Experiment 09: Design test cases and generate test scripts in Selenium

Learning Objective: Students will able to create unit test cases.

Tools: Selenium record and playback

Theory:

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation.

Write a program to calculate the square of a number in the range 1-100

```
#include <stdio.h>
int main()
{
int n, res;
printf("Enter a number: ");
scanf("%d", &n);
if (n >= 1 && n <= 100)
   {
res = n * n;  printf("\n Square of %d is%d\n", n, res);
   }
else if (n<= 0 || n > 100)     printf("Beyond the range");
   return 0;
}
```

Sr no	Input	Output
1	-2	Beyond the range
2	0	Beyond the range
3	1	Square of 1 is 1
4	100	Square of 100 is 10000
5	101	Beyond the range
6	4	Square of 4 is 16
7	62	Square of 62 is 3844

Test Cases

Test case 1 : {I1 ,O1}

Test case 2 : {I2 ,O2}

Test case 3 : {I3, O3}

Test case 4 : {I4, O4}

Test case 5 : {I5, O5}

Test case 6 : {I6, O6}

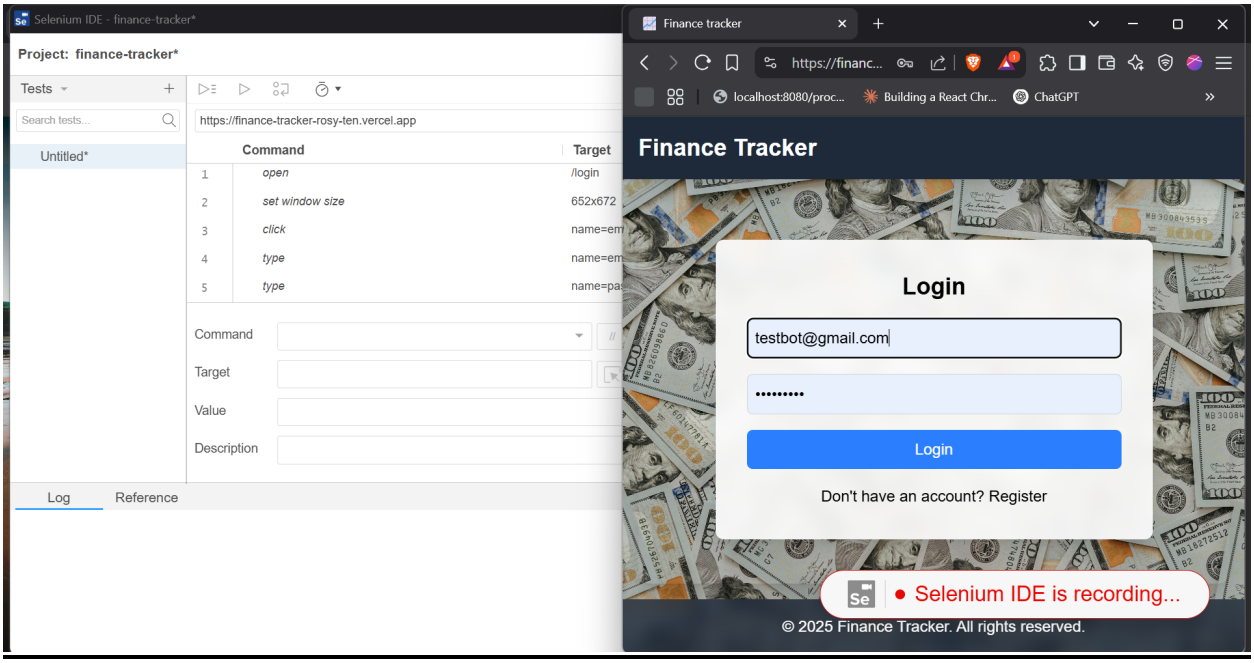
Test case 7 : {I7, O7}

Black-box testing

Knowing the specified function that a product has been designed to perform, test to see if that function is fully operational and error free. Includes tests that are conducted at the software interface. Not concerned with internal logical structure of the software It uncovers

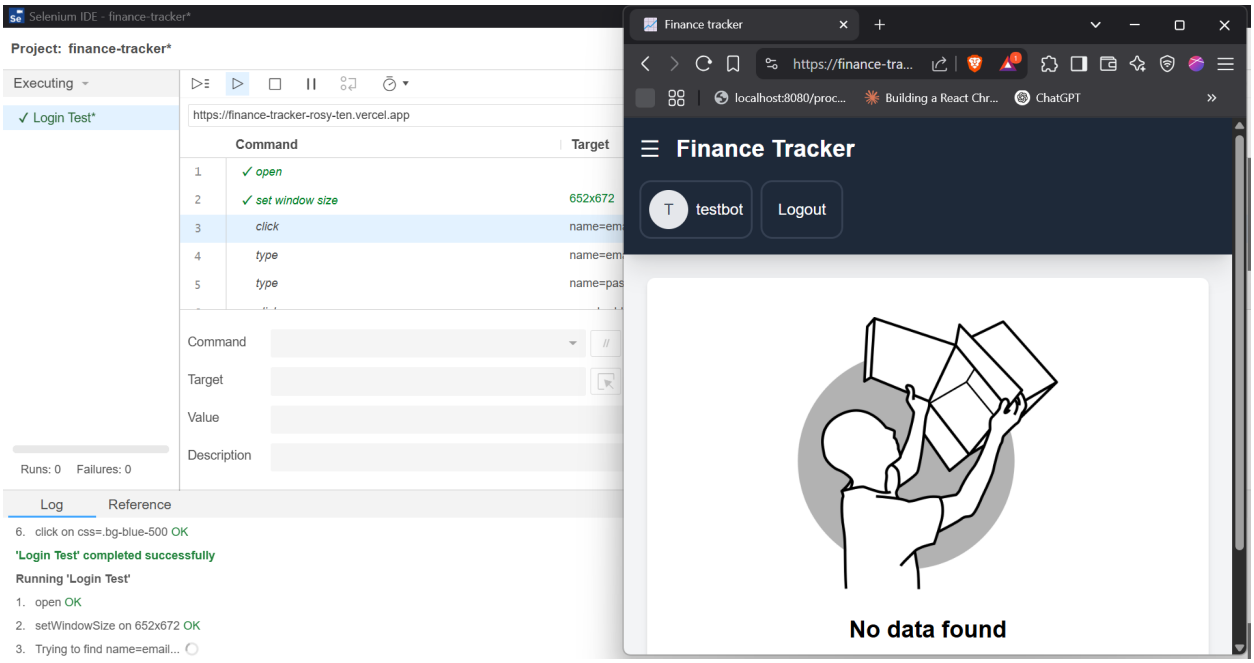
- Incorrect or missing functions
- Interface errors
- Errors in data structures or external data base access
- Behavior or performance errors
- Initialization and termination errors

Result and Discussion:



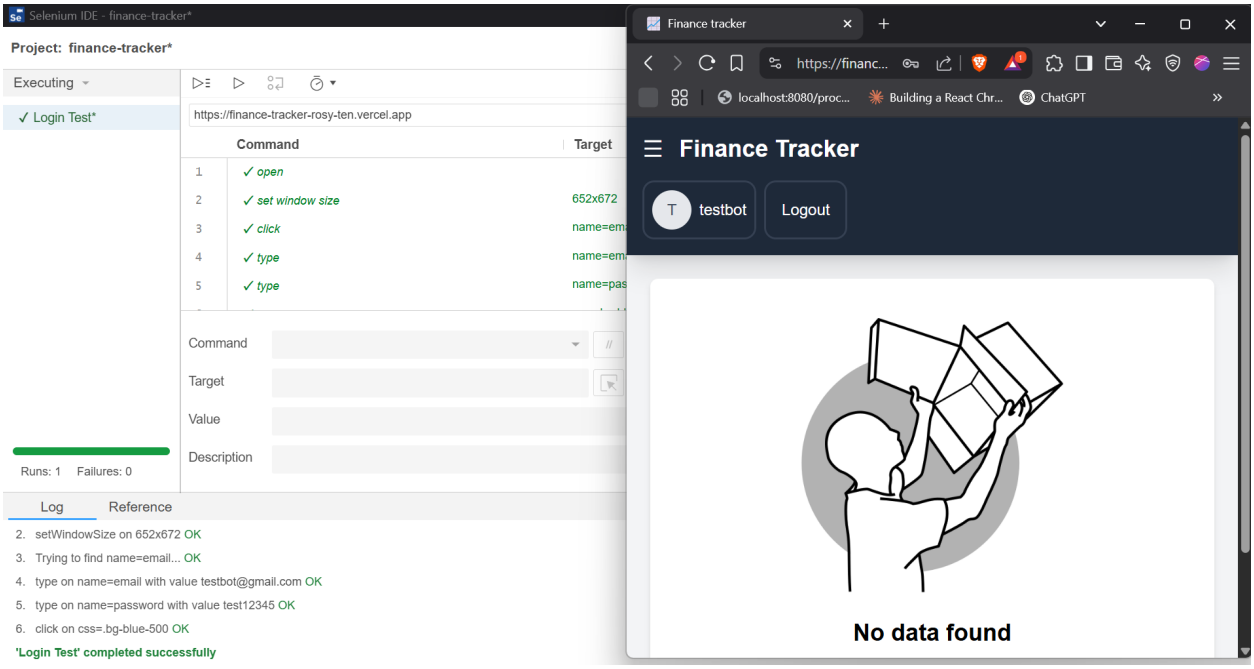
Step 1: Starting Selenium IDE and Recording the Test

In the first screenshot, we have opened Selenium IDE and started recording a new test for the Finance Tracker login page (https://finance-tracker-rosy-ten.vercel.app/login). As you interact with the web page — by clicking on the user ID and password — Selenium IDE records each of these steps. The commands such as click and type are automatically added to the test script. Here, the test inputs are: User ID as testbot@gmail.com , Password as test12345, Finally, we click on the "Login" button, and that action is also recorded.



Step 2: Playing Back the Test

In the second screenshot, you are running (playing back) the recorded test. The Selenium IDE interface shows that the test named **Login Test** has started executing. The page on the right side of the screen says "Preparing to run your test," indicating that the browser is getting ready to repeat the same actions that were recorded. At this point, the script has already opened the login page and set the window size, and it's about to enter the User email Id.



Step 3: Test Execution Completed Successfully (Third Image)

In the final screenshot, all the steps of your test case have been executed successfully. Selenium IDE shows green check marks next to each command — confirming that it was able to open the site, click and type in the user ID, password and finally click the login button without any issues. At the bottom of the Selenium IDE window, the message "Login Test completed successfully" confirms that your automated test passed and that the login form automation worked as expected.

Learning Outcomes:

LO1: Students will be able to understand Software Testing Concepts and the various Software standards.
LO2: to test a software with the help of Junit
LO3: create test cases
LO4: To understand different tools for testing

Course outcomes: Upon completion of the course students will be able to write test cases for the project.

Conclusion:

For Faculty Use

Correction Parameters	Formative Assessment [40%]	Timely completion of Practical [40%]	Attendance / Learning Attitude [20%]	
Marks Obtained				