

---

## PROJECT: EXPENSIFY

---

IT314 – SOFTWARE ENGINEERING

GROUP 26

**TASK : GUI Testing**



Done by: Kkavy Dave – 202201421

### **GROUP MEMBERS:**

ID	NAME
202201410	SIDDHANT KOTAK
202201419	SHAH MAAHI BIJAL
202201421	KKAVY KAPIL DAVE
202201440	MUNJAPARA DHARMIL JAYANTILAL
202201446	SONI DEVARSH SUSHANT
202201449	LATHIGARA PRIYANSH HARESHBHAI
202201452	DETROJA AVI ASHISHBHAI
202201456	MANAV TIRATH PAREKH
202201458	MAKWANA SRUSHTI NARESHKUMAR

# Webpages in Expensify:

1. Home
2. About Us
3. Sign Up
4. Sign In
5. Dashboard
6. Budgets
  - a. Create Budget
  - b. Edit budget
  - c. Delete Budget
  - d. Add expense
  - e. Delete expense
  - f. Payment methods
  - g. Currency selection
7. Reports
8. Downloads
  - a. Choose Date
9. Help
10. Profile Photo
  - a. Manage account
  - b. Sign Out

## GUI Testing using Selenium IDE

### 1. Sign In

Selenium IDE - ExpenseTrackerFinal\*

Project: ExpenseTrackerFinal\*

Tests + [Run current test Ctrl+R] [Run all tests]

Search tests...

Command	Target	Value
1 open	/	
2 set window size	1158x864	
3 click	css= bg-blue-500	
4 click	css= ci-socialButtonsProviderIcon__google	
5 click	css= yAIK0b	
6 click	css= VIPpkd-ksKsZd-mWPk3d-OWXEXe-TvBISd-UJZMc > VIPpkd-RLmnJb	
7 click	css= bg-blue-500	
8 click	css= gap-2	

Command [ ] [ ] [ ] [ ]

Target [ ] [ ] [ ] [ ]

Value [ ] [ ] [ ] [ ]

Description [ ] [ ] [ ] [ ]

Log Reference

3. click on css= bg-blue-500 OK	01:51:55
4. Trying to find css= ci-socialButtonsProviderIcon__google... OK	01:51:55
5. click on css= yAIK0b OK	01:51:56
6. Trying to find css= VIPpkd-ksKsZd-mWPk3d-OWXEXe-TvBISd-UJZMc > VIPpkd-RLmnJb... OK	01:51:59
7. click on css= bg-blue-500 OK	01:52:13
8. Trying to find css= gap-2... OK	01:52:15
'Signin' completed successfully	01:52:17

All fields work fine and were verified by running the test case from the IDE as shown above in the screenshot, ran successfully without any errors.

## 2. Adding a budget

The screenshot displays the Selenium IDE interface for a project named 'ExpenseTrackerFinal'. The URL bar shows 'https://expensify-final.vercel.app'. The test case 'Adding a budget from dashboard' is selected, and its steps are listed in a table below.

	Command	Target	Value
1	open	/	
2	set window size	1158x864	
3	click	css=bg-blue-500	
4	click	css=gap-2	
5	click	css=a:nth-child(2) > flex	
6	click	css=bg-[#2d2d2d] > .text-3xl	
7	click	css=mt-4:nth-child(2) > flex	
8	type	css=mt-4:nth-child(2) > flex	TestBudgetbyGUI
9	click	css=mt-4:nth-child(3) > flex	
10	type	css=mt-4:nth-child(3) > flex	1000
11	click	css=bg-[#3a3a3a]	

Below the table, there are input fields for Command, Target, Value, and Description. The Log panel at the bottom shows the execution results of the test case, indicating that all steps passed successfully.

Log

Step	Message	Time
6	Trying to find css=bg-[#2d2d2d] > .text-3xl... OK	01:55:04
7	click on css=mt-4:nth-child(2) > flex OK	01:55:05
8	type on css=mt-4:nth-child(2) > flex with value TestBudgetbyGUI OK	01:55:05
9	click on css=mt-4:nth-child(3) > flex OK	01:55:05
10	type on css=mt-4:nth-child(3) > flex with value 1000 OK	01:55:06
11	click on css=bg-[#3a3a3a] OK	01:55:06
'Adding a budget from dashboard' completed successfully		01:55:06

All fields work fine and were verified by running the test case from the IDE as shown above in the screenshot, ran successfully without any errors.

### 3. Adding an expense

The screenshot shows the Selenium IDE interface for a project named 'ExpenseTrackerFinal'. The 'Tests' tab is active, displaying a test case titled 'Add expense \*'. The test case consists of 12 steps, each with a command, target, and value. The commands are: 'set window size', 'click', 'click', 'click', 'click', 'click', 'type', 'type', 'select', and 'click'. The targets are: '1162x864', 'css= bg-blue-500', 'css= gap-2', 'css=a:nth-child(2) > flex', 'css=p-5:nth-child(5) .flex > flex > div', 'css=div:nth-child(1) > border', 'css=div:nth-child(1) > border', 'css=div:nth-child(2) > border-input', 'css=div:nth-child(4) > .bg-[#3a3a3a]', 'css=div:nth-child(4) > .bg-[#3a3a3a]', and 'css= inline-flex'. The values are: 'Testexpense' and '100'. The 'Log' tab is also visible, showing the execution of the test case, with a final status of 'Add expense \* completed successfully'.

Step	Command	Target	Value
2	set window size	1162x864	
3	click	css= bg-blue-500	
4	click	css= gap-2	
5	click	css=a:nth-child(2) > flex	
6	click	css=p-5:nth-child(5) .flex > flex > div	
7	click	css=div:nth-child(1) > border	
8	type	css=div:nth-child(1) > border	Testexpense
9	type	css=div:nth-child(2) > border-input	100
10	click	css=div:nth-child(4) > .bg-[#3a3a3a]	
11	select	css=div:nth-child(4) > .bg-[#3a3a3a]	label=Debit
12	click	css= inline-flex	

Log Reference

7. click on css=div:nth-child(1) > border OK 02:01:20

8. type on css=div:nth-child(1) > border with value Testexpense OK 02:01:22

9. type on css=div:nth-child(2) > border-input with value 100 OK 02:01:24

10. click on css=div:nth-child(4) > .bg-[#3a3a3a] OK 02:01:26

11. select on css=div:nth-child(4) > .bg-[#3a3a3a] with value label=Debit OK 02:01:28

12. click on css= inline-flex OK 02:01:30

'Add expense \* completed successfully' 02:01:30

All fields work fine and were verified by running the test case from the IDE as shown above in the screenshot, ran successfully without any errors.

### 4. Deleting an expense

The screenshot shows the Selenium IDE interface for a project named 'ExpenseTrackerFinal'. The 'Tests' tab is active, displaying a test case titled 'Deleting expense \*'. The test case consists of 7 steps, each with a command, target, and value. The commands are: 'open', 'set window size', 'click', 'click', 'click', 'click', and 'click'. The targets are: '/', '1158x864', 'css= bg-blue-500', 'css= gap-2', 'css=a:nth-child(2) > flex', 'css=p-5:nth-child(5) .font-semantic', and 'css= grid:nth-child(3) lucide'. The values are: 'lucide'. The 'Log' tab is also visible, showing the execution of the test case, with a final status of 'Deleting expense \* completed successfully'.

Step	Command	Target	Value
1	open	/	
2	set window size	1158x864	
3	click	css= bg-blue-500	
4	click	css= gap-2	
5	click	css=a:nth-child(2) > flex	
6	click	css=p-5:nth-child(5) .font-semantic	
7	click	css= grid:nth-child(3) lucide	lucide

Log Reference

2. setWindowSize on 1158x864 OK 02:04:42

3. click on css= bg-blue-500 OK 02:04:44

4. click on css= gap-2 OK 02:04:46

5. click on css=a:nth-child(2) > flex OK 02:04:49

6. click on css=p-5:nth-child(5) .font-semantic OK 02:04:51

7. Trying to find css= grid:nth-child(3) lucide OK 02:04:53

'Deleting expense \* completed successfully' 02:04:54

All fields work fine and were verified by running the test case from the IDE as shown above in the screenshot, ran successfully without any errors.

## 5. Editing budget

The screenshot shows the Selenium IDE interface for a project named 'ExpenseTrackerFinal'. The test case 'Editing a budget' is selected and is running. The test steps are as follows:

Step	Command	Target	Value
1	open	/	
2	set window size	1158x864	
3	click	css=bg-blue-500	
4	click	css=gap-2	
5	click	css=a:nth-child(2) > flex	
6	click	css=p-5:nth-child(5) > div	
7	click	css=text-primary-foreground	
8	click	css=mt-4:nth-child(3) > flex	
9	type	css=mt-4:nth-child(3) > flex	10000
10	click	css=hover(3A bg-!#555)	

The log shows the following steps:

- 5. click on css=a:nth-child(2) > flex OK
- 6. click on css=p-5:nth-child(5) > div OK
- 7. click on css=text-primary-foreground OK
- 8. click on css=mt-4:nth-child(3) > flex OK
- 9. type on css=mt-4:nth-child(3) > flex with value 10000 OK
- 10. click on css=hover(3A bg-!#555) OK

The test case 'Editing a budget' completed successfully.

All fields work fine and were verified by running the test case from the IDE as shown above in the screenshot, ran successfully without any errors.

## 6. Deleting a budget

The screenshot shows the Selenium IDE interface for a project named 'ExpenseTrackerFinal'. The test case 'Deleting a budget' is selected and is running. The test steps are as follows:

Step	Command	Target	Value
1	open	/	
2	set window size	1158x864	
3	click	css=bg-blue-500	
4	click	css=gap-2	
5	click	css=a:nth-child(2) > flex	
6	mouse over	css=a:nth-child(3) > flex	
7	click	css=p-5:nth-child(5) flex > flex font-bold	
8	click	css=bg-red-600	
9	click	css=px-6	

The log shows the following steps:

- 4. Trying to find css=gap-2... OK
- 5. click on css=a:nth-child(2) > flex OK
- 6. mouseOver on css=a:nth-child(3) > flex OK
- 7. click on css=p-5:nth-child(5) flex > flex font-bold OK
- 8. click on css=bg-red-600 OK
- 9. click on css=px-6 OK

The test case 'Deleting a budget' completed successfully.

All fields work fine and were verified by running the test case from the IDE as shown above in the screenshot, ran successfully without any errors.

## 7. Changing password

The screenshot displays the Selenium IDE interface for a project named "ExpenseTrackerFinal". The "Tests" tab is active, showing a list of test cases on the left and a detailed view of the selected test case, "Adding password in settings", on the right. The test case steps are as follows:

Step	Command	Target	Value
3	click	css=bg-blue-500	
4	click	css=gap-2	
5	click	css=cl-rootBox:nth-child(1) cl-avatarImage	
6	click	css=cl-userButtonPopoverActionButton__manageAccount	
7	click	css=cl-navbarButton__security	
8	click	css=cl-profileSectionPrimaryButton__password	
9	type	id=newPassword-field	hellothisistest
10	type	id=confirmPassword-field	hellothisistest
11	click	id=signOutOfOtherSessions-field	
12	click	css=cl-formButtonPrimary	

Below the test case steps, there are input fields for "Command", "Target", "Value", and "Description". At the bottom, the "Log" tab is active, showing a list of log entries for the test case, all of which are successful:

Log Entry	Time
7. click on css=cl-navbarButton__security OK	02:16:32
8. click on css=cl-profileSectionPrimaryButton__password OK	02:16:33
9. type on id=newPassword-field with value hellothisistest OK	02:16:34
10. type on id=confirmPassword-field with value hellothisistest OK	02:16:34
11. click on id=signOutOfOtherSessions-field OK	02:16:35
12. click on css=cl-formButtonPrimary OK	02:16:36
'Adding password in settings' completed successfully	02:16:36

All fields work fine and were verified by running the test case from the IDE as shown above in the screenshot, ran successfully without any errors.

## 8. Sorting expense by date and amount / changing currency to USD, EUR, INR

The screenshot shows the Selenium IDE interface for a project named "ExpenseTrackerFinal". The test case "Sorting Expense" is selected in the left sidebar. The main table lists the following commands and targets:

Command	Target	Value
click	css= gap-2	
click	css= px-4 nth-child(2)	
select	css= px-4 nth-child(2)	label=Sort by Amount
click	css= px-4 nth-child(2)	
select	css= px-4 nth-child(2)	label=Sort by Date
click	css= mr-2	
select	css= mr-2	label=EUR
click	css= mr-2	
select	css= mr-2	label=INR
click	css= mr-2	
select	css= mr-2	label=USD

The Log tab at the bottom shows the execution results:

Log	Reference
9. click on css= mr-2 OK	02:23:44
10. select on css= mr-2 with value label=EUR OK	02:23:44
11. click on css= mr-2 OK	02:23:45
12. select on css= mr-2 with value label=INR OK	02:23:46
13. click on css= mr-2 OK	02:23:47
14. select on css= mr-2 with value label=USD OK	02:23:48
'Sorting Expense' completed successfully	

All fields work fine and were verified by running the test case from the IDE as shown above in the screenshot, ran successfully without any errors.

## 9. Testing Logo to homepage

The screenshot shows the Selenium IDE interface for the same project. The test case "Testing Logo to homepage" is selected. The main table lists the following commands and targets:

Command	Target	Value
open	/	
set window size	1162x864	
click	css= bg-blue-500	
click	css= gap-2	
click	css= .items-center > cursor-pointer	

The Log tab at the bottom shows the execution results:

Log	Reference
Running "Testing Logo to homepage"	
1. open on / OK	02:25:41
2. setWindowSize on 1162x864 OK	02:25:41
3. click on css= bg-blue-500 OK	02:25:42
4. Trying to find css= gap-2... OK	02:25:43
5. Trying to find css= .items-center > cursor-pointer... OK	02:25:45
'Testing Logo to homepage' completed successfully	

All fields work fine and were verified by running the test case from the IDE as shown above in the screenshot, ran successfully without any errors.

## 10. Report Page

The screenshot displays the Selenium IDE interface for a project named 'ExpenseTrackerFinal'. The 'Tests' pane on the left lists several test cases, with 'ReportPage\*' selected. The main workspace shows a table of commands for the 'ReportPage\*' test case:

Command	Target	Value
open	/	
set window size	1162x864	
click	css=bg-blue-500	
click	css=gap-2	
click	css=a:nth-child(3) > flex	
click	css=canvas	

Below the table, there are input fields for 'Command', 'Target', 'Value', and 'Description'. The 'Log' pane at the bottom shows the execution results:

- 1. open on / OK 02:27:59
- 2. setWindowSize on 1162x864 OK 02:28:00
- 3. click on css=bg-blue-500 OK 02:28:00
- 4. Trying to find css=gap-2... OK 02:28:02
- 5. click on css=a:nth-child(3) > flex OK 02:28:04
- 6. Trying to find css=canvas... OK 02:28:05
- 'ReportPage' completed successfully 02:28:07

All fields work fine and were verified by running the test case from the IDE as shown above in the screenshot, ran successfully without any errors.

## 11. Help Page

The screenshot displays the Selenium IDE interface for a project named 'ExpenseTrackerFinal'. The 'Tests' pane on the left lists several test cases, with 'HelpPage\*' selected. The main workspace shows a table of commands for the 'HelpPage\*' test case:

Command	Target	Value
open	/	
set window size	1166x864	
click	css=bg-blue-500	
click	css=gap-2	
mouse over	css=a:nth-child(2) > flex	
mouse out	css=a:nth-child(2) > flex	
click	css=a:nth-child(5) > flex	
click	css=items-center > cursor-pointer	

Below the table, there are input fields for 'Command', 'Target', 'Value', and 'Description'. The 'Log' pane at the bottom shows the execution results:

- 3. click on css=bg-blue-500 OK 02:29:18
- 4. Trying to find css=gap-2... OK 02:29:19
- 5. mouseOver on css=a:nth-child(2) > flex OK 02:29:21
- 6. mouseOut on css=a:nth-child(2) > flex OK 02:29:22
- 7. click on css=a:nth-child(5) > flex OK 02:29:22
- 8. click on css=items-center > cursor-pointer OK 02:29:23
- 'HelpPage' completed successfully 02:29:23



All fields work fine and were verified by running the test case from the IDE as shown above in the screenshot, ran successfully without any errors.

## 12. Go to home page

**Selenium IDE - ExpenseTrackerFinal**

Project: ExpenseTrackerFinal

Search tests... Run current test Ctrl+R rcel.app

	Command	Target	Value
✓ Add expense *			
✓ Adding a budget from das	4 ✓ click	css= gap-2	
✓ Adding password in settin	5 ✓ click	css= block > flex	
✓ Deleting a budget*	6 ✓ click	css= gap-2	
✓ Deleting expense*	7 ✓ run script	window.scrollTo(0,51.2000076293945)	
✓ Editing a budget *	8 ✓ run script	window.scrollTo(0,351.20001220703125)	
✓ Go to home page*	9 ✓ click	css=a:nth-child(3) > flex	
✓ HelpPage*	10 ✓ run script	window.scrollTo(0,96.80000305175781)	
✓ ReportPage*	11 ✓ click	css=a:nth-child(4) > flex	
✓ SignIN*	12 ✓ click	css=a:nth-child(5) > flex	
✓ Sorting Expense*	13 ✓ click	css= block > flex	
✓ Testing Logo to homepage	14 ✓ run script	window.scrollTo(0,0)	

Command: [dropdown] [icon] [icon]

Target: [input] [icon] [icon]

Value: [input]

Description: [input]

**Log** Reference

9. click on css=a:nth-child(3) > flex OK	02:30:41
10. runScript on window.scrollTo(0,96.80000305175781) OK	02:30:42
11. click on css=a:nth-child(4) > flex OK	02:30:43
12. click on css=a:nth-child(5) > flex OK	02:30:43
13. click on css= block > flex OK	02:30:44
14. runScript on window.scrollTo(0,0) OK	02:30:45
'Go to home page' completed successfully	

All fields work fine and were verified by running the test case from the IDE as shown above in the screenshot, ran successfully without any errors.

### 13. Downloading latest expense

Selenium IDE - ExpenseTrackerFinal\*

Project: ExpenseTrackerFinal\*

Tests

Search tests...

✓ Add expense \*

✓ Adding a budget from das

✓ Adding password in settin

✓ Deleting a budget\*

✓ Deleting expense\*

✓ Download Expense\*

✓ Editing a budget \*

✓ Go to home page\*

✓ HelpPage\*

✓ ReportPage\*

✓ SignIN\*

✓ Sorting Expense\*

✓ Testing Logo to homepage

https://expensify-final.vercel.app

	Command	Target	Value
2	✓ set window size	1162x864	
3	✓ click	css=bg-blue-500	
4	✓ click	css=gap-2	
5	✓ click	css=a:nth-child(4) > flex	
6	✓ mouse over	css=a:nth-child(3) > flex	
7	✓ click	css=px-4:nth-child(1)	
8	✓ type	css=px-4:nth-child(1)	2024-11-27
9	✓ click	css=px-4:nth-child(2)	
10	✓ type	css=px-4:nth-child(2)	2024-12-02
11	✓ click	css=px-6	

Command

Target

Value

Description

Log

Reference

6. mouseOver on css=a:nth-child(3) > flex OK02:33:19

7. click on css=px-4:nth-child(1) OK02:33:20

8. type on css=px-4:nth-child(1) with value 2024-11-27 OK02:33:21

9. click on css=px-4:nth-child(2) OK02:33:21

10. type on css=px-4:nth-child(2) with value 2024-12-02 OK02:33:22

11. click on css=px-6 OK02:33:23

'Download Expense' completed successfully02:33:23

All fields work fine and were verified by running the test case from the IDE as shown above in the screenshot, ran successfully without any errors.