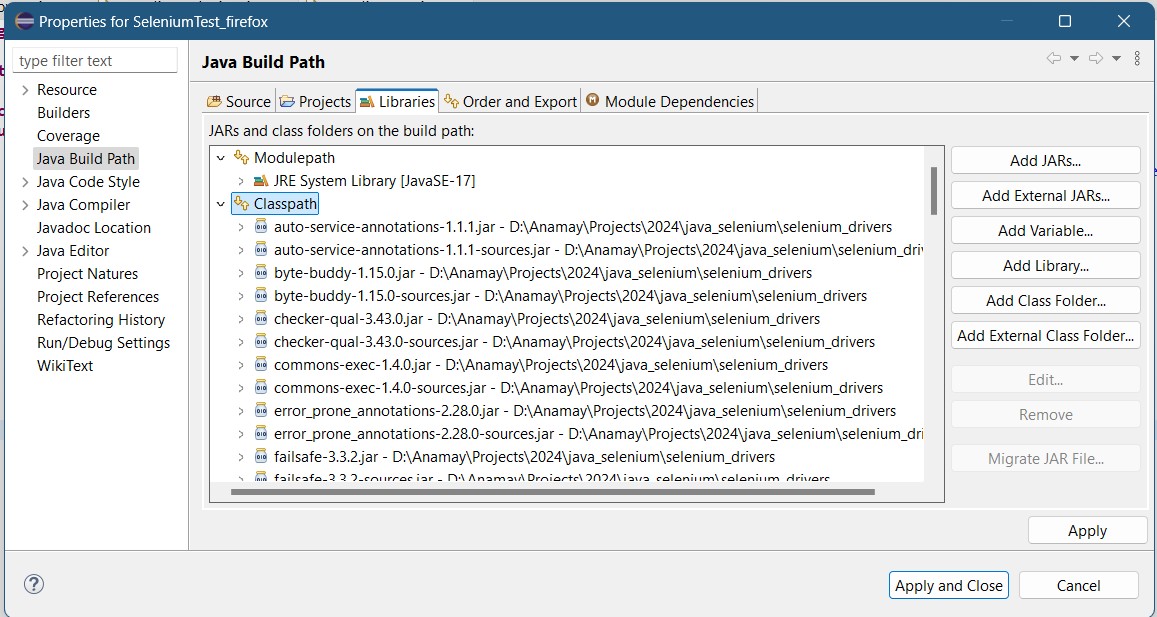


Step 4: Right-click on your project in the Package Explorer and select Build Path > Configure Build Path. In the Libraries tab, click on Classpath and then focus on the right-hand side.

pg. 36

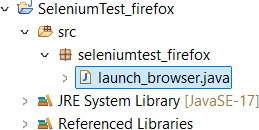
NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

Click Add External JARs and navigate to the folder where you extracted the Selenium files. Select all the JAR files. Click on Apply and Close. Note: Ensure that the JAR files are added under the Classpath section, not the Modulepath.



Step 5: If a module-info.java file is present in your project, delete it immediately. This file is not required for non-modular projects and can cause issues.

Step 6: Right-click on the src folder and create a new package named as seleniumtest\_firefox. Once the package is created, right-click on it and create a new Java class (name it as per your preference). Write your Selenium Java code in this class.



package seleniumtest\_firefox;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver; import org.openqa.selenium.firefox.FirefoxOptions;

public class launch\_browser {

public static void main(String[] args) {

// Set the path to geckodriver executable System.setProperty("webdriver.gecko.driver",

"D:\\Users\\Projects\\2024\\java\_selenium\\firefox\_gecko\_driver\\geckod river.exe");

// Set the path to Firefox binary

pg. 37

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

FirefoxOptions options = new FirefoxOptions(); options.setBinary("C:\\Program Files\\Mozilla

Firefox\\firefox.exe"); // Change this path to your Firefox installation path

// Create a new instance of the Firefox driver WebDriver driver = new FirefoxDriver(options);

// Launch the website driver.get("https://nmitd.edu.in/");

// Optionally, you can add a wait here try {

Thread.sleep(5000); // Wait for 5 seconds

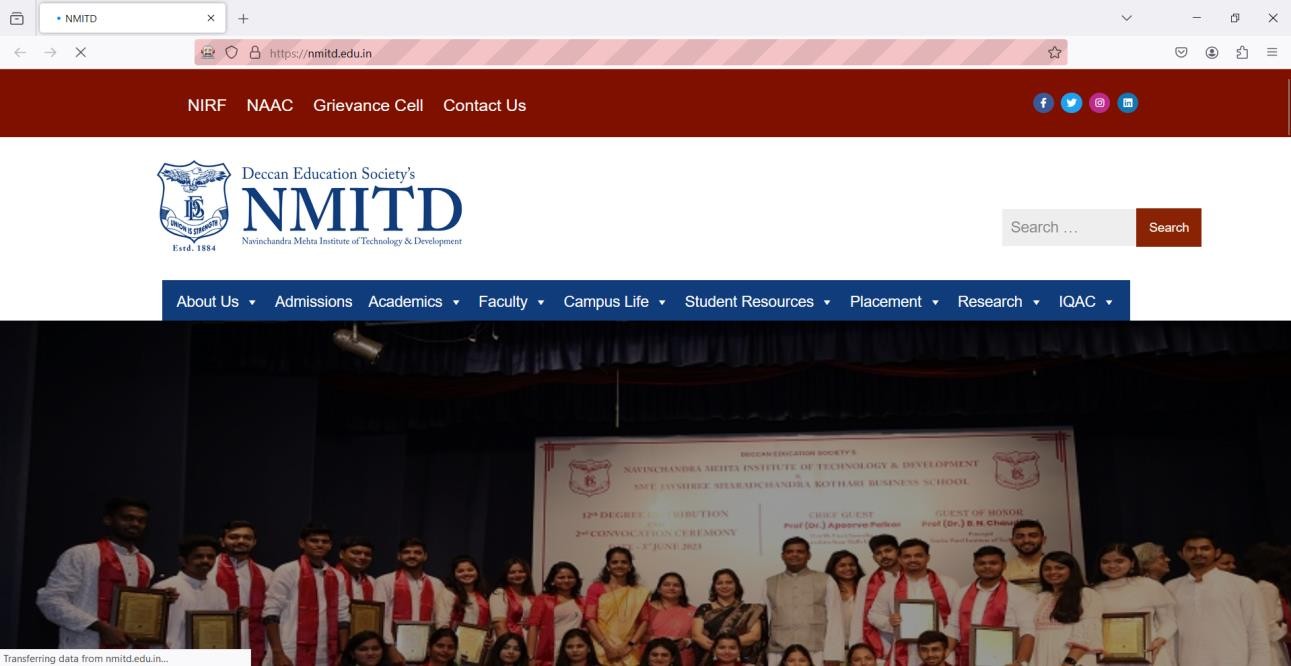
} catch (InterruptedException e) { e.printStackTrace();

}

// Close the browser driver.quit();

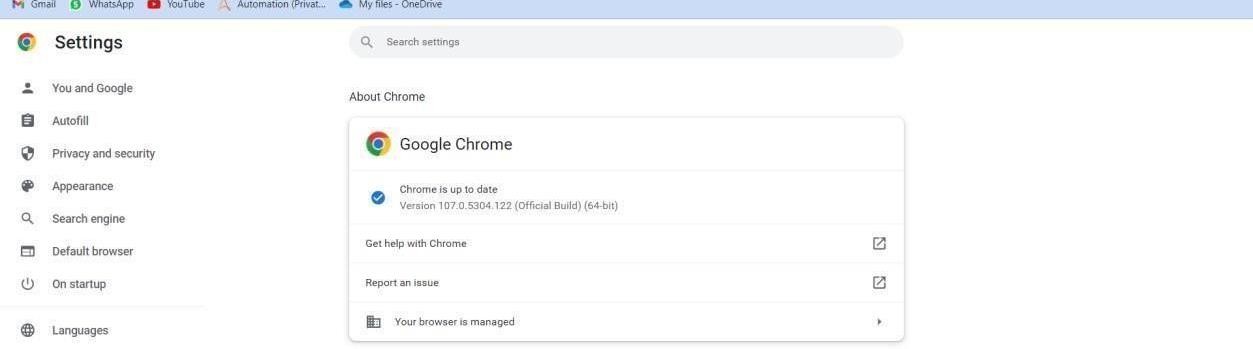
}

}



Installation steps of Web Drivers on Chrome Browsers.

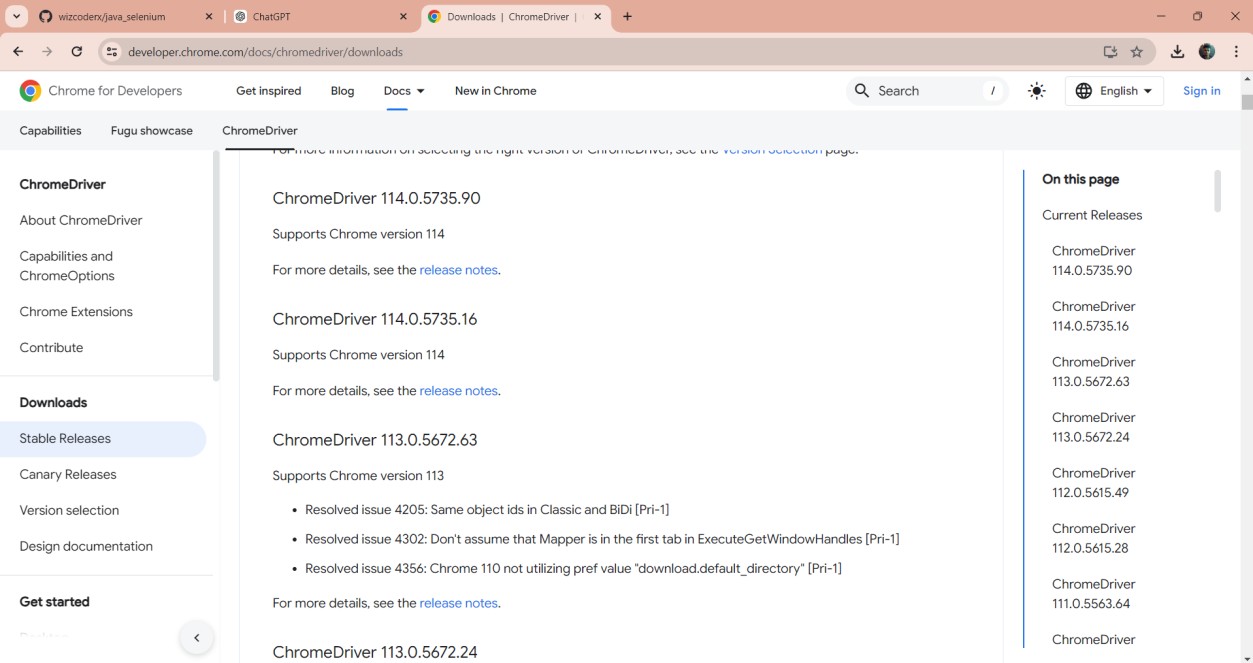
Step 1: Check the chrome version



pg. 38

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

Step 2: Search for chrome drivers for selenium



Repeat the Step 3 to Step 5 from the Firefox installation part and jump on the below step 6 part.

Step 6: Write the selenium java code to the class which you have created.

package com.selenium.test;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class SeleniumLaunch {

public static void main(String[] args) {

// Set the path to your ChromeDriver executable System.setProperty("webdriver.chrome.driver",

"D:\\User\\Projects\\2024\\java\_selenium\\chrome\_driver\\chromedriver.e xe");

// Create a new instance of the ChromeDriver WebDriver driver = new ChromeDriver();

// Launch the opposite driver.get("https://nmitd.edu.in/");

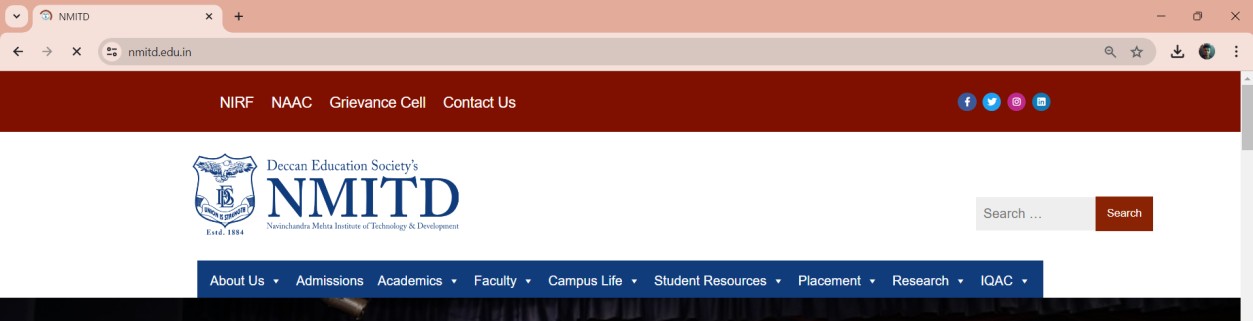
// Close the browser driver.quit();

}

}

pg. 39

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB



**PRACTICAL NO.6:** Demonstrate Handling multiple frames in Selenium

Step 1: Go the build path and delete the jar files named as ‘byte-byte’ under the classpath,

then click on apply and close. Step 2: Run the following code:

package seleniumtest\_firefox;

import org.openqa.selenium.By; import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.firefox.FirefoxDriver; import org.openqa.selenium.firefox.FirefoxOptions;

public class HandleFrames {

public static void main(String[] args) {

// Set up Firefox driver System.setProperty("webdriver.gecko.driver",

"D:\\Saiganesh\\Projects\\2024\\java\_selenium\\firefox\_gecko\_driver\\ge ckodriver.exe");

// Set Firefox binary path

FirefoxOptions options = new FirefoxOptions(); options.setBinary("C:\\Program Files\\Mozilla

Firefox\\firefox.exe");

WebDriver driver = new FirefoxDriver(options);

// Open the specified URL driver.get(["h](http://demo.guru99.com/test/guru99home/)t[tp://demo.guru99.com/test/guru99home/](http://demo.guru99.com/test/guru99home/)");

// Maximize the browser window driver.manage().window().maximize();

try {

// Switch to the iframe with name or ID "a077aa5e" driver.switchTo().frame("a077aa5e");

System.out.println("Switched to the iframe with ID 'a077aa5e'");

// Locate and click the image inside the iframe WebElement iframeElement =

driver.findElement(By.xpath("/html/body/a/img")); iframeElement.click(); // Perform the click action System.out.println("\*We are done\*");

pg. 40

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

// Optionally, switch back to main content if further interactions are needed

driver.switchTo().defaultContent();

} catch (Exception e) { e.printStackTrace();

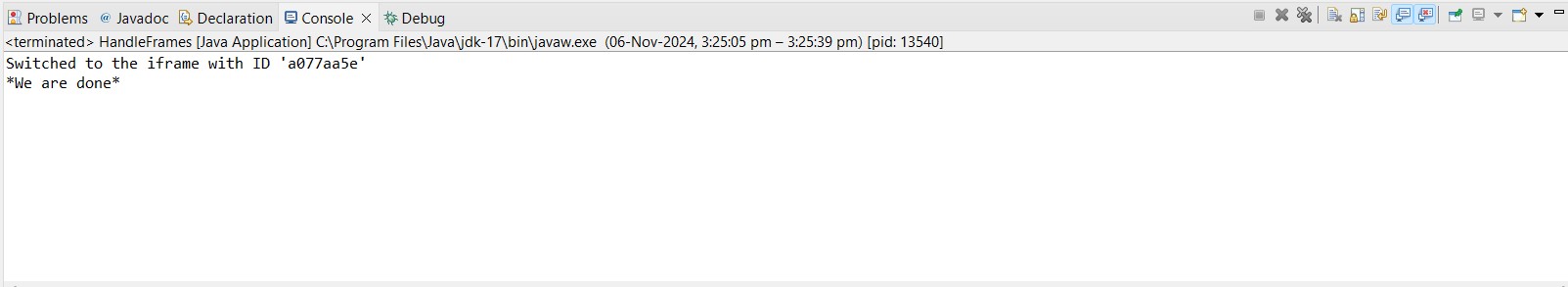
} finally {

// Close the browser driver.quit();

}

}

}



pg. 41

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

**PRACTICAL NO.7:** Implement browser commands and navigation commands Step 1: Write the below code

package seleniumtest\_firefox;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver; import org.openqa.selenium.firefox.FirefoxOptions;

public class BrowserNavigationCommands { public static void main(String[] args) {

// Set up Firefox driver System.setProperty("webdriver.gecko.driver",

"D:\\Saiganesh\\Projects\\2024\\java\_selenium\\firefox\_gecko\_driver\\ge ckodriver.exe");

// Set Firefox binary path

FirefoxOptions options = new FirefoxOptions(); options.setBinary("C:\\Program Files\\Mozilla

Firefox\\firefox.exe");

WebDriver driver = new FirefoxDriver(options);

// Maximize the browser window driver.manage().window().maximize();

try {

// Open Google driver.get("https://[www.google.com"](http://www.google.com/)); System.out.println("Opened Google homepage");

// Navigate to another website, for example, OpenAI driver.navigate().to("https:/[/www.openai.co](http://www.openai.com/)m"); System.out.println("Navigated to OpenAI");

// Go back to Google driver.navigate().back();

System.out.println("Navigated back to Google");

// Go forward to OpenAI driver.navigate().forward(); System.out.println("Navigated forward to OpenAI");

// Refresh the page driver.navigate().refresh(); System.out.println("Page refreshed");

} catch (Exception e) { e.printStackTrace();

} finally {

// Close the browser driver.quit();

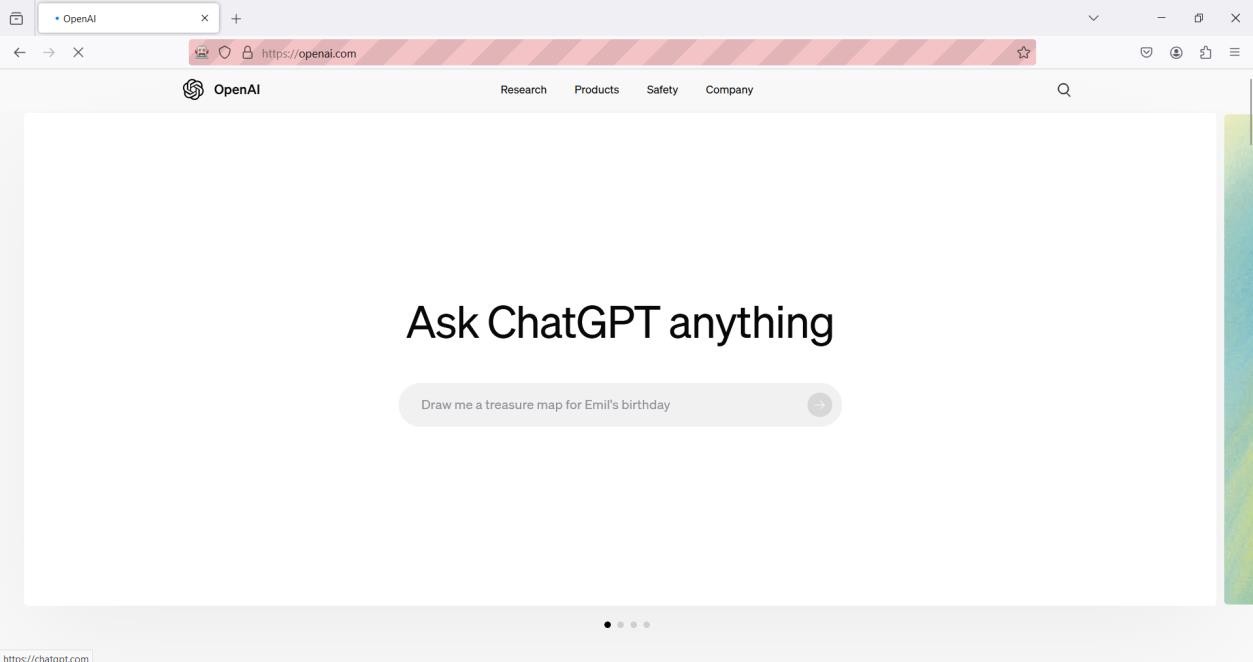
}

}

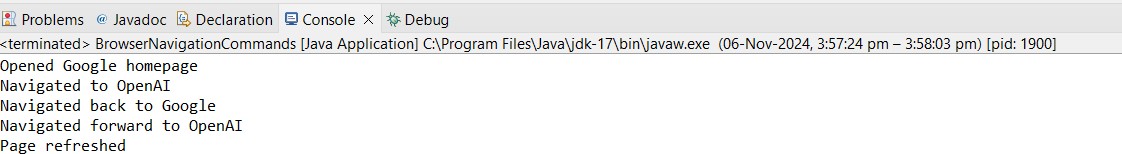
}

pg. 42

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB







pg. 43

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

**PRACTICAL NO.8:** Implement the find element command Step 1: Write the following code

package seleniumtest\_firefox;

import org.openqa.selenium.By; import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.firefox.FirefoxDriver; import org.openqa.selenium.firefox.FirefoxOptions;

public class FacebookLoginAutomation { public static void main(String[] args) {

// Set up Firefox driver System.setProperty("webdriver.gecko.driver",

"D:\\Saiganesh\\Projects\\2024\\java\_selenium\\firefox\_gecko\_driver\\ge ckodriver.exe");

// Set Firefox binary path

FirefoxOptions options = new FirefoxOptions(); options.setBinary("C:\\Program Files\\Mozilla

Firefox\\firefox.exe");

WebDriver driver = new FirefoxDriver(options);

// Maximize the browser window driver.manage().window().maximize();

try {

// Navigate to Facebook login page driver.get("https://[www.facebook.com/](http://www.facebook.com/)"); System.out.println("Facebook login page opened");

// Find the email input field by ID and enter the email WebElement emailField = driver.findElement(By.id("email")); emailField.sendKeys(["test\_us](mailto:test_user@gmail.com)e[r@gmail.com"](mailto:test_user@gmail.com)); System.out.println("Email entered");

// Find the password input field by ID and enter the

password

WebElement passwordField =

driver.findElement(By.id("pass"));

passwordField.sendKeys("test\_password\_12345"); System.out.println("Password entered");

// Find the login button by name and click it WebElement loginButton =

driver.findElement(By.name("login"));

loginButton.click(); System.out.println("Login button clicked");

// Optionally, wait for a few seconds to observe the login action (or use WebDriverWait for explicit waits)

try {

Thread.sleep(5000); // Wait for 5 seconds (replace with WebDriverWait in production)

pg. 44

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

} catch (InterruptedException e) { e.printStackTrace();

}

} catch (Exception e) { e.printStackTrace();

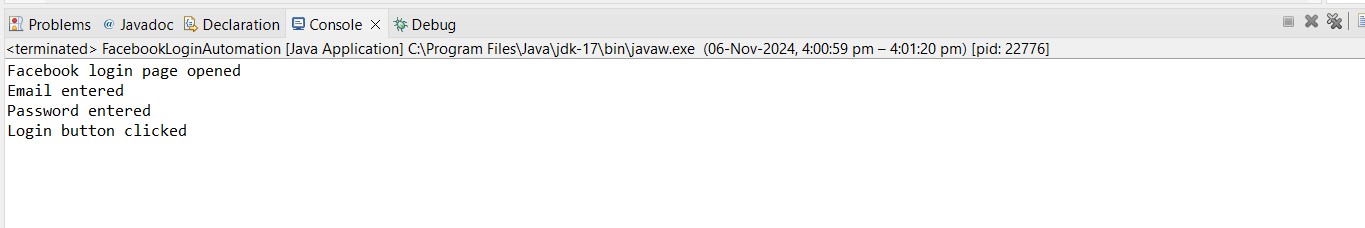
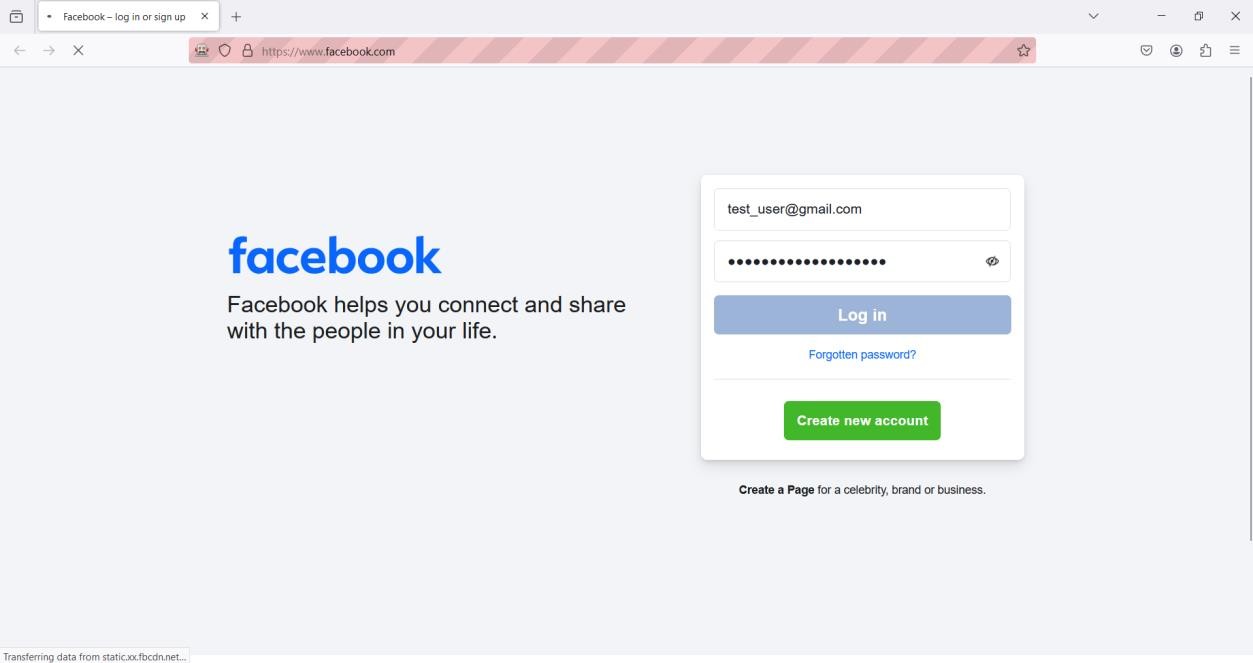
} finally {

// Close the browser driver.quit();

}

}

}



pg. 45

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

**PRACTICAL NO.9:** Demonstrate the Locator(id,css selector, path)

Step 1: Write the following code

package seleniumtest\_firefox; import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver; import org.openqa.selenium.WebElement;

import org.openqa.selenium.firefox.FirefoxDriver; import org.openqa.selenium.firefox.FirefoxOptions;

public class Locators {

public static void main(String[] args) {

// Set up Firefox driver System.setProperty("webdriver.gecko.driver",

"D:\\Saiganesh\\Projects\\2024\\java\_selenium\\firefox\_gecko\_driver\\ge ckodriver.exe");

// Set Firefox binary path (if required) FirefoxOptions options = new FirefoxOptions(); options.setBinary("C:\\Program Files\\Mozilla

Firefox\\firefox.exe");

// Initialize WebDriver

WebDriver driver = new FirefoxDriver(options);

// Open Facebook driver.get("https://[www.facebook.com/](http://www.facebook.com/)");

try {

// 1. Get the title and current URL String title = driver.getTitle(); System.out.println("Title is: " + title); System.out.println("Current URL is: " +

driver.getCurrentUrl());

Thread.sleep(3000); // Wait for 3 seconds

// 2. Locate element by ID

WebElement emailField = driver.findElement(By.id("email")); System.out.println("Email Field (ID): " +

emailField.getTagName()); // HTML tag name of the element Thread.sleep(3000); // Wait for 3 seconds

// 3. Locate element by CSS Selector WebElement emailFieldByCss =

driver.findElement(By.cssSelector("input[name='email']")); System.out.println("Email Field (CSS Selector): " +

emailFieldByCss.getTagName()); // HTML tag name of the element Thread.sleep(3000); // Wait for 3 seconds

// 4. Locate element by XPath WebElement emailFieldByXpath =

driver.findElement(By.xpath("//\*[@id='email']")); System.out.println("Email Field (XPath): " +

emailFieldByXpath.getTagName()); // HTML tag name of the element Thread.sleep(3000); // Wait for 3 seconds

pg. 46

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

selected

// 5. Check if the element is displayed, enabled, and

System.out.println("Is the Email Field Displayed: " +

emailField.isDisplayed());

System.out.println("Is the Email Field Enabled: " + emailField.isEnabled());

System.out.println("Is the Email Field Selected: " + emailField.isSelected());

Thread.sleep(3000); // Wait for 3 seconds

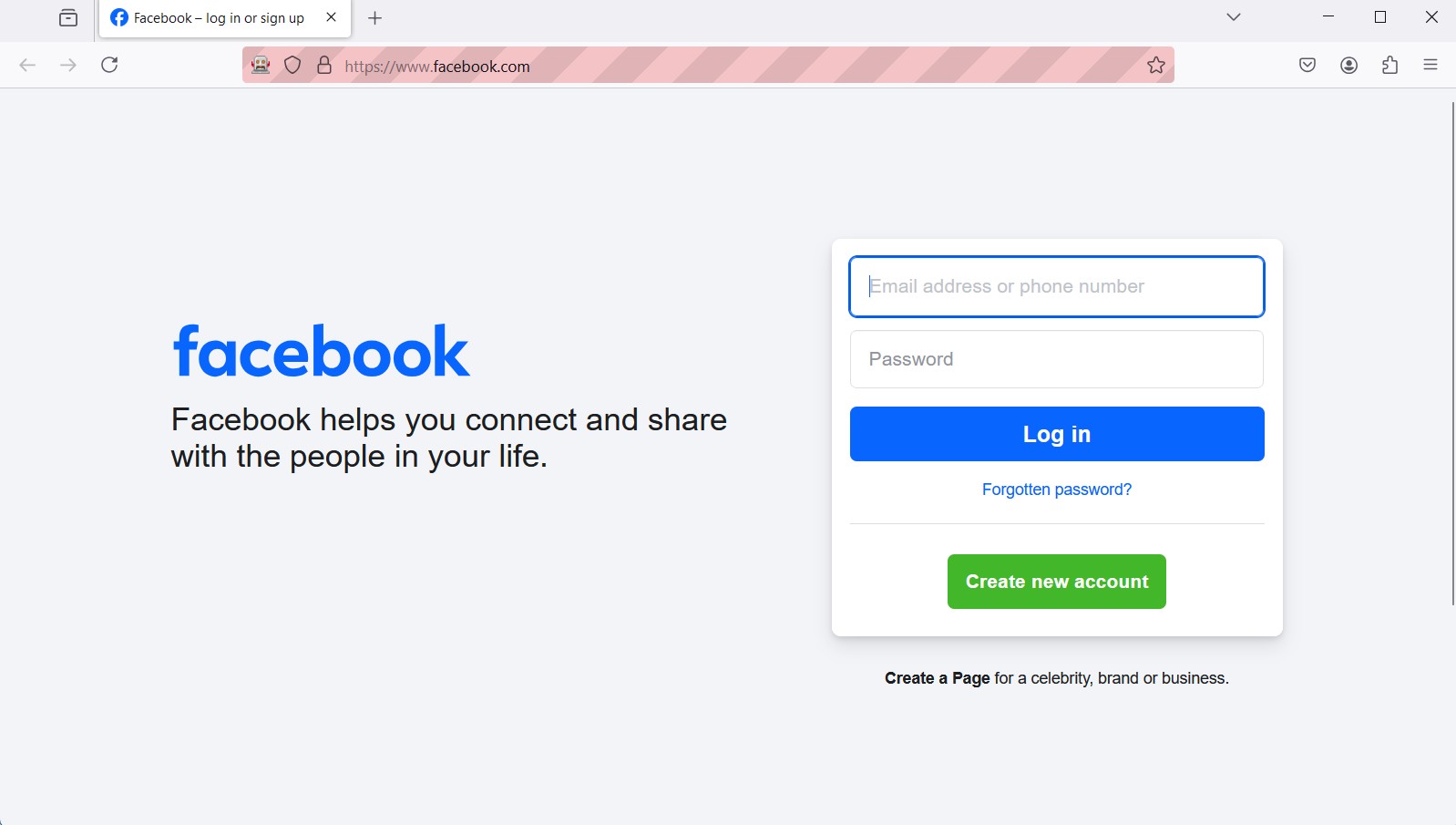
} catch (InterruptedException e) { e.printStackTrace();

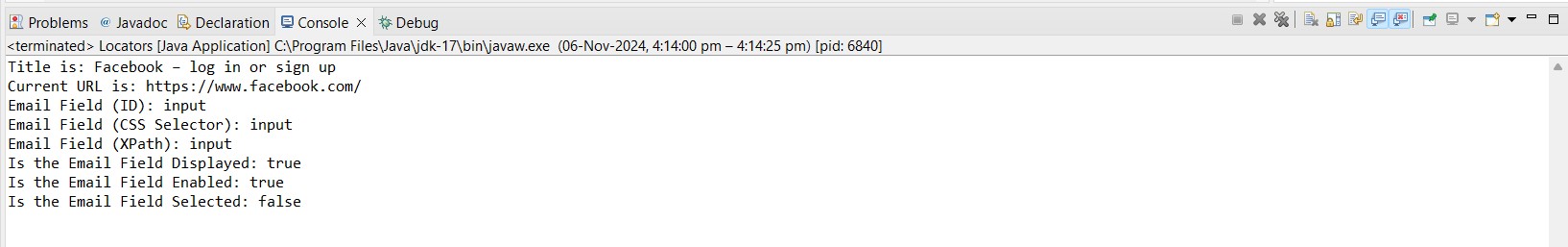
}

// Close the browser driver.quit();

}

}





pg. 47

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

**PRACTICAL NO.10:** Demonstrate different types of alerts Step 1: Write the following code:

package seleniumtest\_firefox;

import org.openqa.selenium.By; import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.firefox.FirefoxDriver; import org.openqa.selenium.firefox.FirefoxOptions;

public class Alerts {

public static void main(String[] args) {

// Setup Firefox WebDriver System.setProperty("webdriver.gecko.driver",

"D:\\Saiganesh\\Projects\\2024\\java\_selenium\\firefox\_gecko\_driver\\ge ckodriver.exe");

FirefoxOptions options = new FirefoxOptions(); options.setBinary("C:\\Program Files\\Mozilla

Firefox\\firefox.exe");

WebDriver driver = new FirefoxDriver(options);

try {

// Test Alert

driver.get(["http://ww](http://www.w3schools.com/js/tryit.asp?filename=tryjs_alert)w[.w3schools.com/js/tryit.asp?filename=tryjs\_alert](http://www.w3schools.com/js/tryit.asp?filename=tryjs_alert)"

);

driver.switchTo().frame("iframeResult"); WebElement alertButton =

driver.findElement(By.cssSelector("html>body>button")); alertButton.click(); System.out.println("Alert Message: " +

driver.switchTo().alert().getText());

Thread.sleep(3000); // Wait for 3 seconds to read the alert driver.switchTo().alert().accept(); // Accept the alert

// Test Confirm (Accept)

driver.get(["http://ww](http://www.w3schools.com/js/tryit.asp?filename=tryjs_confir)w[.w3schools.com/js/tryit.asp?filename=tryjs\_confir](http://www.w3schools.com/js/tryit.asp?filename=tryjs_confir) m");

driver.switchTo().frame("iframeResult"); WebElement confirmButton =

driver.findElement(By.cssSelector("html>body>button")); confirmButton.click(); System.out.println("Confirm Message after OK: " +

driver.switchTo().alert().getText());

Thread.sleep(3000); // Wait for 3 seconds to read the confirm message

driver.switchTo().alert().accept(); // Accept the confirm

// Test Confirm (Dismiss)

driver.get(["http://ww](http://www.w3schools.com/js/tryit.asp?filename=tryjs_confir)w[.w3schools.com/js/tryit.asp?filename=tryjs\_confir](http://www.w3schools.com/js/tryit.asp?filename=tryjs_confir) m");

driver.switchTo().frame("iframeResult"); confirmButton.click();

pg. 48

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

System.out.println("Confirm Message after Cancel: " + driver.switchTo().alert().getText());

Thread.sleep(3000); // Wait for 3 seconds to read the confirm message

driver.switchTo().alert().dismiss(); // Dismiss the confirm

// Test Prompt

driver.get(["http://ww](http://www.w3schools.com/js/tryit.asp?filename=tryjs_prompt)w[.w3schools.com/js/tryit.asp?filename=tryjs\_prompt](http://www.w3schools.com/js/tryit.asp?filename=tryjs_prompt) ");

driver.switchTo().frame("iframeResult"); WebElement promptButton =

driver.findElement(By.cssSelector("html>body>button")); promptButton.click(); driver.switchTo().alert().sendKeys("Test User"); driver.switchTo().alert().accept(); // Accept the prompt System.out.println("Prompt Message: " +

driver.findElement(By.id("demo")).getText());

Thread.sleep(3000); // Wait for 3 seconds to read the prompt message

} catch (InterruptedException e) { e.printStackTrace();

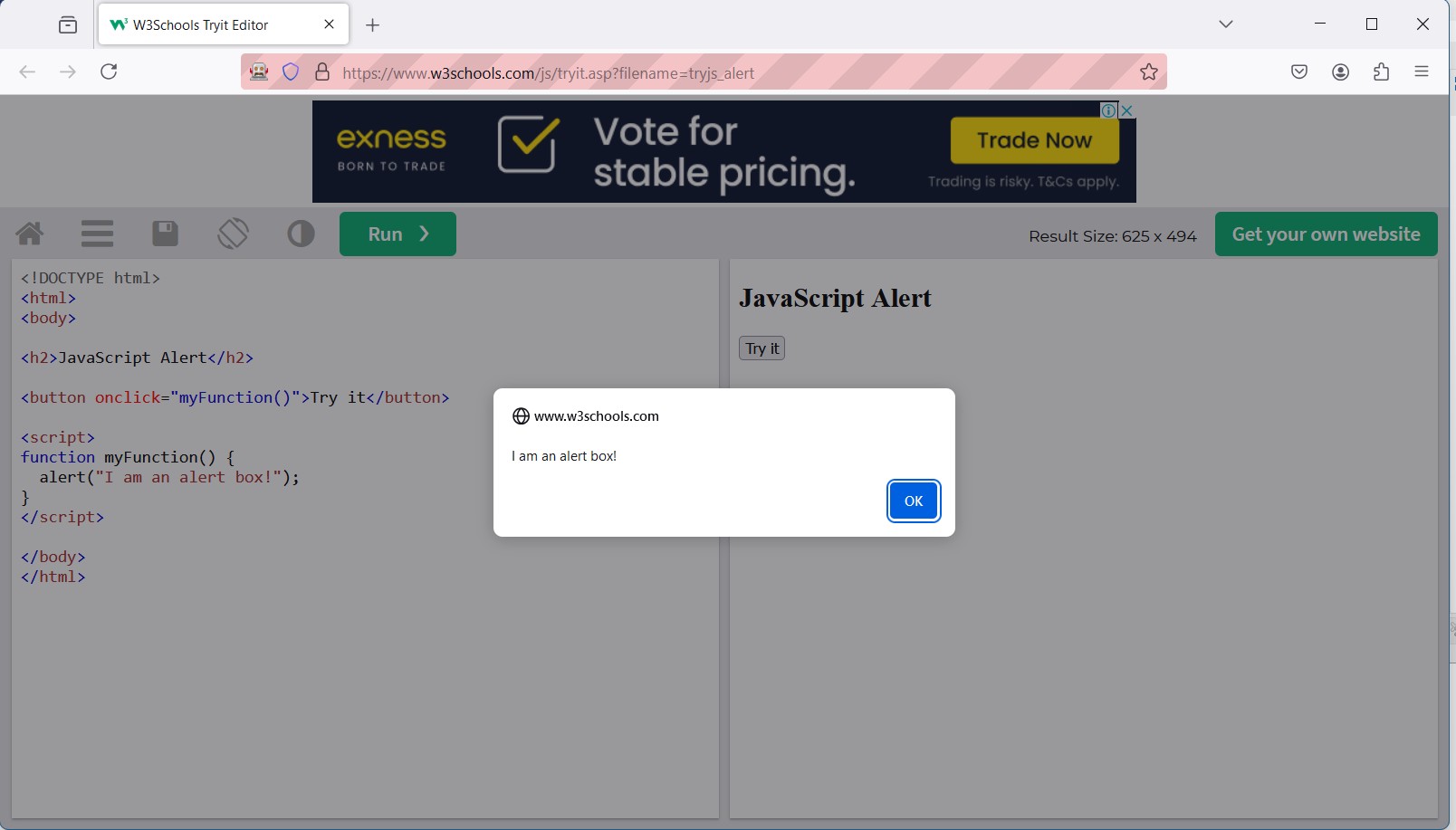
} finally {

// Close the browser driver.quit();

}

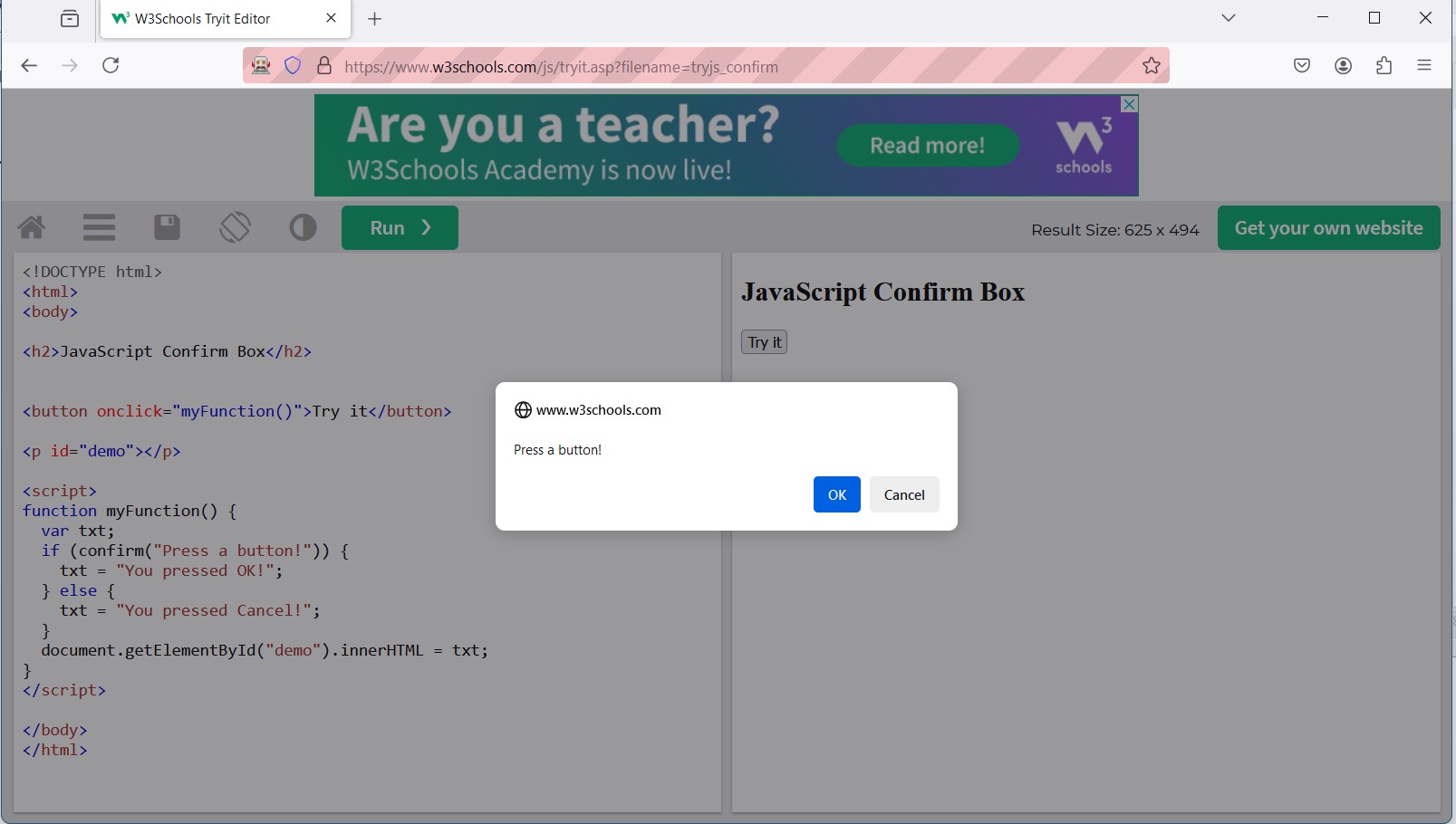
}

}



pg. 49

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB



pg. 50

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

**PRACTICAL NO.11:** Demonstrate the following actions

Prepare test cases for below given applications using Boundary Value Analysis and Equivalence Class Partitioning:

1. Handling dropdown’s
2. List Boxes
3. Command Buttons
4. Radio Buttons & text boxes
5. Waits command in selenium Step 1: Write the code below:

package seleniumtest\_firefox;

import org.openqa.selenium.By; import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.firefox.FirefoxDriver; import org.openqa.selenium.firefox.FirefoxOptions; import org.openqa.selenium.support.ui.Select;

import java.util.List;

public class DropDownAndListBoxHandling { public static void main(String[] args) {

// Setup Firefox WebDriver System.setProperty("webdriver.gecko.driver",

"D:\\Saiganesh\\Projects\\2024\\java\_selenium\\firefox\_gecko\_driver\\ge ckodriver.exe");

FirefoxOptions options = new FirefoxOptions(); options.setBinary("C:\\Program Files\\Mozilla

Firefox\\firefox.exe");

WebDriver driver = new FirefoxDriver(options);

try {

// Task 1: Navigate to Facebook registration page and handle dropdown and list box

driver.get("https://[www.facebook.com/r.php?r=101](http://www.facebook.com/r.php?r=101)");

// 1. Dropdown Handling (Selecting a month from dropdown) Select month = new

Select(driver.findElement(By.id("month"))); month.selectByVisibleText("Apr"); // Select October Thread.sleep(3000); // Wait for 3 seconds to read the

selection

// 2. List Box Handling (Selecting Gender options) List<WebElement> gender =

driver.findElements(By.name("sex"));

int cnt = gender.size();

System.out.println("Number of Gender options: " + cnt);

// Loop through and click each gender option for (int i = 0; i < cnt; i++) {

String text = gender.get(i).getAttribute("value");

pg. 51

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

System.out.println("Gender option: " + text); gender.get(i).click();

Thread.sleep(2000); // Wait for 2 seconds before clicking the next option

}

// Task 2: Navigate to Google and get the title driver.get("https://[www.google.com/](http://www.google.com/)");

String title = driver.getTitle();

System.out.println("Title is: " + title); // Print the title of Google

System.out.println(driver.getTitle());

} catch (InterruptedException e) { e.printStackTrace();

} finally {

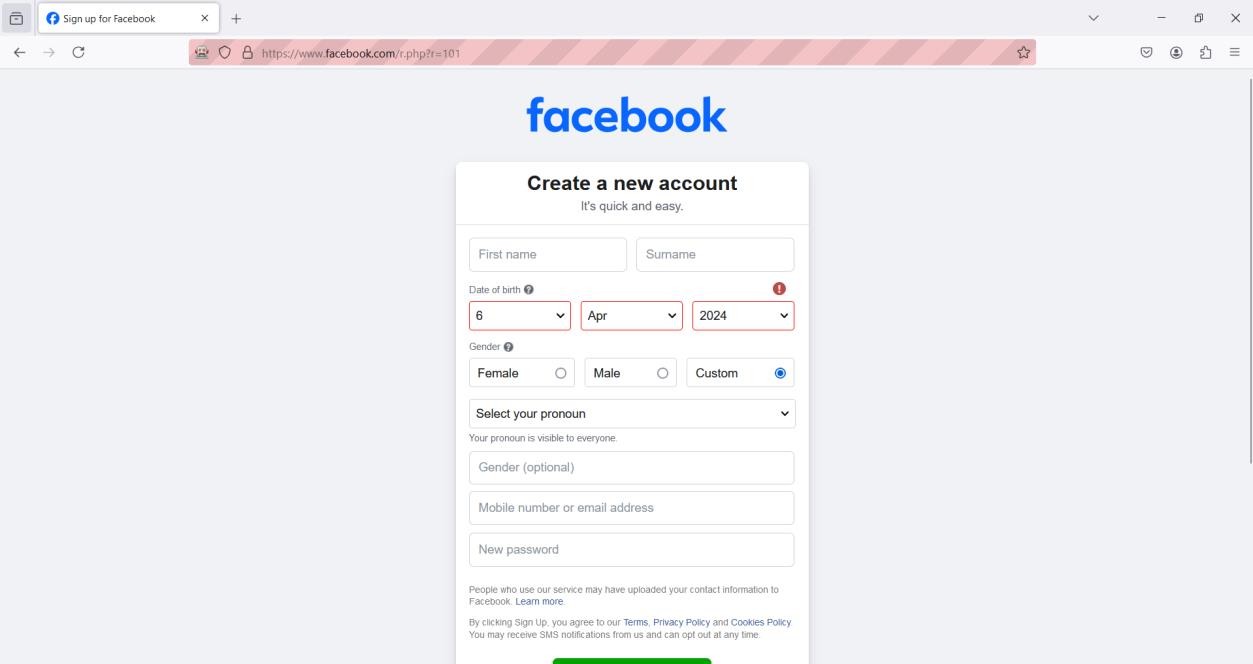
// Close the browser driver.quit();

}

}

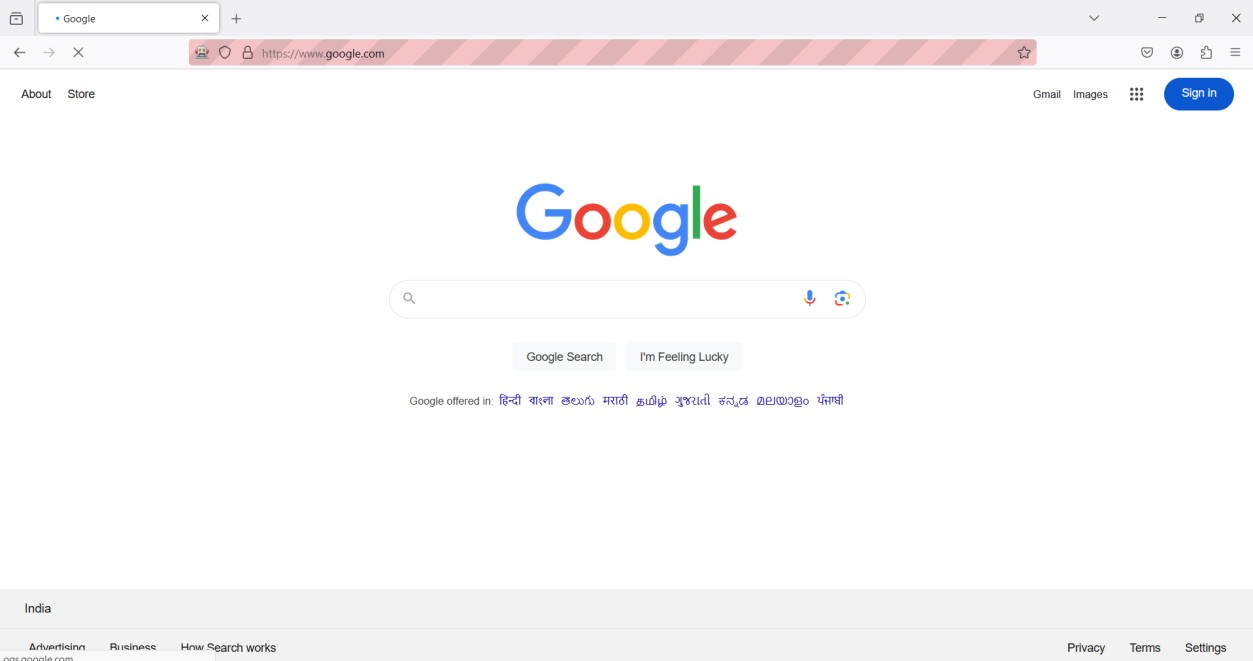
}

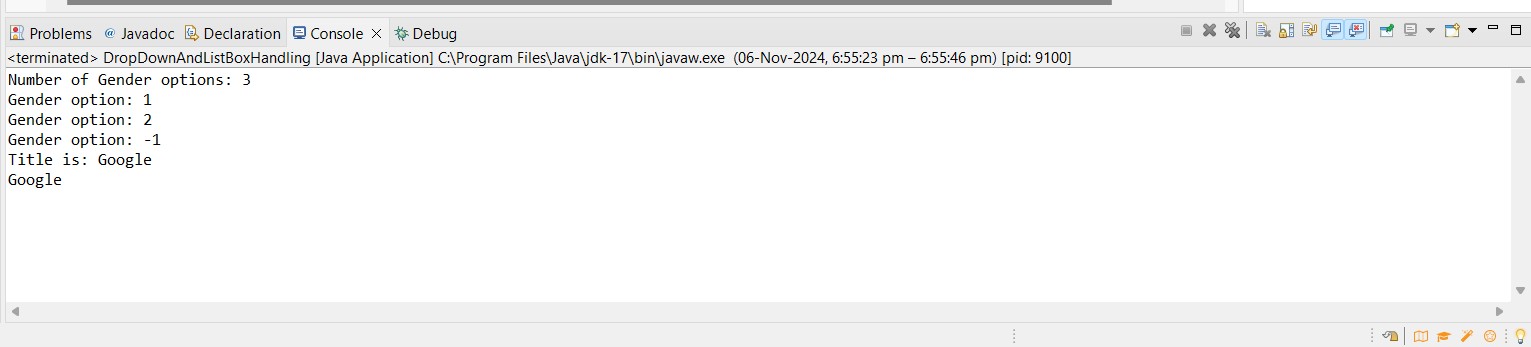
Output below:



pg. 52

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB





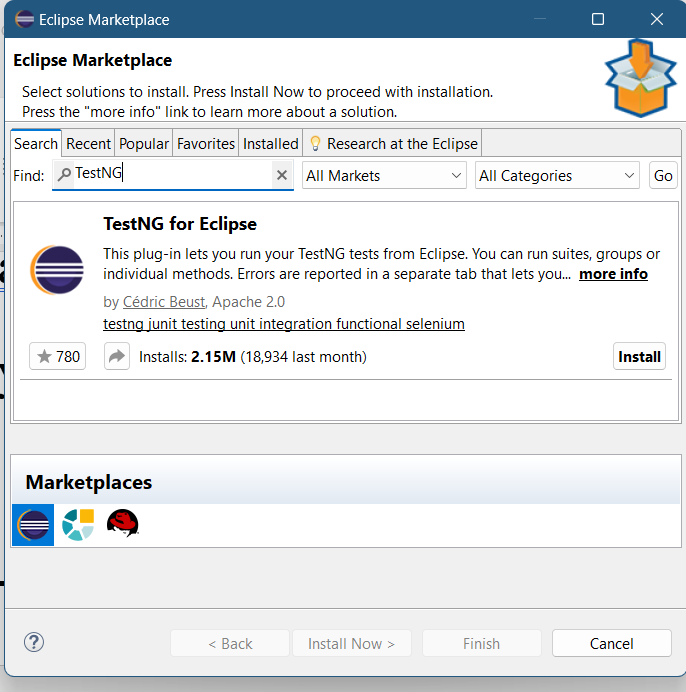
pg. 53

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

**PRACTICAL NO.12:** Installation of TestNg , running testNg and TestNg annotations.

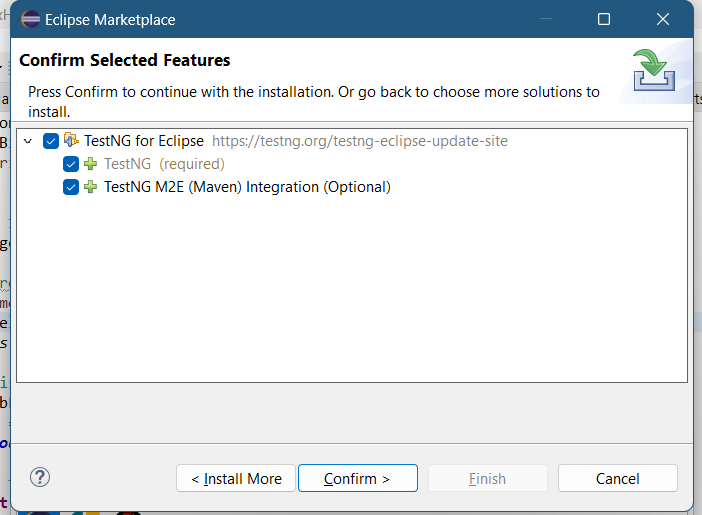
The following are the steps to install TestNG framework

Step 1: Search for TestNG in Eclipse marketplace and download it

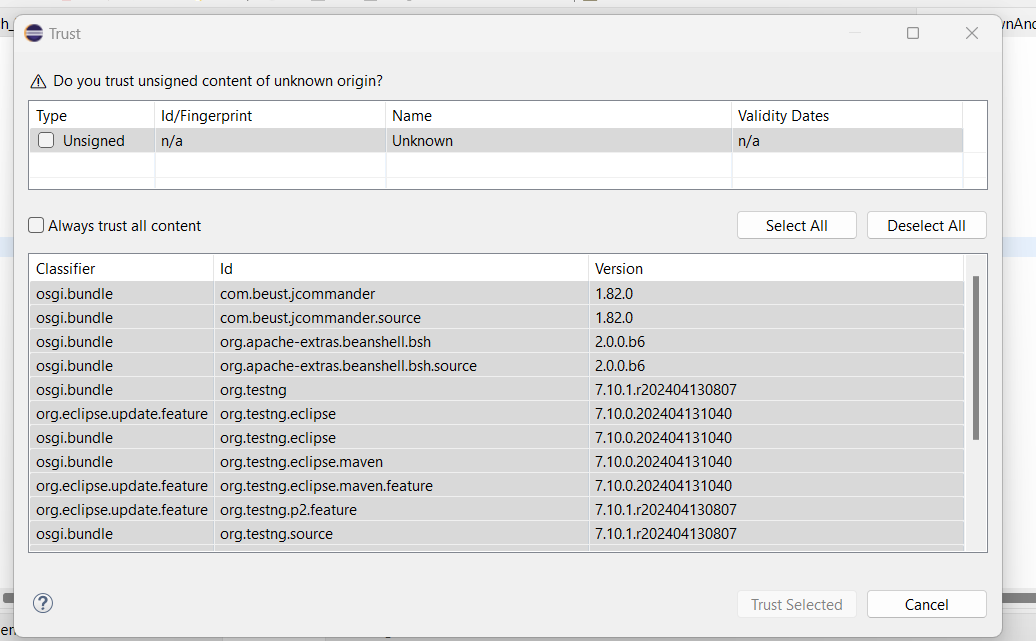


pg. 54

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB



Step 2: If you encounter any security issue click on ‘Trust all’ and install it



Step 3: Finish the installation and restart your system

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

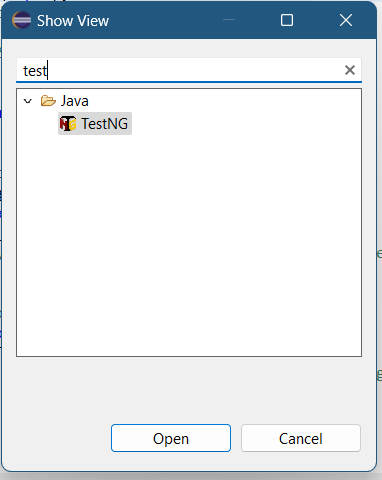
“Restart now.”

pg. 55

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

Verify if the installation is done properly

After the restart, verify if TestNG for Eclipse was successfully installed. Click Window > Show View > Other.



pg. 56

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

The following are the steps to Run TestNG Annotation.

Step 1: First download the two external JAR files and add them in your build path under the classpath section. The JAR files are slf4j-api.jar and slf4j-simple.jar from <http://www.java2s.com/>. Then Add those JARs in Configure Built-Path > Under the Classpath.

Step 2: Write the code for TestNG1:

package seleniumtest\_firefox;

import org.openqa.selenium.By; import org.openqa.selenium.Keys; import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver; import org.openqa.selenium.firefox.FirefoxOptions; import org.testng.annotations.Test;

public class TestNG1 {

public String baseUrl = "https:/[/www](http://www.google.com/).[google.com/"](http://www.google.com/); String driverPath =

"D:\\Saiganesh\\Projects\\2024\\java\_selenium\\firefox\_gecko\_driver\\ge ckodriver.exe";

public WebDriver driver;

@Test

public void f() throws InterruptedException { System.out.println("Launching Firefox browser"); System.setProperty("webdriver.gecko.driver", driverPath);

// Set Firefox binary path if needed FirefoxOptions options = new FirefoxOptions(); options.setBinary("C:\\Program Files\\Mozilla

Firefox\\firefox.exe");

driver = new FirefoxDriver(options); driver.get(baseUrl);

// Perform a search on Google driver.findElement(By.name("q")).sendKeys("DES NMITD",

Keys.ENTER);

Thread.sleep(2000);

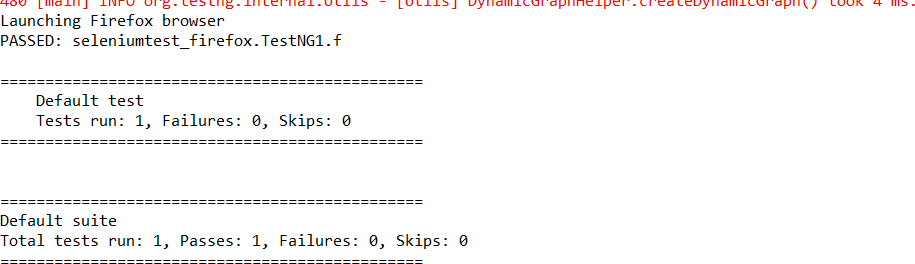
// Close the browser driver.quit();

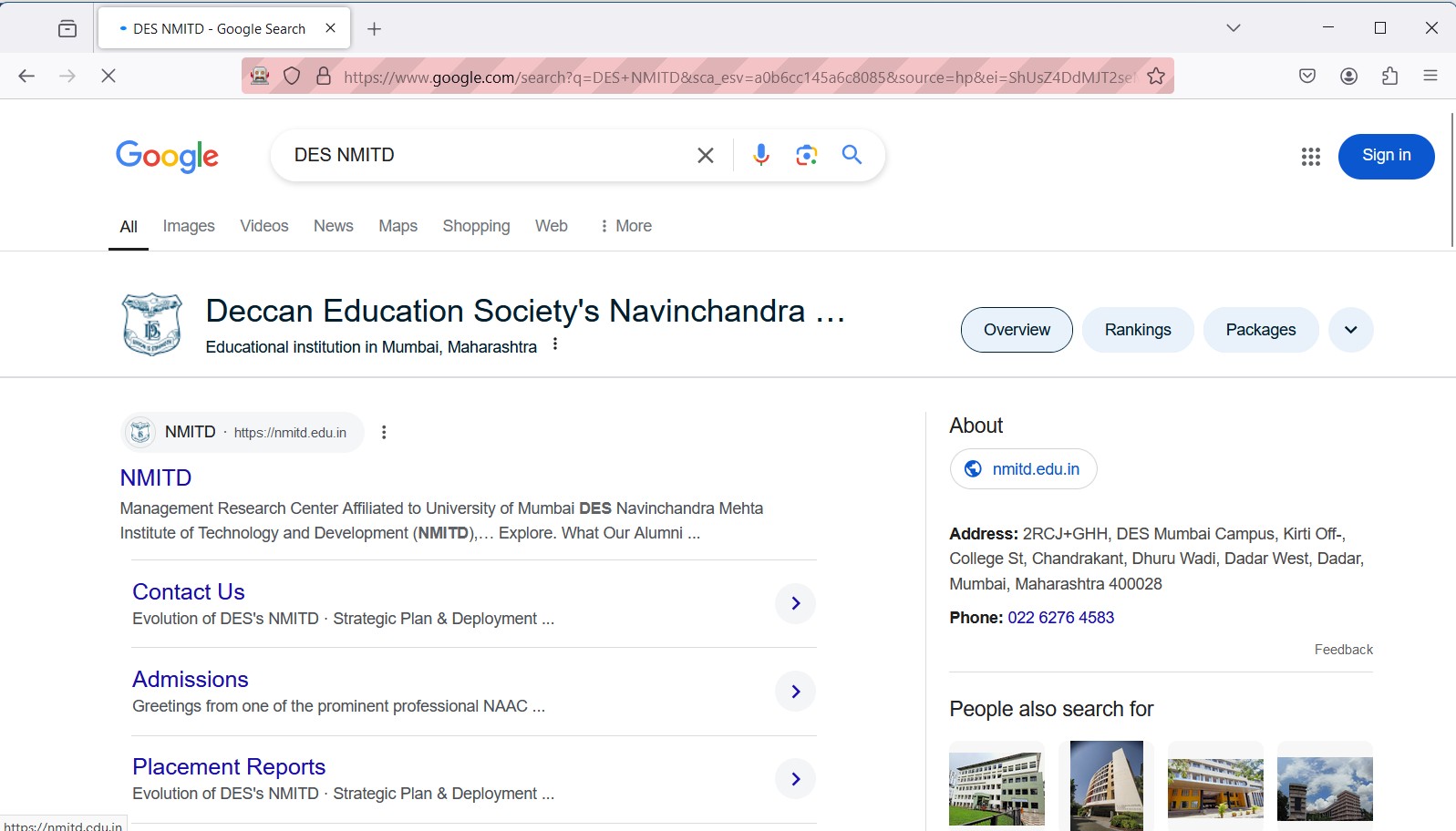
}

}

pg. 57

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB





Step 3: Write the code for TestNG2 – Running TestNG:

package seleniumtest\_firefox;

import org.openqa.selenium.By; import org.openqa.selenium.Keys; import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver; import org.openqa.selenium.firefox.FirefoxOptions; import org.testng.annotations.Test;

public class TestNG2 { String driverPath =

"D:\\Saiganesh\\Projects\\2024\\java\_selenium\\firefox\_gecko\_driver\\ge ckodriver.exe";

@Test

public void TestGoogle() throws InterruptedException { System.setProperty("webdriver.gecko.driver", driverPath);

// Set Firefox binary path if needed FirefoxOptions options = new FirefoxOptions(); options.setBinary("C:\\Program Files\\Mozilla

Firefox\\firefox.exe");

pg. 58

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

WebDriver driver = new FirefoxDriver(options); driver.get("https://[www.google.com/](http://www.google.com/)");

// Perform a search on Google driver.findElement(By.name("q")).sendKeys("DES's NMITD",

Keys.ENTER);

Thread.sleep(2000);

// Close the browser driver.quit();

}

@Test

public void TestFacebook() throws InterruptedException { System.setProperty("webdriver.gecko.driver", driverPath);

// Set Firefox binary path if needed FirefoxOptions options = new FirefoxOptions(); options.setBinary("C:\\Program Files\\Mozilla

Firefox\\firefox.exe");

WebDriver driver = new FirefoxDriver(options); driver.get("https://[www.facebook.com/](http://www.facebook.com/)");

// Enter text in the Facebook email field driver.findElement(By.name("email")).sendKeys("DES's NMITD",

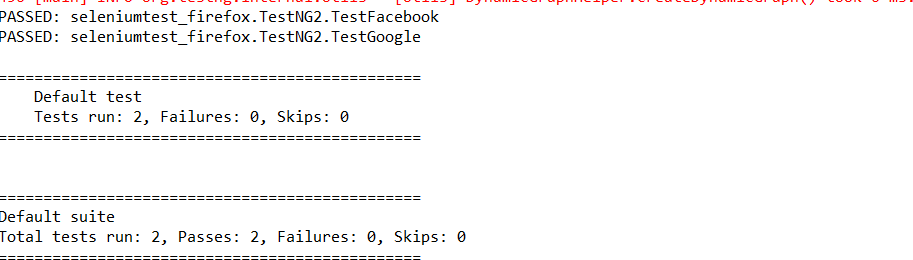
Keys.ENTER);

Thread.sleep(2000);

// Close the browser driver.quit();

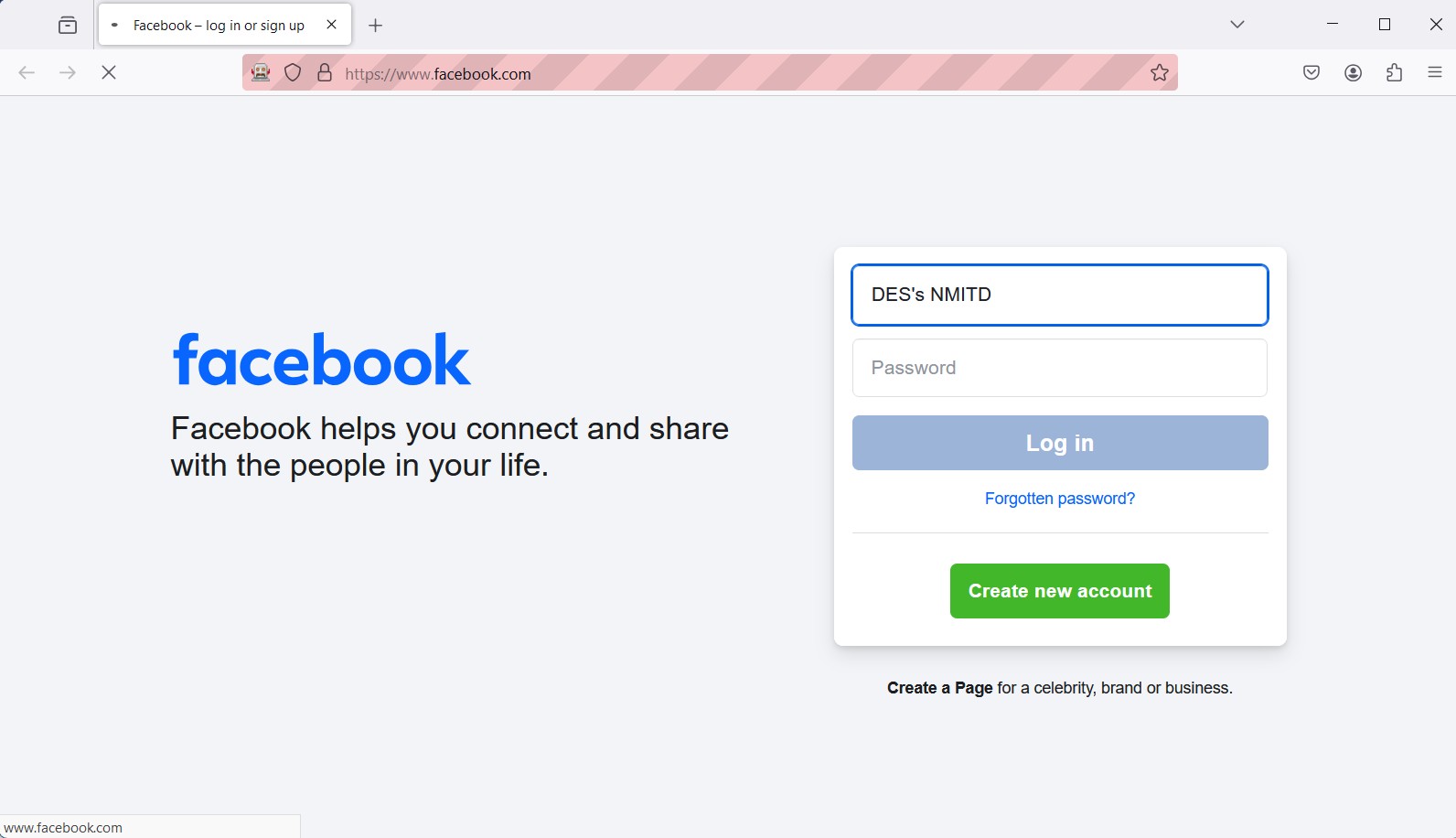
}

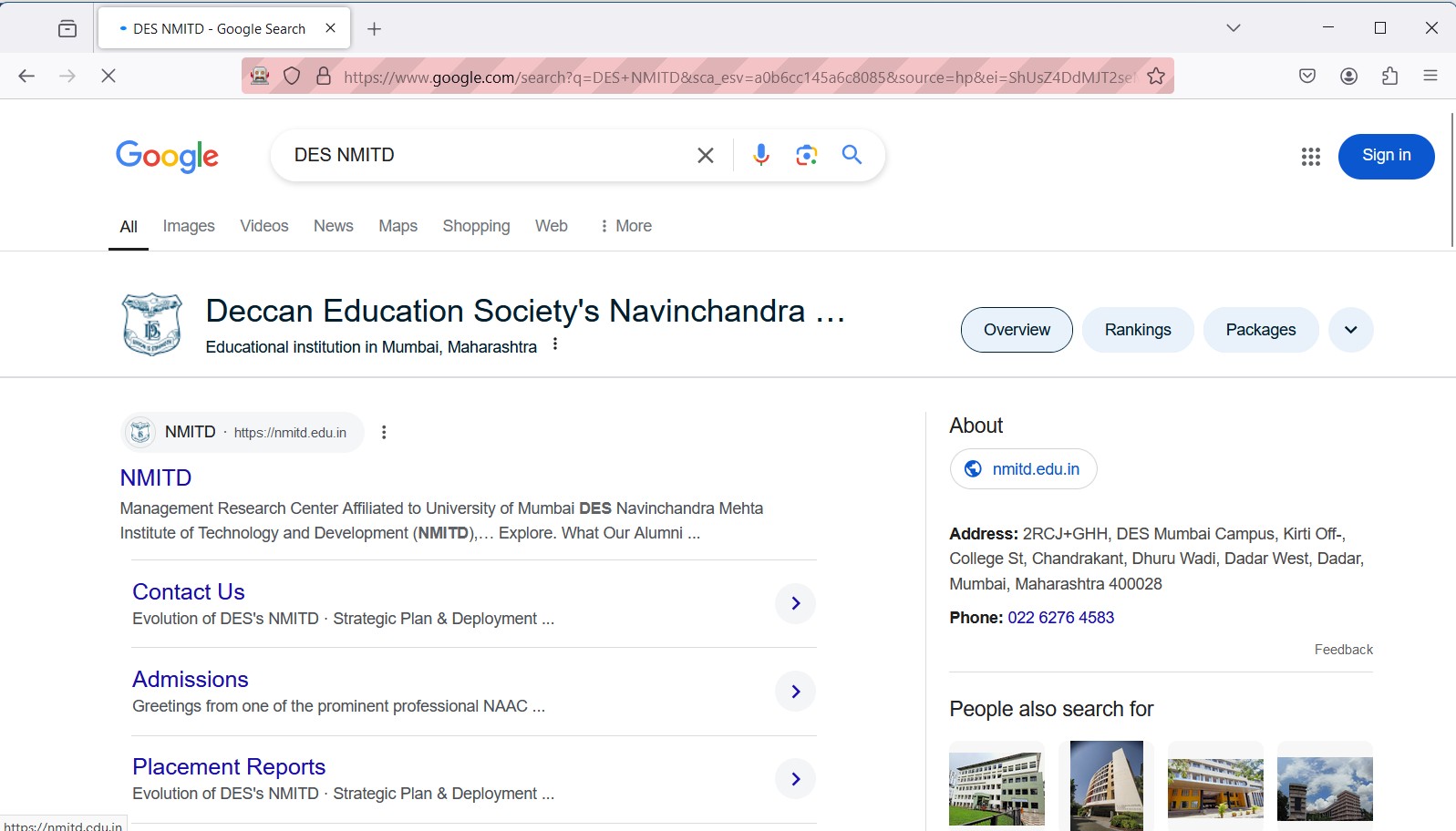
}



pg. 59

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB





package seleniumtest\_firefox;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver; import org.openqa.selenium.firefox.FirefoxOptions; import org.testng.Assert;

import org.testng.annotations.AfterTest; import org.testng.annotations.BeforeTest; import org.testng.annotations.Test;

public class TestNG3 {

String actualTitle, expectedTitle; WebDriver driver;

pg. 60

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

@BeforeTest

public void launchBrowser() { System.out.println("Launching Firefox browser"); System.setProperty("webdriver.gecko.driver",

"D:\\Saiganesh\\Projects\\2024\\java\_selenium\\firefox\_gecko\_driver\\ge ckodriver.exe");

FirefoxOptions options = new FirefoxOptions(); options.setBinary("C:\\Program Files\\Mozilla

Firefox\\firefox.exe");

driver = new FirefoxDriver(options); driver.get("https://demo.guru99.com/test/newtours/");

}

@Test

public void verifyHomepageTitle() { expectedTitle = "Welcome: Mercury Tours"; actualTitle = driver.getTitle();

Assert.assertEquals(actualTitle, expectedTitle);

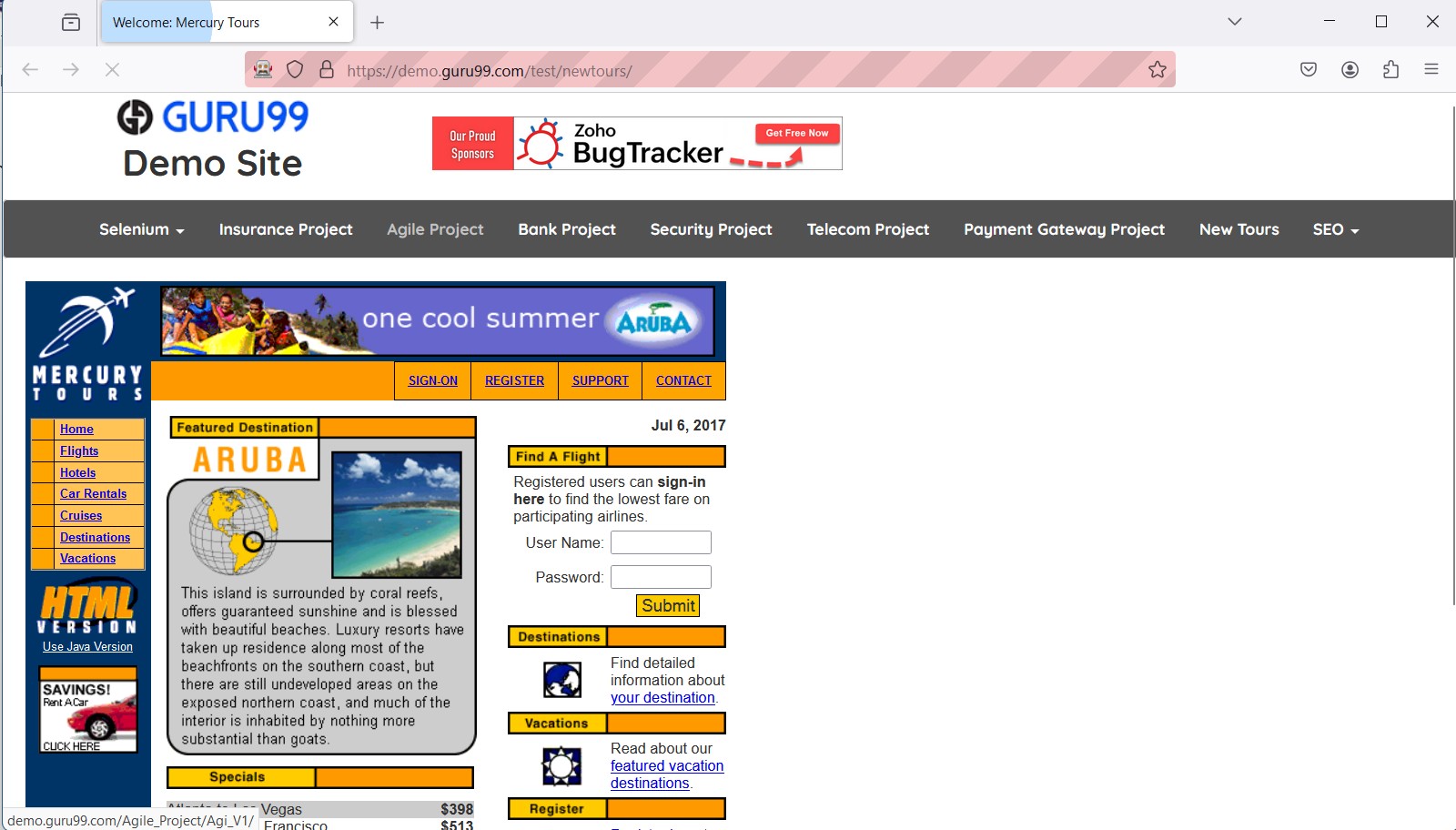
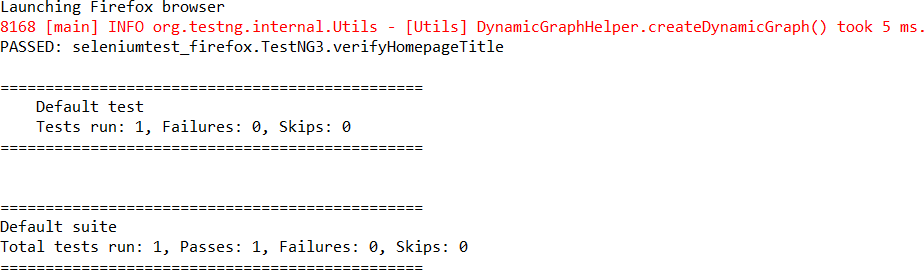
}

@AfterTest

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

}

}



pg. 61

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

**PRACTICAL NO.13:** Demonstrate Validation Testing Step 1: Write the below code:

package seleniumtest\_firefox;

import org.openqa.selenium.By; import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

public class Validations\_Test {

public static void main(String[] args) { System.setProperty("webdriver.gecko.driver",

"D:\\Saiganesh\\Projects\\2024\\java\_selenium\\firefox\_gecko\_driver\\ge ckodriver.exe");

WebDriver driver = new FirefoxDriver(); driver.get("https://[www.google.com/](http://www.google.com/)");

// Validate the title of the page String title = driver.getTitle();

System.out.println("Title is: " + title);

// Validate the current URL System.out.println("Current URL is: " +

driver.getCurrentUrl());

// Check if the search box is visible using the updated XPath boolean status =

driver.findElement(By.xpath("//textarea[@name='q']")).isDisplayed(); if (status) {

System.out.println("Search box is visible");

} else {

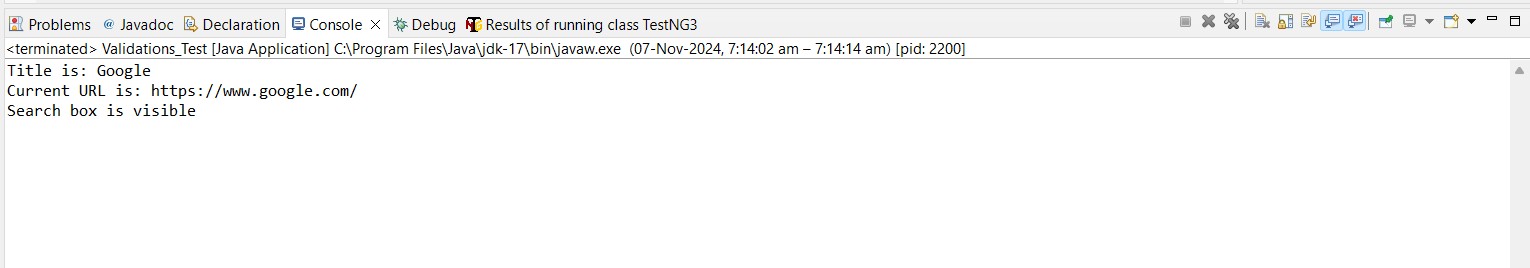
System.out.println("Search box is not visible");

}

// Close the browser driver.quit();

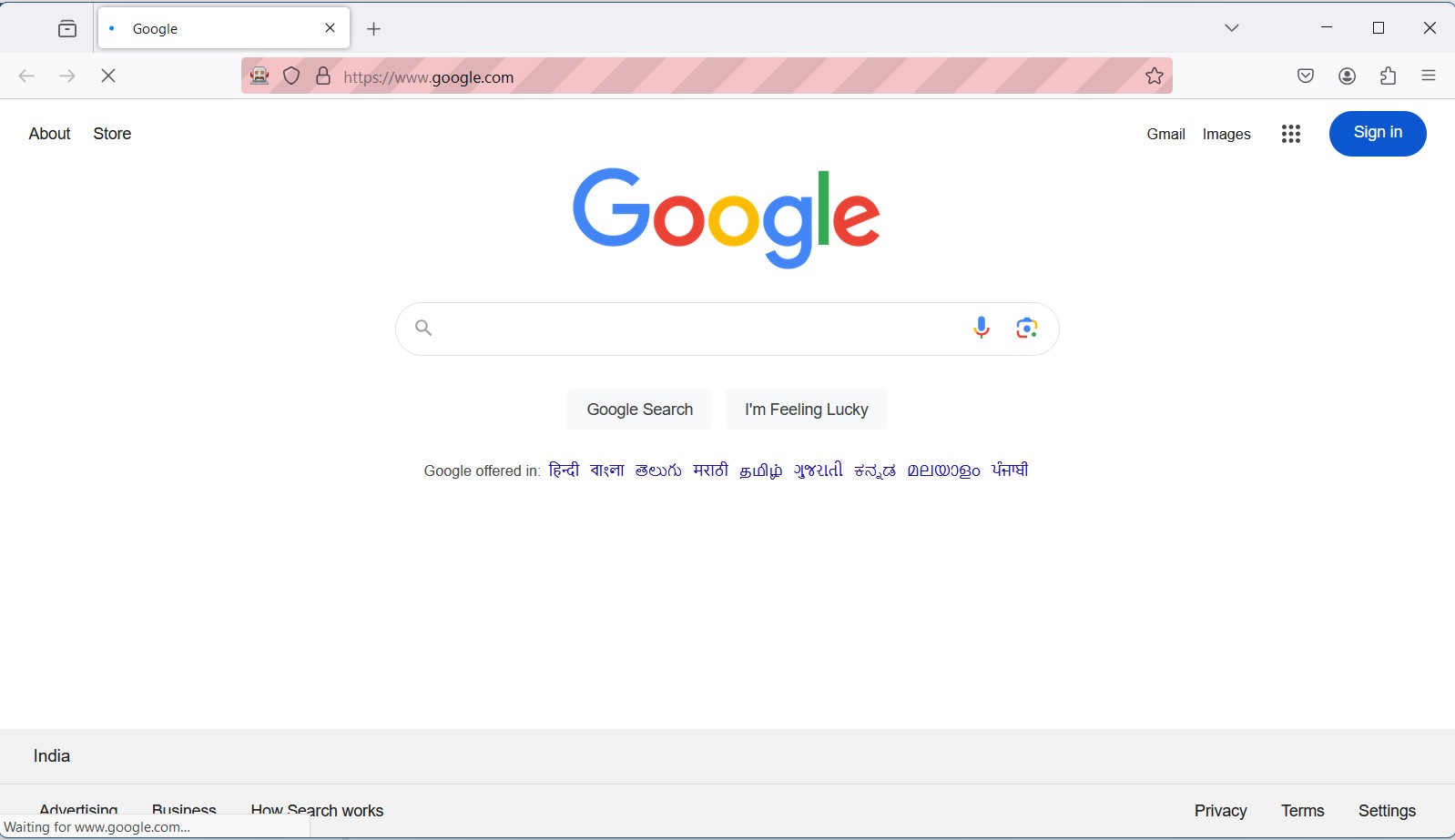
}

}



pg. 62

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB



pg. 63

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

**PRACTICAL NO.14:** Perform Regression Testing

Step 1: Create a java class inside a package and paste the following code .package seleniumtest\_firefox;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver; import org.testng.Assert;

import org.testng.annotations.AfterTest; import org.testng.annotations.BeforeTest; import org.testng.annotations.Test;

public class RegressionTest { WebDriver driver;

String baseUrl = "https://[www.google.com/](http://www.google.com/)"; String expectedTitle = "Google";

@BeforeTest

public void setUp() { System.setProperty("webdriver.gecko.driver",

"path\_to\_your\_geckodriver");

driver = new FirefoxDriver(); driver.get(baseUrl);

}

@Test

public void validateTitle() {

String actualTitle = driver.getTitle();

Assert.assertEquals(actualTitle, expectedTitle, "Title doesn't match!");

}

@AfterTest

public void tearDown() { driver.quit();

}

}

Step 2: Create a new package inside the same package and create two java classes

Step 3: Paste the code in a java class inside the newly created package(here the name of class is Tranform.java)

package seleniumtest\_firefox.listeners;

import java.lang.reflect.Constructor; import java.lang.reflect.Method;

import org.testng.IAnnotationTransformer;

pg. 64

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

import org.testng.annotations.ITestAnnotation;

public class Transform implements IAnnotationTransformer { @Override

public void transform(ITestAnnotation testAnnotation, Class testClass, Constructor testConstructor, Method testMethod) {

// Manually set the retry analyzer class

if (testAnnotation.getRetryAnalyzerClass() == null) { testAnnotation.setRetryAnalyzer(RetryFailedTestCases.class);

}

}

}

Step 4: Paste the code in a java class inside the newly created package(here the name of class is RetryFailedTestCases.java)

package seleniumtest\_firefox.listeners;

import org.testng.IRetryAnalyzer; import org.testng.ITestResult;

public class RetryFailedTestCases implements IRetryAnalyzer { int noOfRetries = 0;

int maxRetries = 3;

@Override

public boolean retry(ITestResult result) { if (noOfRetries < maxRetries) {

System.out.println(result.getName() + " retrying test, attempt " + (noOfRetries + 1)); noOfRetries++;

pg. 65

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

return true;

}

return false;

}

}

Step 5 Create a XML file(here the name of file is testing.xml) outside the package but inside the project src folder. The folder structure should look like this: your\_project\_name/src/testing.xml

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE suite SYSTEM ["http://testng.org/test](http://testng.org/testng-1.0.dtd)n[g-1.0.dtd"](http://testng.org/testng-1.0.dtd)>

<suite name="RegressionSuite">

<listeners>

<listener class-name="seleniumtest\_firefox.listeners.Transform" />

</listeners>

<test name="RegressionTest">

<classes>

<class name="seleniumtest\_firefox.RegressionTest" />

</classes>

</test>

</suite> Step 6:

1. Right-click on the testng.xml file in your project.
2. Go to Run Configurations:

Right-click on your project in the Package Explorer.

Select Run As > Run Configurations….

Create a New TestNG Configuration:

pg. 66

NAME: Vaishnavi Shahaji Patil ROLL NO.C23090 STQA LAB

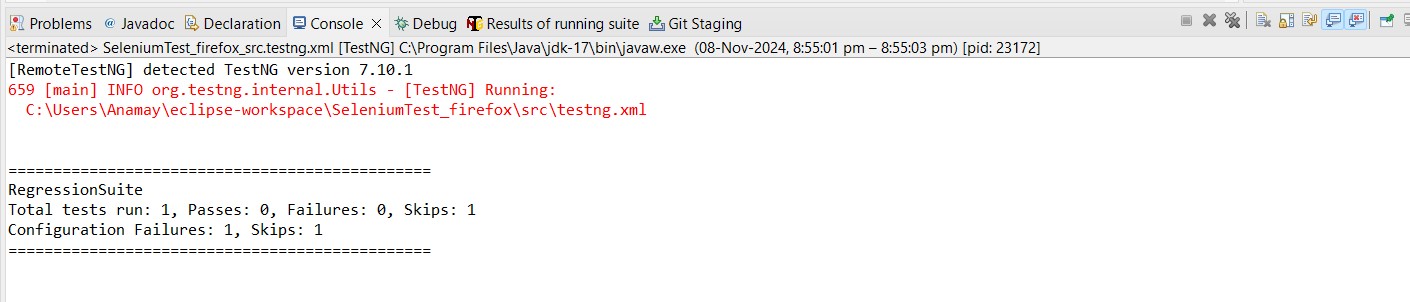
In the Run Configurations window, double-click on TestNG on the left panel to create a new configuration.

In the right panel, give your configuration a name (e.g., "RegressionTestSuite"). Select the Suite to Run:

In the Test tab, select Suite and then Browse to choose your testng.xml file. Make sure testng.xml is listed as the suite to run.

Apply and Run:

Click Apply to save your configuration, then click Run.



pg. 67