Experiment 5

Student Name: Rahul Saxena

UID: 24MCI10204

Branch: MCA (AI-ML)

Section/Group: MAM - 3(B)

Semester: II

Subject Name: Software Testing [24CAH-654]

Aim: Using Selenium IDE, create automated test scripts for any of the web applications, execute them, and analyze the results.

Definition: Selenium is an open-source framework used for automating web browsers, enabling developers and testers to write scripts in various languages to test web applications across different browsers and platforms.

Steps:

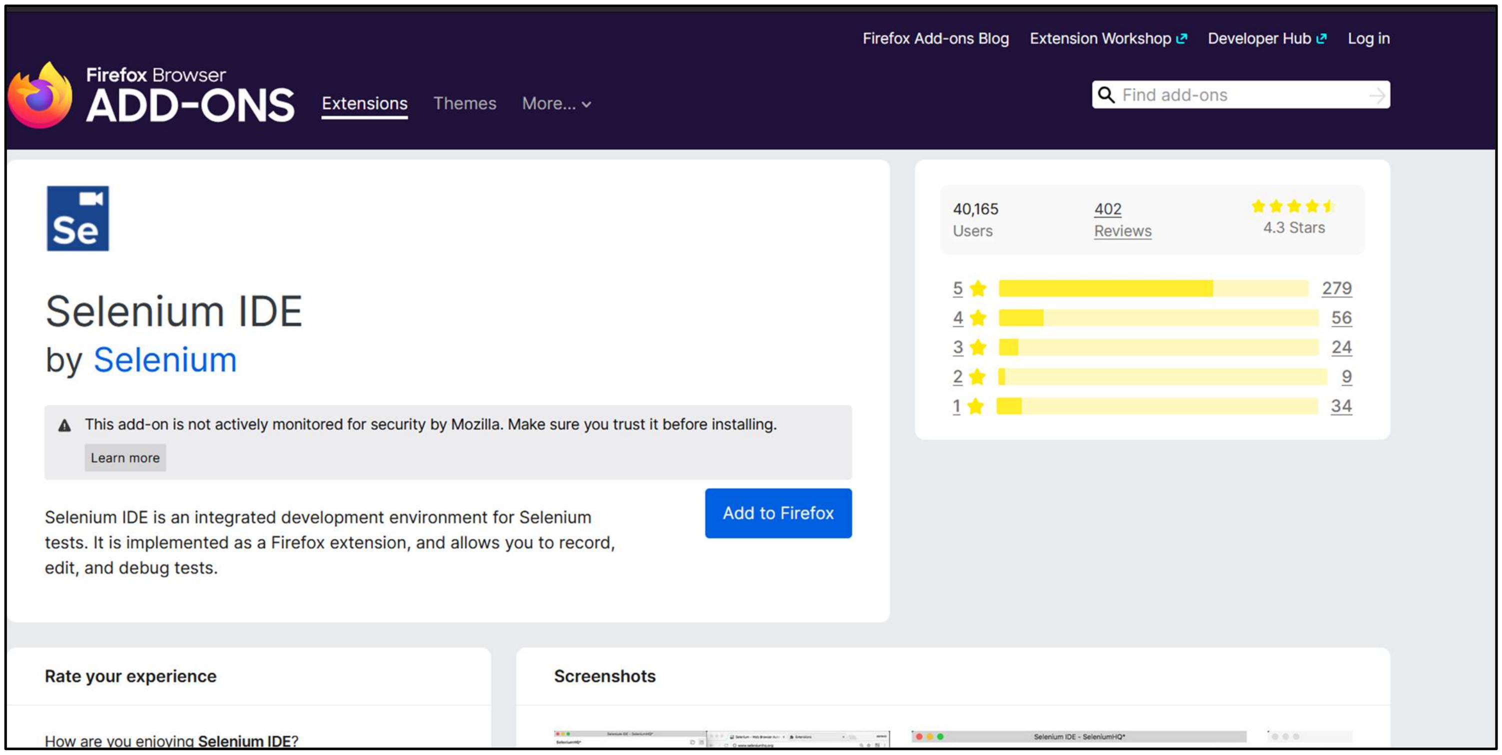
- Step 1: Install Selenium IDE Extension
  - Open Google Chrome or Mozilla Firefox.
  - Go to the Firefox Add-ons (for Firefox).
  - Search for "Selenium IDE".
  - Click "Add to Firefox and confirm the installation.
  - Once installed, you will see the Selenium IDE icon in the browser toolbar.
- Step 2: Open Selenium IDE
  - Click on the Selenium IDE icon in the browser toolbar.
  - The Selenium IDE window will open with options to create a new test case.
- Step 3: Create a New Project and Test Case
  - Click "Create a New Project" and name it (e.g., "Login Test").
  - Click on "Record a new test".
  - Enter the URL of the web application you want to test (e.g., https://example.com).
  - Click "Start Recording"—a new browser window opens where all actions will be recorded.
- Step 4: Record a Login Test
  - In the new browser window, enter a username and password in the login fields.
  - Click the Login button.
  - Wait for the page to load and verify successful login (e.g., check if the dashboard

appears).
- Stop the recording by clicking "Stop Recording" in Selenium IDE.
- Save the test case with a meaningful name (e.g., LoginTest).

- Step 5: Execute the Test Script
  - Click the "Run Current Test" button to execute the test.
  - Observe the execution—Selenium IDE will simulate user actions and attempt to log in.
  - If the test passes, a green checkmark appears next to each step.
  - If the test fails, a red cross appears, indicating errors.
- Step 6: Analyse the Test Results
  - Check the Log Panel at the bottom of Selenium IDE for test execution details.
  - View error messages (if any) and debug issues.
  - Modify test scripts if necessary (e.g., add waits for elements to load).
  - Re-run the test after making corrections.

Website Tested:
https://monkeytype.com/login

Selenium IDE: Install by clicking “Add to Firefox” Button.



Selenium IDE
by Selenium

This add-on is not actively monitored for security by Mozilla. Make sure you trust it before installing.

Learn more

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.

Add to Firefox

40,165 Users | 402 Reviews | 4.3 Stars

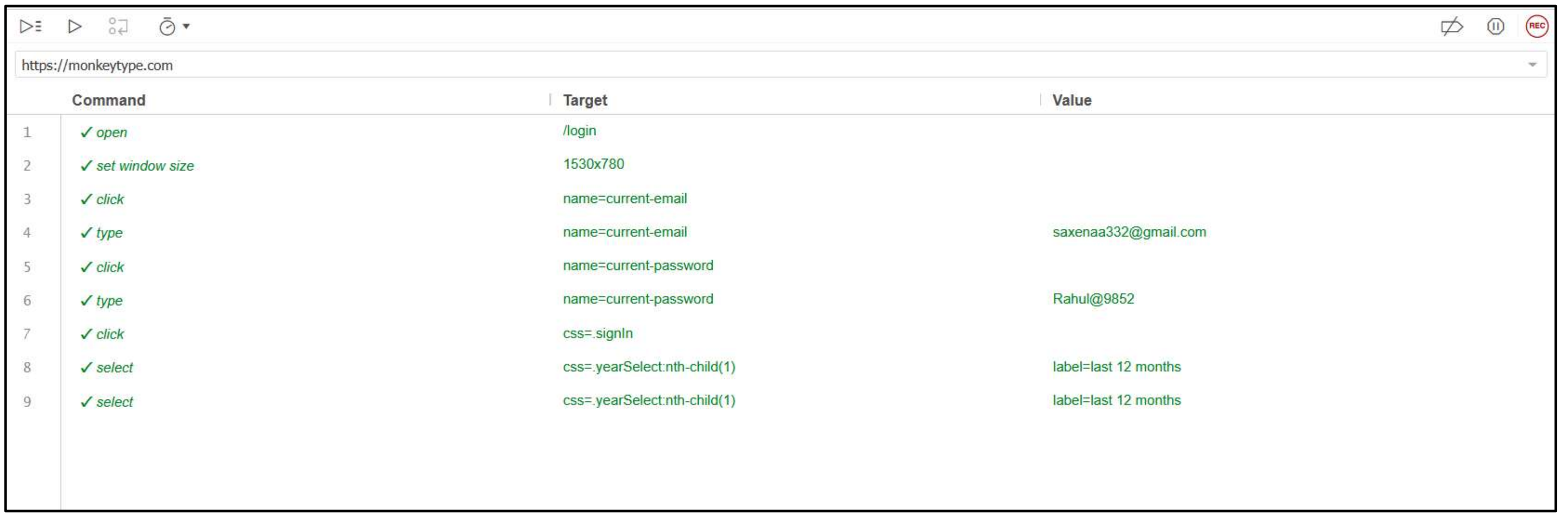
5	279
4	56
3	24
2	9
1	34

Rate your experience

How are you enjoying Selenium IDE?

Screenshots

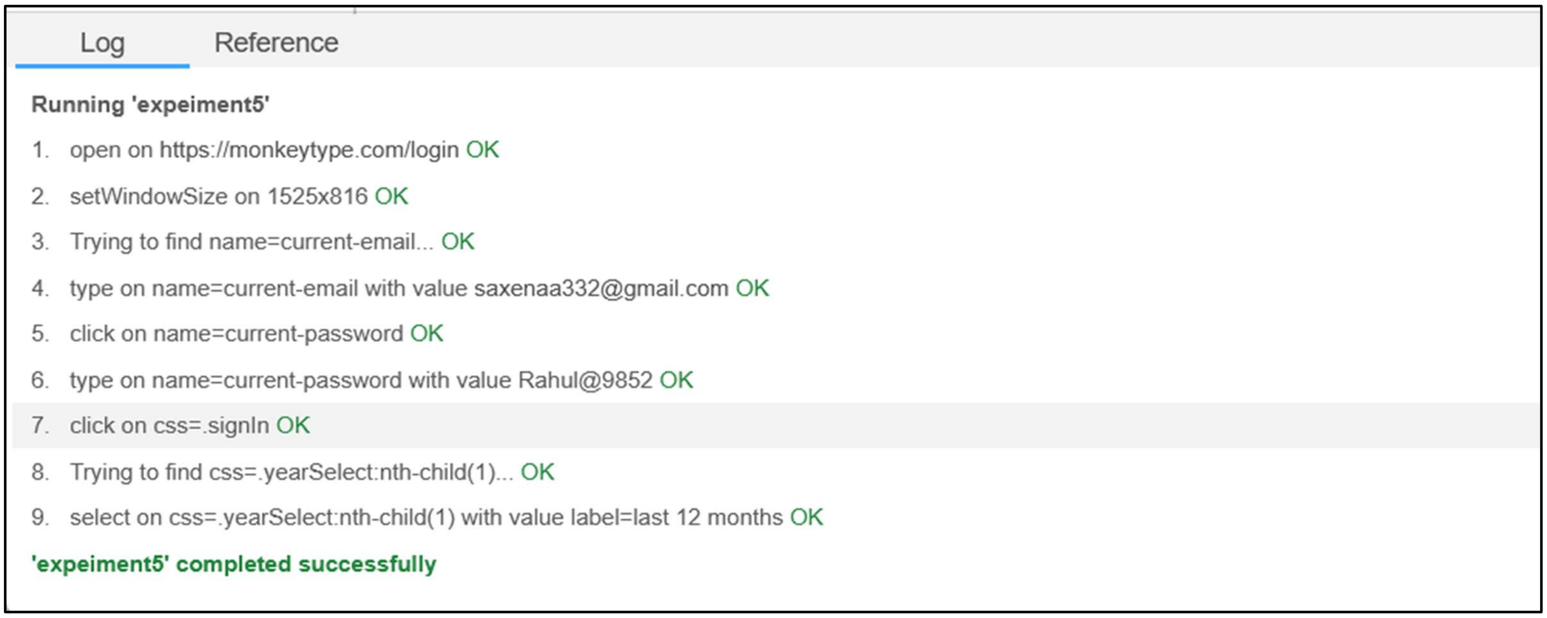
Test Script:



https://monkeytype.com

	Command	Target	Value
1	✓ open	/login	
2	✓ set window size	1530x780	
3	✓ click	name=current-email	
4	✓ type	name=current-email	saxenaa332@gmail.com
5	✓ click	name=current-password	
6	✓ type	name=current-password	Rahul@9852
7	✓ click	css=.signIn	
8	✓ select	css=.yearSelect:nth-child(1)	label=last 12 months
9	✓ select	css=.yearSelect:nth-child(1)	label=last 12 months

Test Log:



```
Running 'expeiment5'
1. open on https://monkeytype.com/login OK
2. setWindowSize on 1525x816 OK
3. Trying to find name=current-email... OK
4. type on name=current-email with value saxenaa332@gmail.com OK
5. click on name=current-password OK
6. type on name=current-password with value Rahul@9852 OK
7. click on css=.signIn OK
8. Trying to find css=.yearSelect:nth-child(1)... OK
9. select on css=.yearSelect:nth-child(1) with value label=last 12 months OK
'expeiment5' completed successfully
```

Learning Outcome:

- Writing Modular Code: Learned how to write reusable and modular methods for basic arithmetic operations.
- User Input Validation: Gained experience in handling invalid inputs gracefully to enhance user experience.
- Test-Driven Development: Developed and ran test cases to ensure the correctness of individual components.