Sree Dattha Institute of Engineering & Science

(SDES)

(Approved by AICTE, New Delhi, Accredited by NAAC and Affiliated to JNTUH)

Sheriguda, Ibrahimpatnam - 501510

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

LAB MANUAL

SOFTWARE TESTING METHODOLOGIES LAB

B.Tech III YEAR II SEM (R18 JNTUH REGULATIONS)

ACADEMIC YEAR 2022-23

Vision & Mission of the Department

Vision of the Department

To be among the region's premier teaching and research Computer Science and Engineering departments producing globally competent and socially responsible graduates in the most conducive academic environment.

Mission of the Department

- To provide faculty with state of the art facilities for continuous professional development and research, both in foundational aspects and of relevance to emerging computing trends.
- To impart skills that transform students to develop technical solutions for societal needs and inculcate entrepreneurial talents.
- To inculcate an ability in students to pursue the advancement of knowledge in various specializations of Computer Science and Engineering and make them industry-ready.
- To engage in collaborative research with academia and industry and generate adequate resources for research activities for seamless transfer of knowledge resulting in sponsored projects and consultancy.
- To cultivate responsibility through sharing of knowledge and innovative computing solutions that benefit the society-at-large.
- To collaborate with academia, industry and community to set high standards in academic excellence and in fulfilling societal responsibilities.

Department of Computer Science and Engineering

PROGRAM OUTCOMES (POs)

1. **Engineering Knowledge:** Apply knowledge of mathematics and science, with fundamentals of Computer Science & Engineering to be able to solve complex engineering problems related to CSE.
2. **Problem Analysis:** Identify, Formulate, review research literature and analyze complex engineering problems related to CSE and reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences
3. **Design/Development of solutions:** Design solutions for complex engineering problems related to CSE and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety and the cultural societal and environmental considerations
4. **Conduct Investigations of Complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern Tool Usage:** Create, Select and apply appropriate techniques, resources and modern engineering and IT tools including prediction and modeling to computer science related complex engineering activities with an understanding of the limitations
6. **The Engineer and Society:** Apply Reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the CSE professional engineering practice
7. **Environment and Sustainability:** Understand the impact of the CSE professional engineering solutions in societal and environmental contexts and demonstrate the knowledge of, and need for sustainable development

8. **Ethics:** Apply Ethical Principles and commit to professional ethics and responsibilities and norms of the engineering practice
9. **Individual and Team Work:** Function effectively as an individual and as a member or leader in diverse teams and in multidisciplinary Settings
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large such as able to comprehend and with write effective reports and design documentation, make effective presentations and give and receive clear instructions.
11. **Project Management and Finance:** Demonstrate knowledge and understanding of the engineering management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multi-disciplinary environments
12. **Life-Long Learning:** Recognize the need for and have the preparation and ability to engage in independent and life-long learning the broadest context of technological change

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1: An ability to analyze the common business functions to design and develop appropriate Computer Science solutions for social upliftments.

PSO2: Shall have expertise on the evolving technologies like Python, Machine Learning, Deep Learning, Internet of Things (IOT), Data Science, Full stack development, Social Networks, Cyber Security, Big Data, Mobile Apps, CRM, ERP etc.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO 1: Graduates will endeavour to excel in their chosen careers as professionals, researchers and entrepreneurs on a global platform.

PEO 2: Graduates will demonstrate the ability to solve challenges in the fields of Engineering and Technology simultaneously catering to societal needs.

PEO 3: Graduates will strive to improve their learning curve by practising Continuing Professional Development (CPD)

PEO 4: Graduates will, at all times, adopt a professional demeanor by communicating effectively, working collaboratively, and maintaining the ethics & core values as befitting their education in interdisciplinary and emerging fields.

Course Outcomes and CO-PO Mapping

Academic year: 2022-23 **Year & Sem:- III - II**

Subject Name: Software Testing Methodologies **Branch: CSE**

Course Outcomes:

CO1: Outline the necessity of testing, debugging using program control flow.

CO2: Apply transaction flow, data flow testing to unit and integration testing. Able to test a domain or an application and identify the nice and ugly domains.

CO3: Apply reduction procedures to a control flowgraph and simplify it into a single path expression and the use of decision tables in testcase design.

CO4: Understand the testing of state graphs.

CO5: Analyze graph matrices for optimizing the code and use of testing tools like WinRunner, JMeter.

CO-PO-PSO Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	1	2	1	3								2		
CO2		2	1	3								2	2	
CO3	1		2										1	
CO4		1												
CO5	1	2	1		3							1		2

CO-PO Mapping Justification

Mapping	Level	Justification
Students will be able to		
CO1-PO1	1	Apply engineering knowledge on tests and testing levels on different software models.
CO1-PO2	2	Able to identify the various bugs after knowing the consequences of the bug.
CO1-PO3	1	Able to design control flow graph for program source code and to test the path with loops.
CO1-PO4	3	Conduct research to find set of solutions to the path predicate expression.
CO1-PO12	2	Learning continuously different test strategies on different software applications.
CO2-PO2	2	Identify transactional flow bugs, data flow bugs in unit and integrated component's.
CO2-PO3	1	Design an approach to test domain or an application.
CO2-PO4	3	Conduct research based method to identify nice and ugly domains.
CO2-PO12	2	Learning different domains continuously developed with latest technologies.
CO3-PO1	1	Apply reduction procedures to control flow graphs to simplify it into single path expression.
CO3-PO3	2	Constructing decision tables and reducing Boolean expressions by karnaugh maps for consistency.
CO4-PO2	1	Examine the software implementation of state graphs.
CO5-PO1	1	Apply the knowledge of mathematics to represent graph matrices.
CO5-PO2	2	Analyze the graph matrices for optimizing the code.
CO5-PO3	1	Implementation of node reduction algorithm without redrawing graphs.
CO5-PO5	3	Use tools for performing system testing like JMeter or WinRunner.
CO5-PO12	1	Learning continuously latest software testing tools.

CO-PSO Mapping Justification

Mapping	Level	Justification
Students will be able to		
CO2-PSO1	2	Analyzing complicated transaction and data flow of the program source code and ability to perform Domain testing by viewing programs as input data classifiers.
CO3-PSO1	1	Construct decision tables, use karnaugh maps for consistency to validate specifications.
CO5-PSO2	2	System testing can be done with tools like UFT,JMeter.

LIST OF EXPERIMENTS

SNO	NAME OF THE EXPERIMENT
	Installation Procedure
1.	Recording in Context Sensitive Mode and Analog Mode
2	GUI Checkpoint for Single Property
3	GUI Checkpoint for Single Object/Window
4	GUI Checkpoint for Multiple Objects
5	Test Case for Calculator in Windows Application

Installation Procedure of Selenium IDE

Selenium IDE

Selenium IDE (Integrated Development Environment) is an open source web automation testing tool under the Selenium Suite. Unlike Selenium WebDriver and RC, it does not require any programming logic to write its test scripts rather you can simply record your interactions with the browser to create test cases. Subsequently, you can use the playback option to re-run the test cases.

Perhaps, creating test cases on Selenium IDE does not require any programming language but when you get to use selenese commands like **runScript**, a little knowledge prior to JavaScript would prove beneficial for you to understand the concepts more clearly

Installation

Install Selenium IDE from either the Chrome or Firefox web store.

Launch the IDE

Once installed, launch it by clicking its icon from the menu bar in your browser.

Troubleshooting

Don't see the icon for Selenium IDE in your menu-bar?

Option 1

Make sure the IDE is enabled in your browser's extension settings.

You can get there quickly by typing the following into your address bar and hitting Enter.

- Chrome: chrome://extensions
- Firefox: about:addons

Option 2

The extension might be enabled but the icon is hidden. Try resizing the menu bar to give it more space.

In Chrome, you can do this by clicking to the right of the address bar, holding the click, and dragging it left or right.

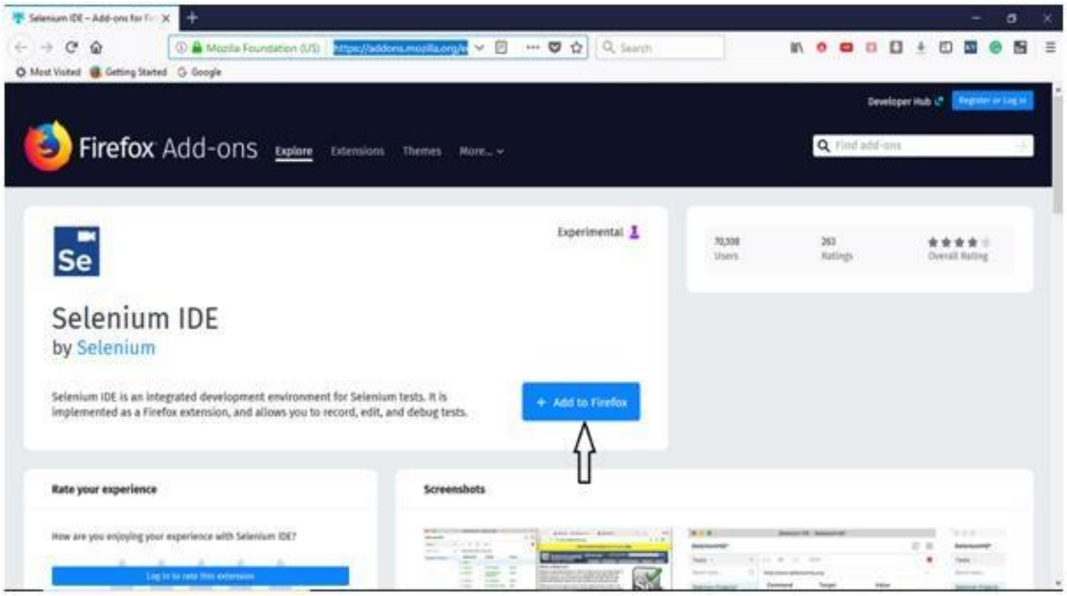
In Firefox you need to right-click, click Customize, make adjustments to the menu bar, and click Done.

What you need

- Mozilla Firefox
- Active Internet Connection

If you do not have Mozilla Firefox yet, you can download it from http://www.mozilla.org/en-US/firefox/new.

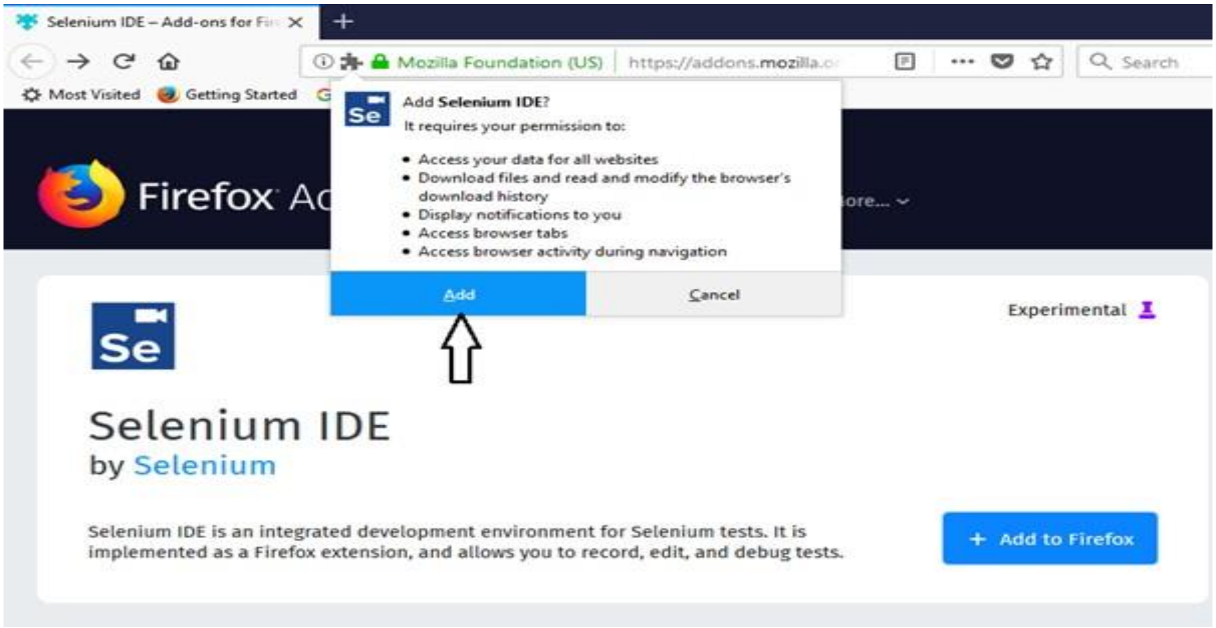
Steps 1) Launch Firefox and navigate to https://addons.mozilla.org/en-US/firefox/addon/selenium-ide/. Click on Add to Firefox



Selenium IDE is an integrated development environment for Selenium tests. It is implemented as a Firefox extension, and allows you to record, edit, and debug tests.

How are you enjoying your experience with Selenium IDE?

Steps 2) Wait until Firefox completes the download and then click "**Add.**"



Add **Selenium IDE**?

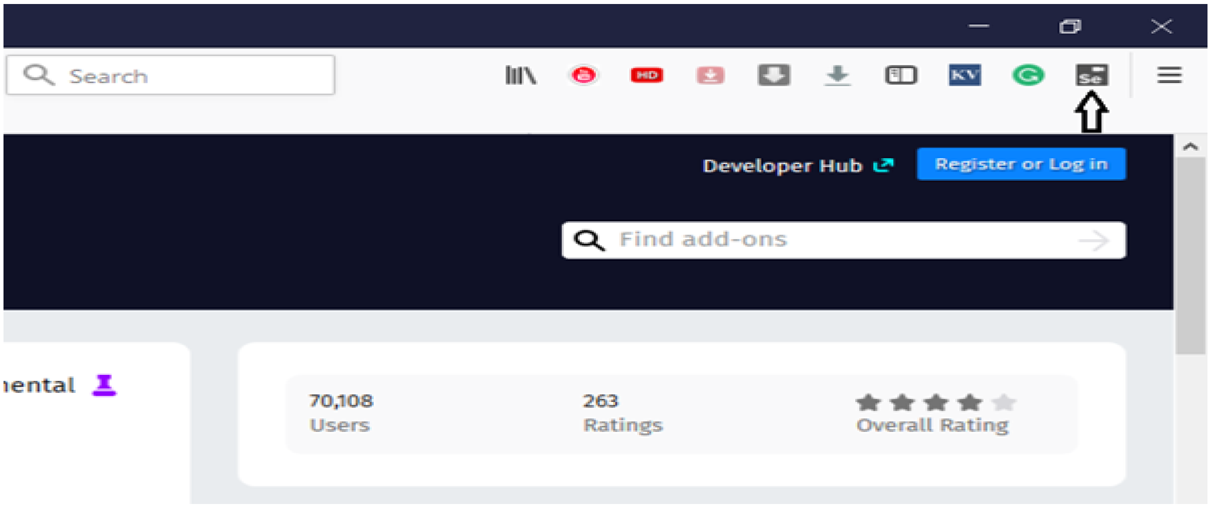
It requires your permission to:

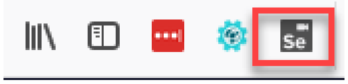
- Access your data for all websites
- Download files and read and modify the browser's download history
- Display notifications to you
- Access browser tabs
- Access browser activity during navigation

Selenium IDE is an integrated development environment for Selenium tests. It is implemented as a Firefox extension, and allows you to record, edit, and debug tests.

Steps 3) Once install is complete, you will get a confirmation message. Click **"OK"**

Steps 4) Click on the Selenium IDE icon





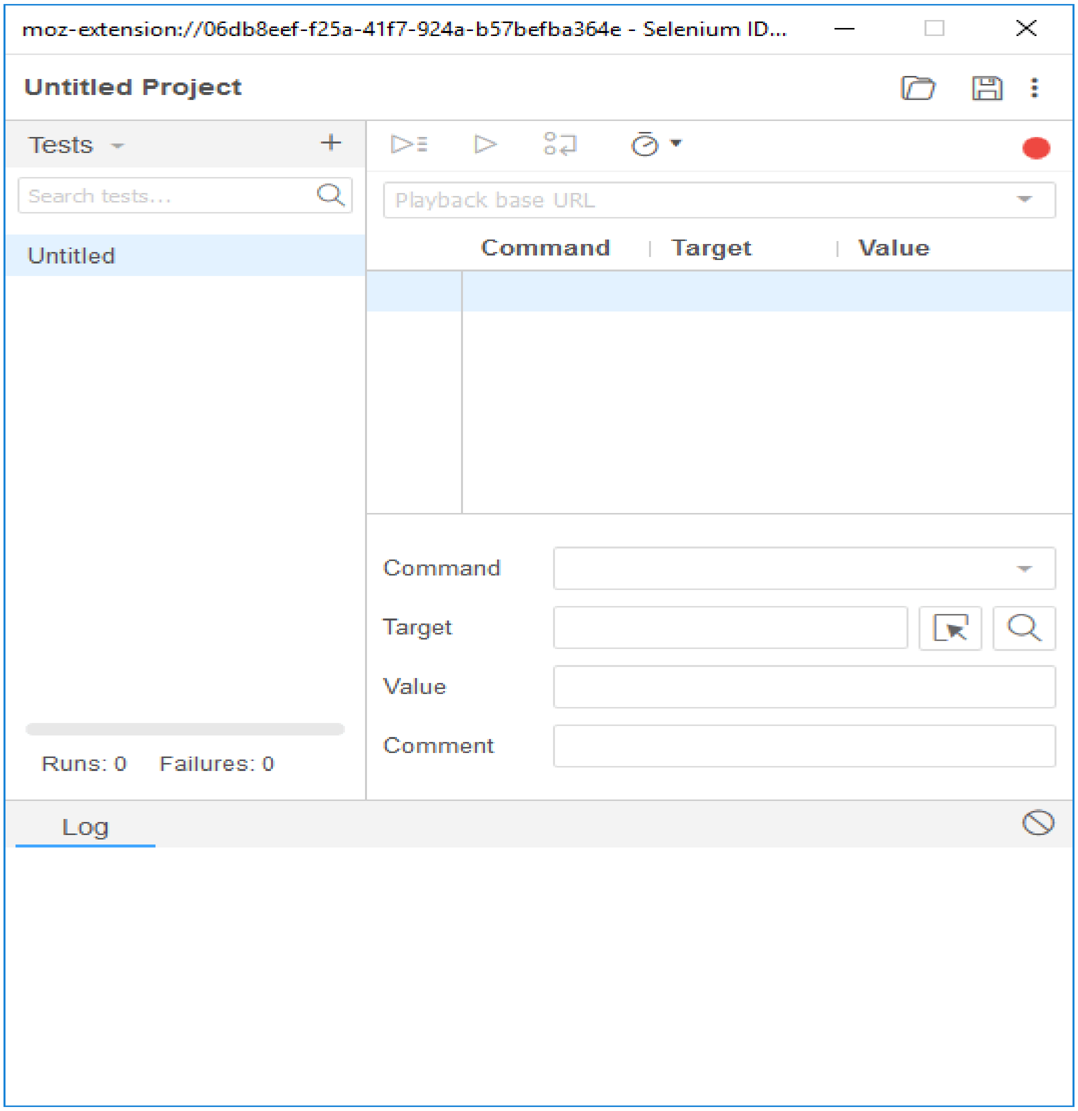
<u>Welcome Screen</u>

Upon launching the IDE you will be presented with a welcome dialog.

This will give you quick access to the following options:

- Record a new test in a new project
- Open an existing project
- Create a new project
- Close the IDE

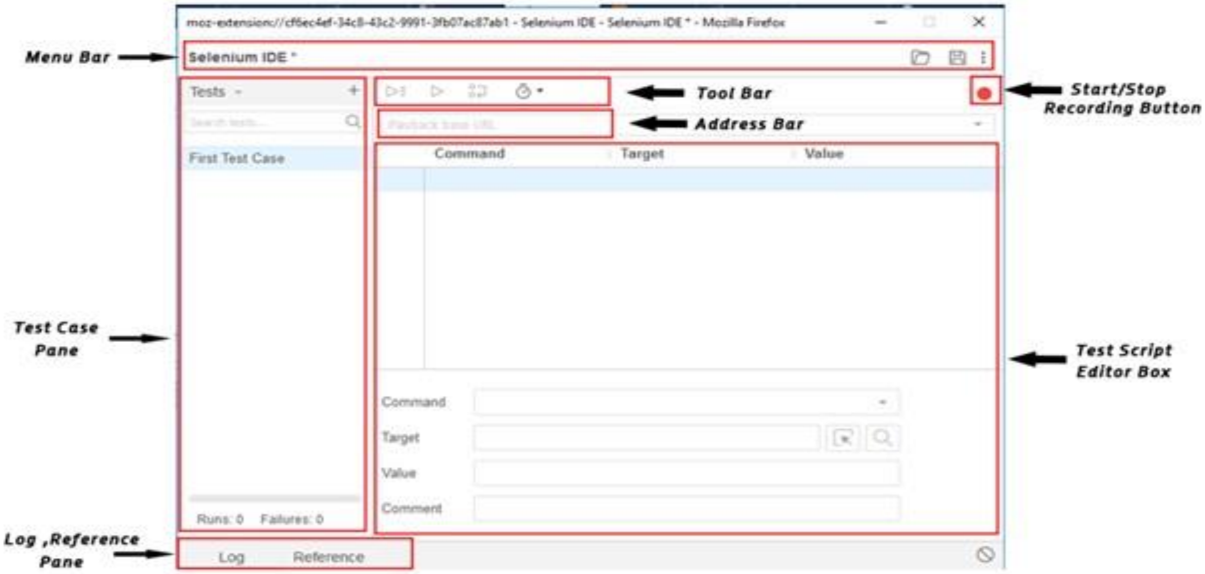
If this is your first time using the IDE (or you are starting a new project), then select the first option.



Selenium IDE-Features

Selenium IDE is divided into different components, each having their own features and functionalities.We have categorized seven different components of Selenium IDE, which includes:

1. Menu Bar
2. Tool Bar
3. Address Bar
4. Test Case Pane
5. Test Script Editor Box
6. Start/Stop Recording Button
7. Log, Reference Pane



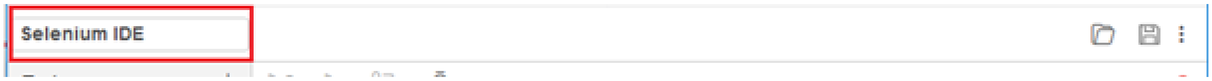
Now, we will look at the features and functionalities of each component in detail.

1. Menu Bar

Menu bar is positioned at the top most portion of the Selenium IDE interface. The most commonly used modules of menu bar include:

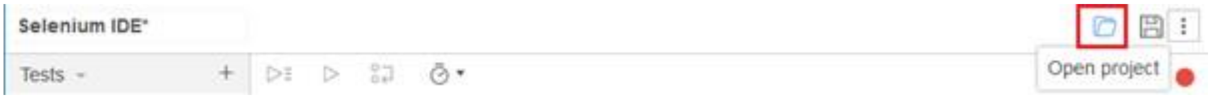
- Project Name

  It allows you to rename your entire project.



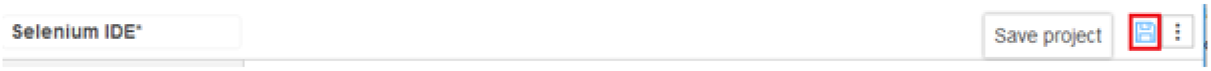
- Open Project

  It allows you to load any existing project from your personal drives.



- Save Project

  It allows you to save the entire project you are currently working on.

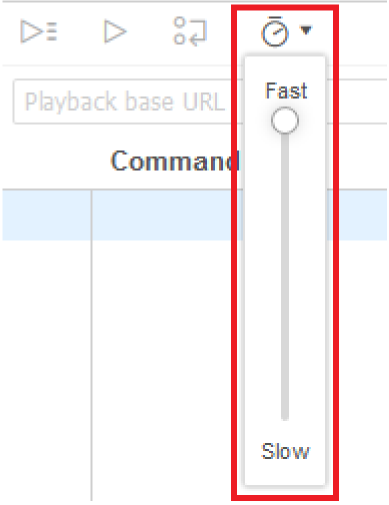


2. Tool Bar

The Tool bar contains modules for controlling the execution of your test cases. In addition, it gives you a step feature for debugging you test cases. The most commonly used modules of Tool Bar menu include:

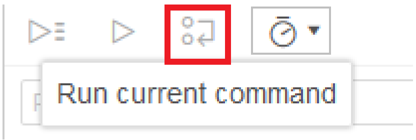
- Speed Control Option

  It allows you to control the execution speed of your test cases.



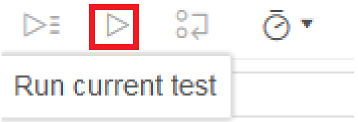
- Step Feature

  It allows you to "step" through a test case by running it one command at a time. Use for debugging test cases.



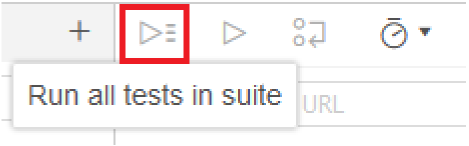
- Run Tests

  It allows you to run the currently selected test. When only a single test is loaded "Run Test" button and "Run all" button have the same effect.



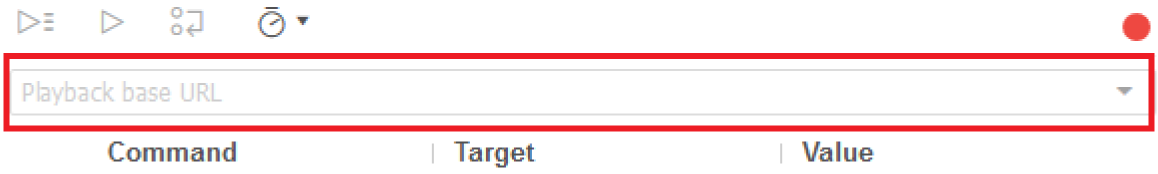
- Run All

  It allows you to run the entire test suite when a test suite with multiple test cases is loaded.



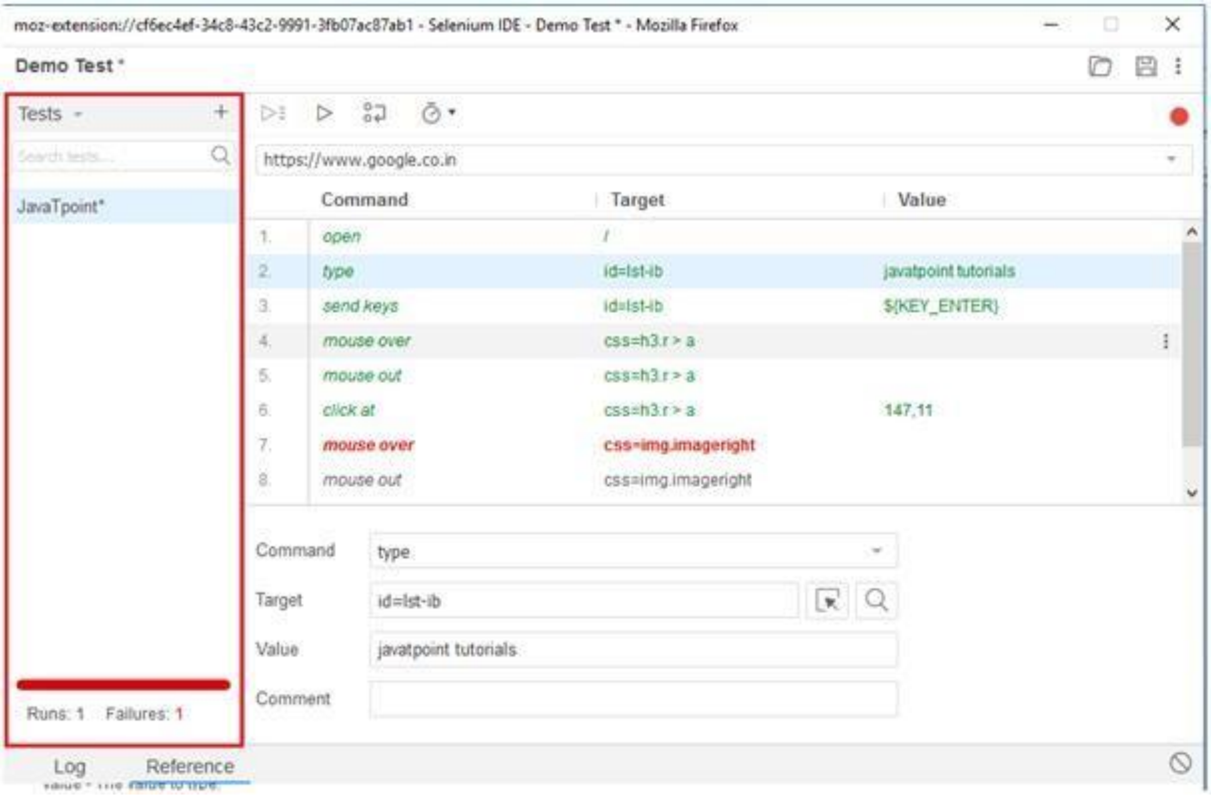
3. Address Bar

This module provides you a dropdown menu that remembers all previous values for base URL. In simple words, the base URL address bar remembers the previously visited websites so that the navigation becomes easy later on.



4. Test Case Pane

This module contains all the test cases that are recorded by IDE. In simple words, it provides the list of all recorded test cases at the same time under the test case pane so that user could easily shuffle between the test cases.



moz-extension://cf6ec4ef-34c8-43c2-9991-3fb07ac87ab1 - Selenium IDE - Demo Test * - Mozilla Firefox

Demo Test *

https://www.google.co.in

	Command	Target	Value
1.	open	/	
2.	type	id=lst-ib	javatpoint tutorials
3.	send keys	id=lst-ib	${KEY_ENTER}
4.	mouse over	css=h3.r > a	
5.	mouse out	css=h3.r > a	
6.	click at	css=h3.r > a	147,11
7.	mouse over	css=img.imageright	
8.	mouse out	css=img.imageright	

Command: type

Target: id=lst-ib

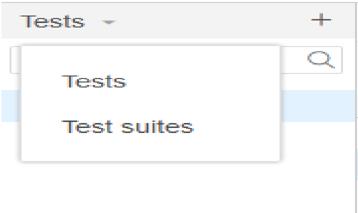
Value: javatpoint tutorials

Comment:

Runs: 1 Failures: 1

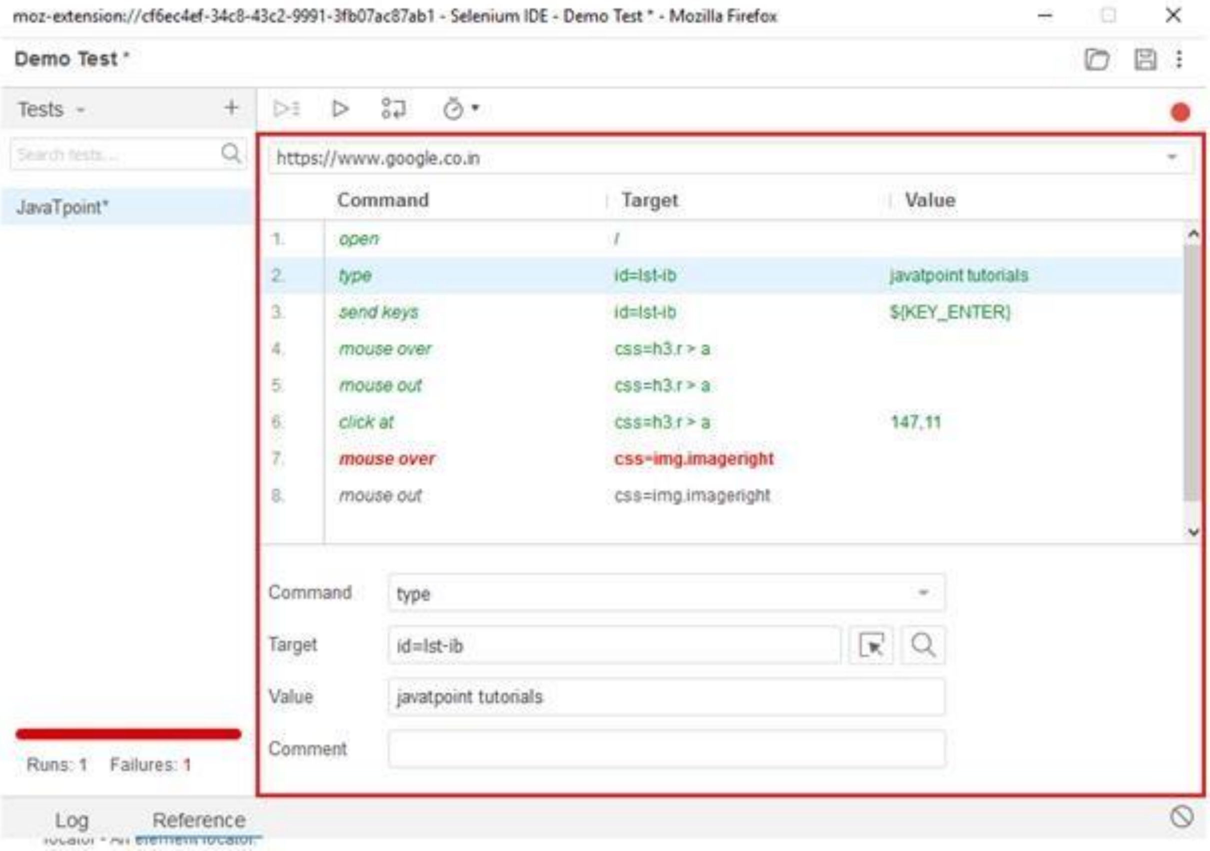
At the bottom portion of the Test Case Pane, you can see the test execution result summary which includes the pass/fail status of various test cases.

Test Case Pane also includes features like Navigation panel which allow users to navigate between test cases and test suites.



5. Test Script Editor Box

Test Script Editor Box displaysall of the test scripts and user interactions that were recorded by the IDE. Each user interaction is displayed in the same order in which they are performed. The Editor box is divided into three columns:Command, Target and Value.



moz-extension://cf6ec4ef-34c8-43c2-9991-3fb07ac87ab1 - Selenium IDE - Demo Test * - Mozilla Firefox
Demo Test *
Tests
Search tests...
JavaTpoint*
Runs: 1 Failures: 1
Log
Reference

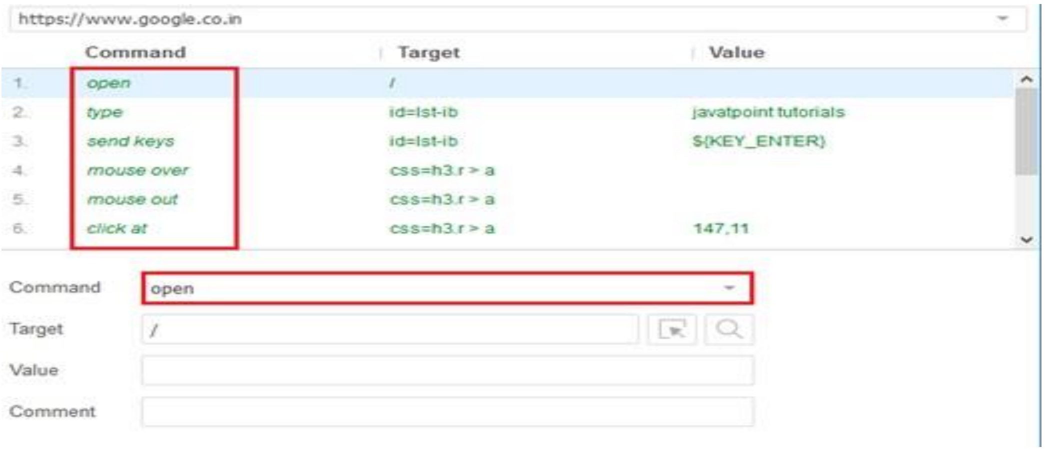
https://www.google.co.in

	Command	Target	Value
1.	open	/	
2.	type	id=lst-ib	javatpoint tutorials
3.	send keys	id=lst-ib	${KEY_ENTER}
4.	mouse over	css=h3.r > a	
5.	mouse out	css=h3.r > a	
6.	click at	css=h3.r > a	147,11
7.	mouse over	css=img.imageright	
8.	mouse out	css=img.imageright	

Command: type
Target: id=lst-ib
Value: javatpoint tutorials
Comment:

- Command:

  Command can be considered as the actual operation/action that is performed on the browser elements. For instance, if you are opening a new URL, the command will be 'open'; if you are clicking on a link or a button on the web page, then the command will be 'clicked'.



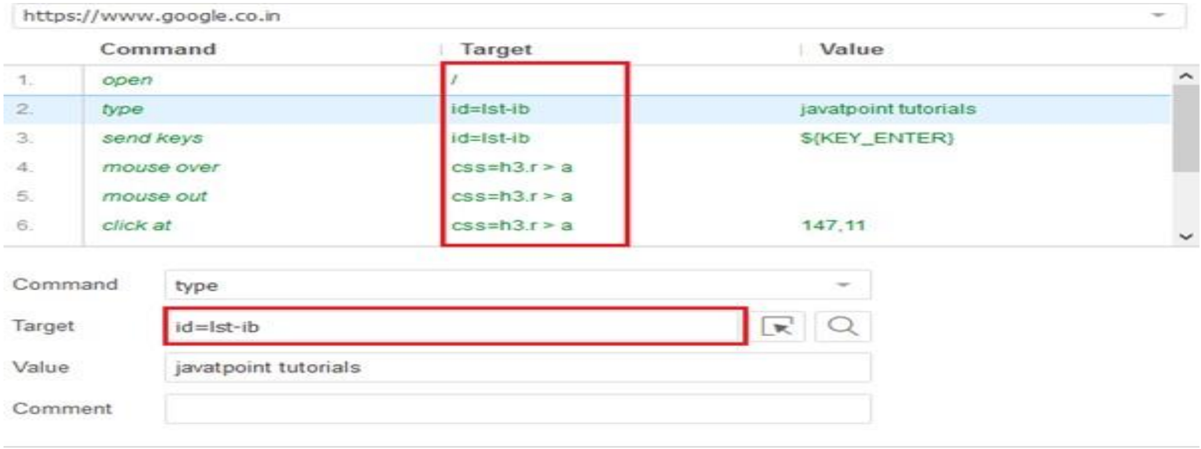
https://www.google.co.in

	Command	Target	Value
1.	open	/	
2.	type	id=lst-ib	javatpoint tutorials
3.	send keys	id=lst-ib	${KEY_ENTER}
4.	mouse over	css=h3.r > a	
5.	mouse out	css=h3.r > a	
6.	click at	css=h3.r > a	147,11

Command: open
Target: /
Value:
Comment:

- Target:

  Target specifies the web element on which the operation has to be performed along with a locator attribute. For instance, if you are clicking on a button called javaTpoint, then the target link will be 'javaTpoint'.



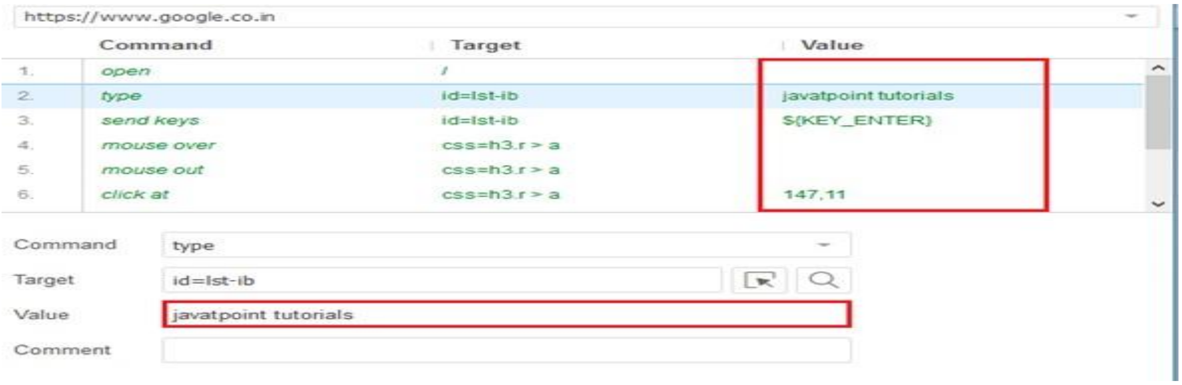
https://www.google.co.in

	Command	Target	Value
1.	open	/	
2.	type	id=lst-ib	javatpoint tutorials
3.	send keys	id=lst-ib	${KEY_ENTER}
4.	mouse over	css=h3.r > a	
5.	mouse out	css=h3.r > a	
6.	click at	css=h3.r > a	147,11

Command	type
Target	id=lst-ib
Value	javatpoint tutorials
Comment	

- Value:

  Value is treated as an optional field and can be used when we need to send some actual parameters. For instance, if you are entering the email address or password in a textbox, then the value will contain the actual credentials.



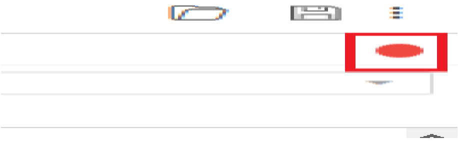
https://www.google.co.in

	Command	Target	Value
1.	open	/	
2.	type	id=lst-ib	javatpoint tutorials
3.	send keys	id=lst-ib	${KEY_ENTER}
4.	mouse over	css=h3.r > a	
5.	mouse out	css=h3.r > a	
6.	click at	css=h3.r > a	147,11

Command	type
Target	id=lst-ib
Value	javatpoint tutorials
Comment	

6. Start/Stop Recording Button

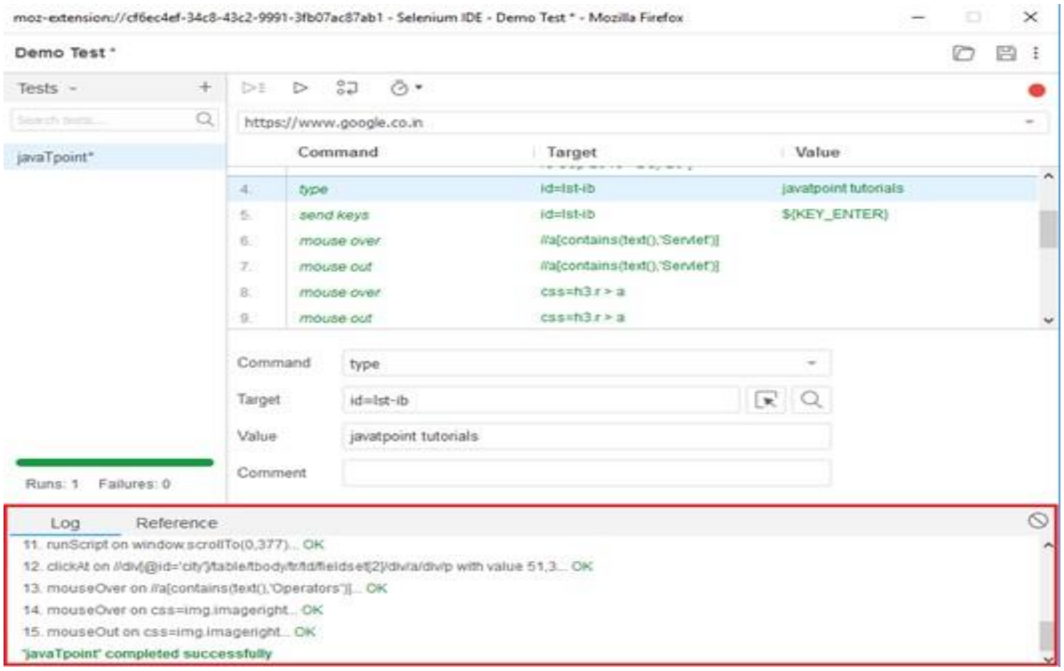
Record button records all of the user actions with the browser.



7. Log, Reference Pane

The Log Pane displays the runtime messages during execution. It provides real-time updates of the actions performed by the IDE. It can be categorized into four types: info, error, debug and warn.

The reference Pane displays the complete detail of the currently selected selenese command in the editor.

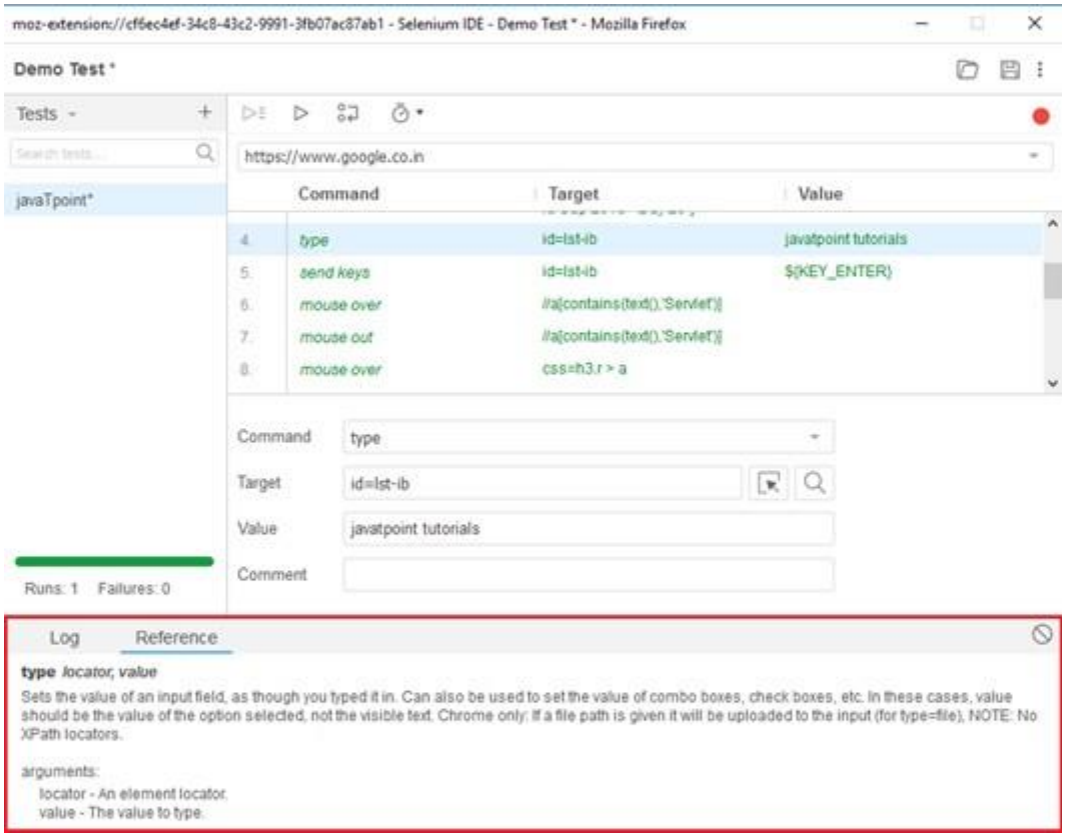

moz-extension://cf6ec4ef-34c8-43c2-9991-3fb07ac87ab1 - Selenium IDE - Demo Test * - Mozilla Firefox
Demo Test *
Tests
javaTpoint*
https://www.google.co.in

	Command	Target	Value
4.	type	id=lst-ib	javatpoint tutorials
5.	send keys	id=lst-ib	${KEY_ENTER}
6.	mouse over	//a[contains(text(),'Servlet')]	
7.	mouse out	//a[contains(text(),'Servlet')]	
8.	mouse over	css=h3.r > a	
9.	mouse out	css=h3.r > a	

Command: type
Target: id=lst-ib
Value: javatpoint tutorials
Comment:

Runs: 1 Failures: 0

```
11. runScript on window.scrollTo(0,377)... OK
12. clickAt on //div[@id='city']/table/tbody/tr/td/fieldset[2]/div/a/div/p with value 51,3... OK
13. mouseOver on //a[contains(text(),'Operators')]... OK
14. mouseOver on css=img.imageright... OK
15. mouseOut on css=img.imageright... OK
'javaTpoint' completed successfully
```


moz-extension://cf6ec4ef-34c8-43c2-9991-3fb07ac87ab1 - Selenium IDE - Demo Test * - Mozilla Firefox
Demo Test *
Tests
javaTpoint*
https://www.google.co.in

	Command	Target	Value
4.	type	id=lst-ib	javatpoint tutorials
5.	send keys	id=lst-ib	${KEY_ENTER}
6.	mouse over	//a[contains(text(),'Servlet')]	
7.	mouse out	//a[contains(text(),'Servlet')]	
8.	mouse over	css=h3.r > a	

Command: type
Target: id=lst-ib
Value: javatpoint tutorials
Comment:

Runs: 1 Failures: 0

type *locator, value*

Sets the value of an input field, as though you typed it in. Can also be used to set the value of combo boxes, check boxes, etc. In these cases, value should be the value of the option selected, not the visible text. Chrome only: If a file path is given it will be uploaded to the input (for type=file). NOTE: No XPath locators.

arguments:

- locator - An element locator.
- value - The value to type.

Program 1:

AIM: Recording in context sensitive mode and analog mode using Selenium IDE

Objective:

- ✓ Student should be able to
- ✓ Describes Context Sensitive mode
- ✓ Record a test script
- ✓ Read the test script
- ✓ Run the recorded test and analyze the results

Context Sensitive Recording mode

- It is the default mode of recording which takes full advantage of Quick Test Professional's test object model.
- It recognizes objects in the application regardless of their location on the screen.
- It records the operations that are preformed in an application by identifying the GUI objects.

Analog recording

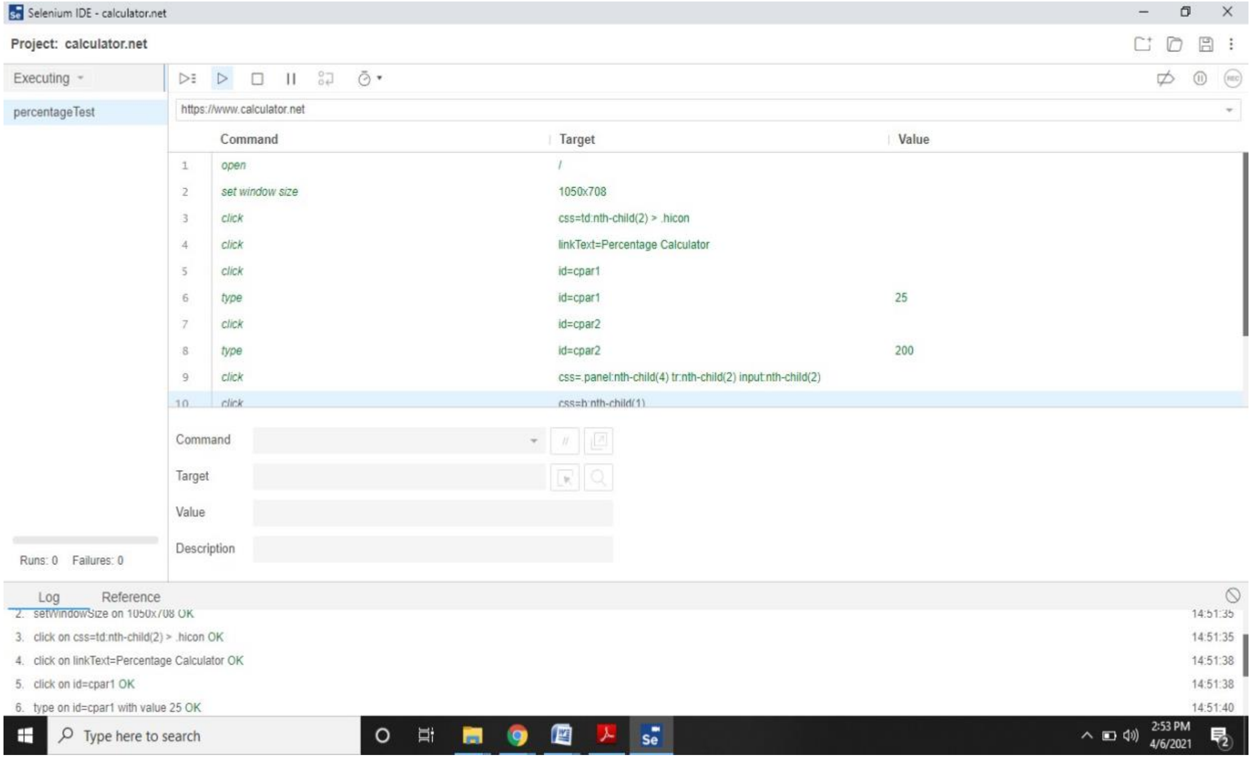
- records the inputs from keyboard, mouse clicks, the x and y coordinates that are travelled by the mouse pointer across the screen.
- It can't capture GUI windows and objects.

Procedure:

(I) Recording (recording user interactions with the browser)

- ➢ Step1:Name the project as "Demo Test".
- ➢ Step2: Name the test case as "Demo1 test".
- ➢ Step 3: Click on the "Start Recording" Button present on the top right corner on the IDE to start recording the test case.
- ➢ Step 4: Type https://www.calculator.net in the search box.
- ➢ Step 5: Hit enter to go to home page

- Step 6: It will redirect you to the that webpage and what operations you perform will be recorded. Meanwhile, you will get the notifications of the actions performed by the IDE at the extreme right corner of your web browser.
- Step 7: Now, go the IDE and click on the "Stop Recording" button to stop recording your actions further.

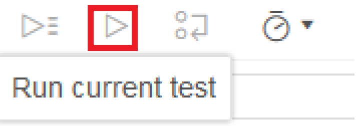


	Command	Target	Value
1	open	/	
2	set window size	1050x708	
3	click	css=td:nth-child(2) > .hicon	
4	click	linkText=Percentage Calculator	
5	click	id=cpar1	
6	type	id=cpar1	25
7	click	id=cpar2	
8	type	id=cpar2	200
9	click	css=.panel:nth-child(4) tr:nth-child(2) input:nth-child(2)	
10	click	css=b:nth-child(1)	

```
2. setWindowSize on 1050x708 OK                          14:51:35
3. click on css=td:nth-child(2) > .hicon OK              14:51:35
4. click on linkText=Percentage Calculator OK            14:51:38
5. click on id=cpar1 OK                                  14:51:38
6. type on id=cpar1 with value 25 OK                     14:51:40
```

(II) Playing back (executing the recorded script)

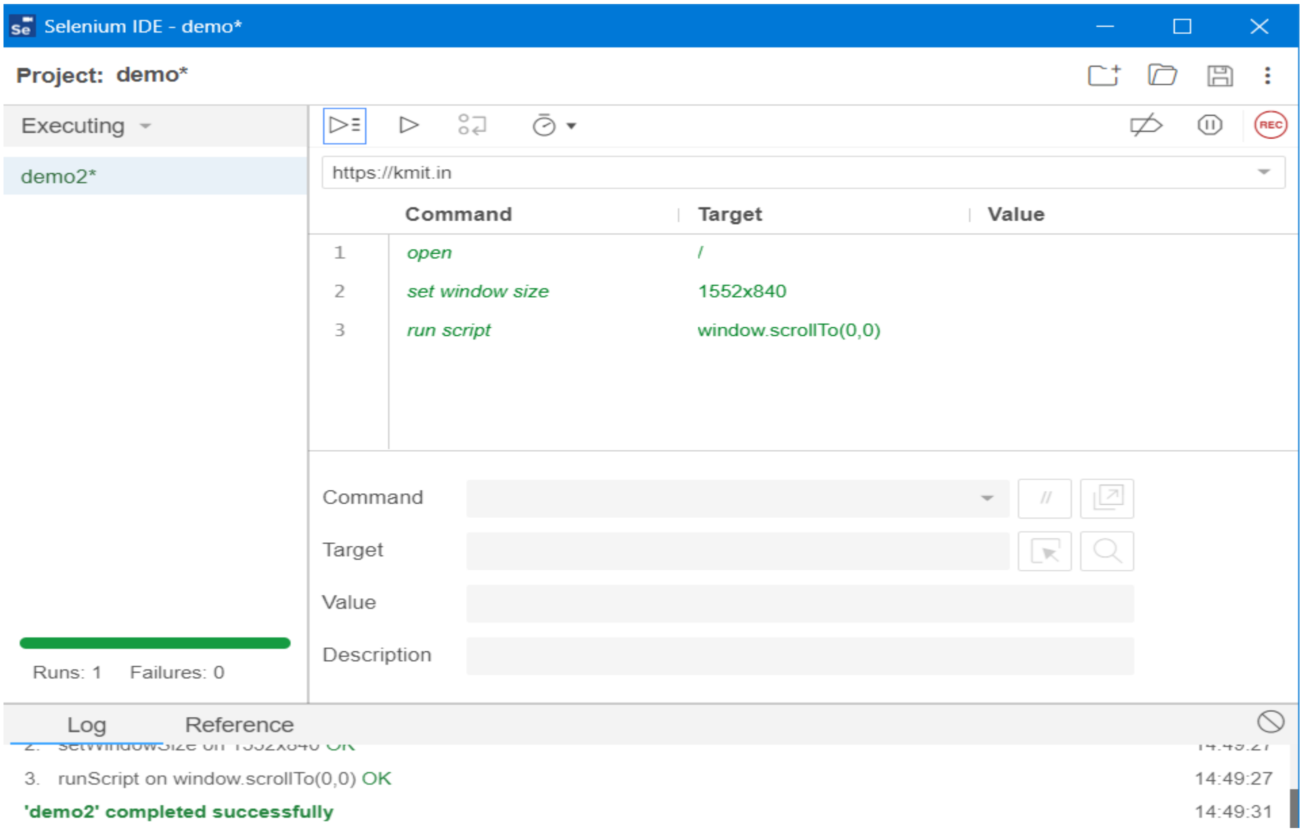
- Click on the "Run Current Test" button present on the tool bar menu of the IDE.
  It will execute all of your interactions with the browser and gives you an overall summary of the executed test script.
- Run Tests
  It allows you to run the currently selected test. When only a single test is loaded "Run Test" button and "Run all" button have the same effect.



(III) Saving the test suite

- Click on the save button present on the extreme right corner of the menu bar.
- save the entire test suite as "Demo Test".

Output:



	Command	Target	Value
1	open	/	
2	set window size	1552x840	
3	run script	window.scrollTo(0,0)	

```
2.  setWindowSize on 1552x840 OK                                14:49:27
3.  runScript on window.scrollTo(0,0) OK                        14:49:27
'demo2' completed successfully                                  14:49:31
```

Program 2:

AIM: GUI checkpoint for single property

Objective: Student should be able to

- ✓ Explain how to check the behavior of GUI objects
- ✓ Create a test that checks GUI objects
- ✓ Run the test on different versions of an application and examine the results

Graphical User Interface:

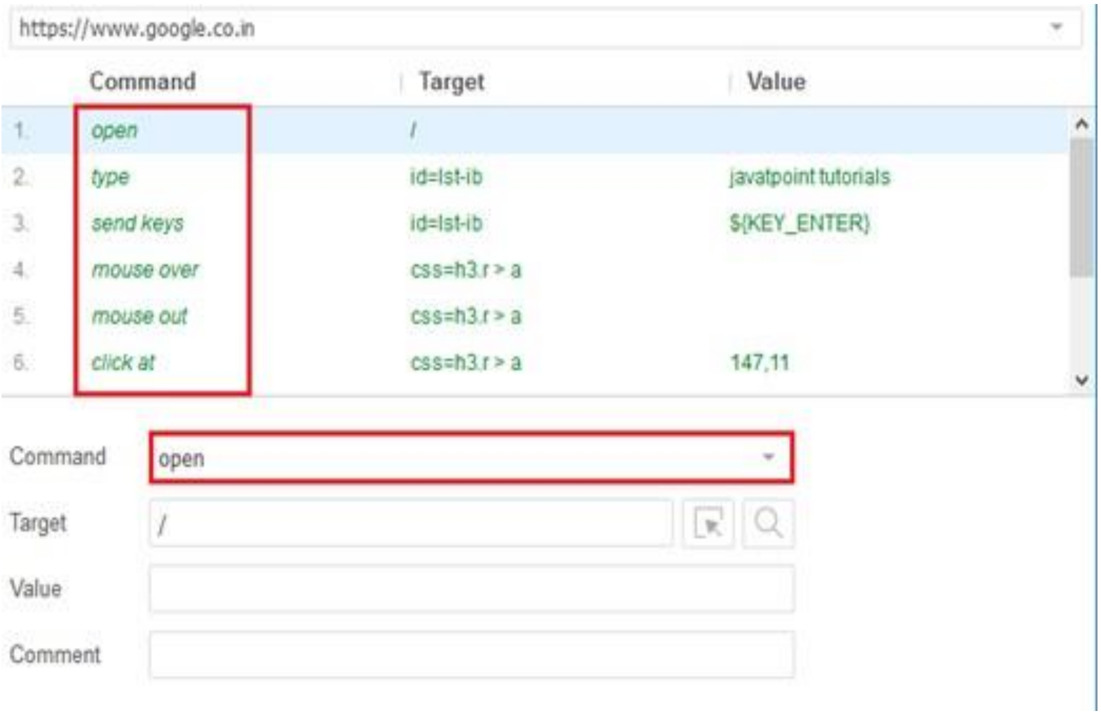
- GUI testing evaluates design elements such as layout, colors, fonts, font sizes, labels, text boxes, text formatting, captions, buttons, lists, icons, links and content.
- GUI testing processes can be either done by the two categories are manual or automatic, and are often performed by third -party companies, rather than developers or end users.
- It can be require a lot of programming and is time consuming whether manual or automatic.
- Usually the software author writes out the intended function of a menu or graphical button.
- GUI testing also tends to test for certain program behaviors that users expect, like an hourglass when the program is busy.

Procedure:

- ➢ Step 1: open selenium IDE and open a new test
- ➢ Step 2: Name the project as "Demo Test".
- ➢ Step 3: Name the test case as "singleproperty_test".
- ➢ Step 4: Click on the "Start Recording" Button present on the top right corner on the IDE to start recording the test case.
- ➢ It will redirect you to the Google search engine page.
- ➢ Step 5: Type "calculator.com" in the Google search box.
- ➢ Step 6: Hit enter to get the search results.Click on the link provided under the URLhttps://www.calculator.com
- ➢ Step 7: Perform certain operation for single property
- ➢ Step 8:After recording the single click then stop the recording

- Step 9: Click on the "Run Current Test" button present on the tool bar menu of the IDE. It will execute all of your interactions with the browser and gives you an overall summary of the executed test script.
- Step 10: give command and target to perform test and value as webpage address
- Step 11: create GUI checkpoint for the click /enter of value to perform operation.
- Step 12: will now check in log and reference about the GUI checkpoint and whether the test is successful or failure.

Output:



https://www.google.co.in

	Command	Target	Value
1.	open	/	
2.	type	id=lst-ib	javatpoint tutorials
3.	send keys	id=lst-ib	${KEY_ENTER}
4.	mouse over	css=h3.r > a	
5.	mouse out	css=h3.r > a	
6.	click at	css=h3.r > a	147,11

Command: open

Target: /

Value:

Comment:

Program 3:

AIM: GUI checkpoint for single object/window

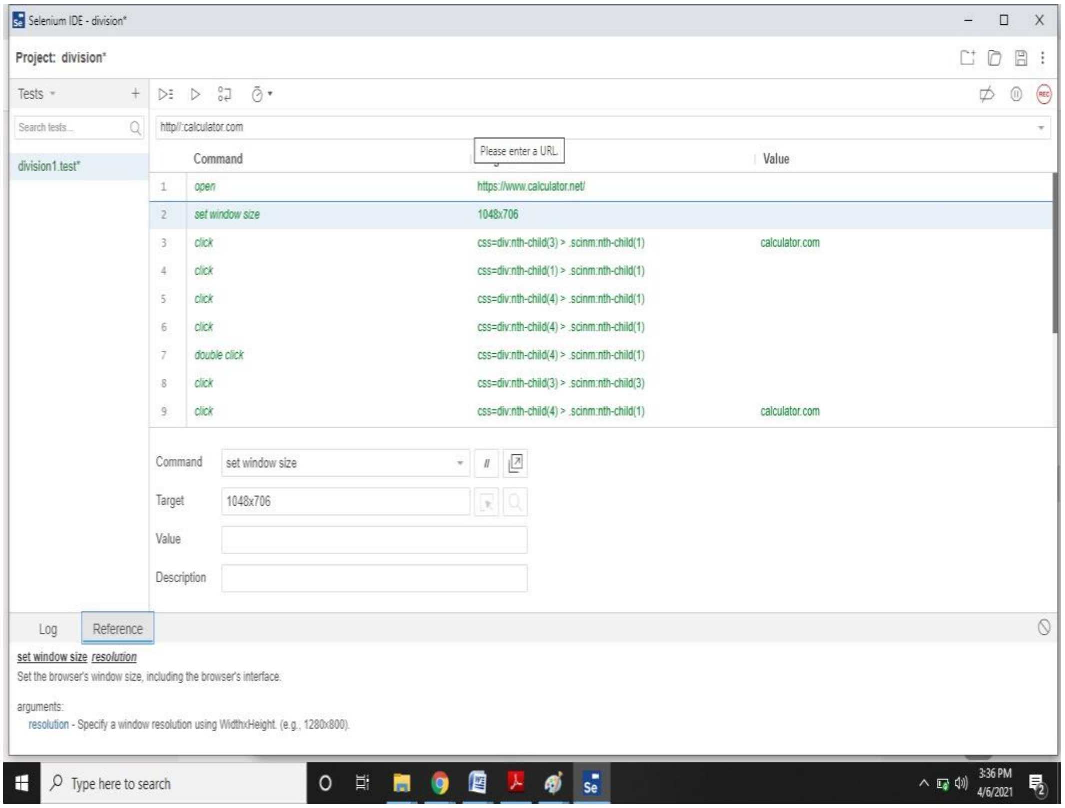
Objectives: Student should be able to

- ✓ Explain how to check the behavior of GUI objects
- ✓ Create a test that checks GUI objects/window
- ✓ Run the test on different versions of an application and examine the results

Procedure:

- ➢ Step 1: open selenium IDE and open a new test
- ➢ Step 2: Name the project as "Demo Test".
- ➢ Step 3: Name the test case as "singlewindow_test".
- ➢ Step 4: Click on the "Start Recording" Button present on the top right corner on the IDE to start recording the test case.
- ➢ It will redirect you to the Google search engine page.
- ➢ Step 5: Type "calculator.com" in the Google search box.
- ➢ Step 6: Hit enter to get the search results Click on the link provided under the URLhttps://www.calculator.com
- ➢ Step 7:Perform certain operation for window/objects
- ➢ Step 8: After recording the single click then stop the recording
- ➢ Step 9:Click on the "Run Current Test" button present on the tool bar menu of the IDE.
  It will execute all of your interactions with the browser and gives you an overall summary of the executed test script.
- ➢ Step 10:give command and target to perform test and value as webpage address
- ➢ Step 11: will now check in log and reference about the GUI checkpoint and whether the test is successful or failure and GUI checkpoint of entry of value in the checkbox of object and a window.

Output:



http//:calculator.com

	Command		Value
1	open	https://www.calculator.net/	
2	set window size	1048x706	
3	click	css=div:nth-child(3) > .scinm:nth-child(1)	calculator.com
4	click	css=div:nth-child(1) > .scinm:nth-child(1)	
5	click	css=div:nth-child(4) > .scinm:nth-child(1)	
6	click	css=div:nth-child(4) > .scinm:nth-child(1)	
7	double click	css=div:nth-child(4) > .scinm:nth-child(1)	
8	click	css=div:nth-child(3) > .scinm:nth-child(3)	
9	click	css=div:nth-child(4) > .scinm:nth-child(1)	calculator.com

Command: set window size

Target: 1048x706

Value:

Description:

set window size *resolution*

Set the browser's window size, including the browser's interface.

arguments:

resolution - Specify a window resolution using WidthxHeight. (e.g., 1280x800).

Program :4

AIM: GUI checkpoint for single multiple objects

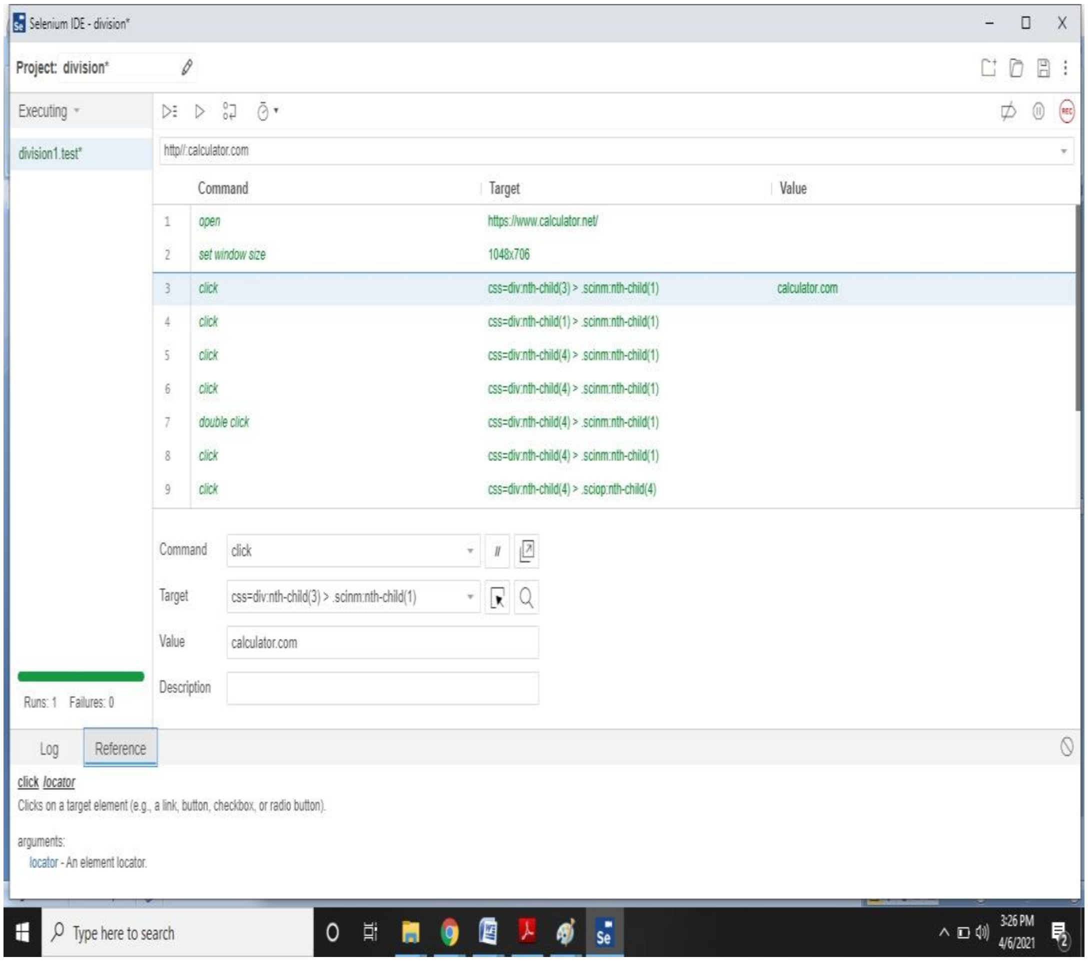
Objective: Student should be able to

- ✓ Explain how to check the behavior of GUI objects
- ✓ Create a test that checks GUI multiple objects
- ✓ Run the test on different versions of an application and examine the results

Procedure:

- ➢ Step 1:open selenium IDE and open a new test
- ➢ Step 2:Name the project as "Demo Test".
- ➢ Step 3: Name the test case as "multipleproperty_test".
- ➢ Step 4: Click on the "Start Recording" Button present on the top right corner on the IDE to start recording the test case.
- ➢ It will redirect you to the Google search engine page.
- ➢ Step 5: Type "calculator.com" in the Google search box.
- ➢ Step 6: Hit enter to get the search results.Click on the link provided under the URLhttps://www.calculator.com
- ➢ Step 7: Perform certain operation for Multiple property
- ➢ Step 8: After recoeding the single click then stop the recording
- ➢ Step 9: Click on the "Run Current Test" button present on the tool bar menu of the IDE.
  It will execute all of your interactions with the browser and gives you an overall summary of the executed test script.
- ➢ Step 10: Give command and target to perform test and value as webpage address
- ➢ Step 11: if the click GUI checkpoint on multiple objects or navigating from one window to another is noted in reference block and in log section to check whether the test is successful or a failure.

Output:



http//:calculator.com

	Command	Target	Value
1	open	https://www.calculator.net/	
2	set window size	1048x706	
3	click	css=div:nth-child(3) > .scinm:nth-child(1)	calculator.com
4	click	css=div:nth-child(1) > .scinm:nth-child(1)	
5	click	css=div:nth-child(4) > .scinm:nth-child(1)	
6	click	css=div:nth-child(4) > .scinm:nth-child(1)	
7	double click	css=div:nth-child(4) > .scinm:nth-child(1)	
8	click	css=div:nth-child(4) > .scinm:nth-child(1)	
9	click	css=div:nth-child(4) > .sciop:nth-child(4)	

Command: click

Target: css=div:nth-child(3) > .scinm:nth-child(1)

Value: calculator.com

Description:

Runs: 1 Failures: 0

click *locator*

Clicks on a target element (e.g., a link, button, checkbox, or radio button).

arguments:

locator - An element locator.

Program 5:

AIM: Test case for calculator in windows application

Objectives: Students should be able to

- ✓ Know about what is Manual testing
- ✓ Able to write Test cases.

- **Manual Testing:** Manual testing is a testing process that is carried out manually in order to find defects without the usage of tools or automation scripting. A test plan document is prepared that acts as a guide to the testing process in order to have the complete test coverage.

Procedure:

Write the test cases based on the following functions and scenarios.

- Check the calculator if it starts by on button. If it is a software-based calculator, then check if it begins via specific means like from searching for a calculator in a search bar and then executing an application. Or by accessing menu items in Windows.
- Check if the calculator window maximizes to the specified window size.
- Check if the calculator closes when the close button is pressed or if the exit menu is clicked from the file > exit option.
- Check if the help document is accessed from Help > Documentation.
- Check if the calculator allows copy and paste functionality.
- Check if the calculator has any specific preferences.
- Check if all the numbers are working (0 to 9)
- Check if the arithmetic keys (+, -, *, %, /) are working.
- Check if the clear key is working.
- Check if the brackets keys are working.
- Check if the sum or corresponding key is working.
- Check if the square and square root keys are working.

Output:

Test Case:

Test scenario ID	Login-1		**Test case ID**	Login-1A		
Test case description	User login – Positive case		**Test case priority**	High		
Pre-requisite	User email id		**Post-requisite**	Username, password		
S.no	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Result**	**Test Comments**
1.	Launch Calculator App	Search in search bar and click on Calculator icon	Calculator homepage	Calculator homepage	Pass	Launch Successful
2.	Testing Maximize button	Click on maximize button()	Maximized window size	Maximized window size	Pass	Worked Successfully
3.	Close Application	Click on close(X)	Calculator closed	Calculator closed	Pass	Close Successful
4.	Testing Help option	Click on “help”	Help documentation	Help documentation	Pass	Worked Successfully
5.	Copy & Paste Functionality	Copy Math expression paste in app	Expression copied	Expression copied	Pass	Worked Successfully
6.	Working of (0-9) keys	Click on any number say 8	Number in result area	Number in result area	Pass	Worked Successfully
7.	Working of arithmetic keys	Click on any arithmetic key say ‘+’ As 2+3	2+3=5	2+3=5	Pass	Worked Successfully
8.	Testing “Clear” key	Click on c or clear key	Result area 0	Result area 0	Pass	Worked Successfully
9.	Testing Bracket keys	Type any expression which includes brackets say 2*(3+2)	2*(3+2)=10	2*(3+2)=10	Pass	Worked Successfully
10.	Testing square & square Root keys	Click on any number then click Square/SquareRoot	Sqr(9)=81	Sqr(9)=81	Pass	Worked Successfully