

# IT-314 Software Engineering

## GUI Testing



### Project Title: QuestAI

### Group 12

#### **1. Initial Authentication Pages - Script-Based Tests with Assertions**

The following pages were tested using manual scripting + assertions, because they required precise validation:

- **Sign-Up & OTP**
- **Successful Login**
- **Forgot Password\_Login\_ResetPassword**

For these pages:

We used `assertElementPresent`, `assertValue`, and toast checks  
We verified input fields, password fields, and page transitions  
Selenium steps were written manually (not recorded)  
Assertions were necessary to confirm correctness

These tests were written first, because OTP-related flows cannot be captured reliably through recording.

## 2. Remaining Application Pages — Recording Method

Once the login-related pages were finished, all normal in-app navigation pages were tested using the Selenium IDE record & playback feature as shown and explained in the demo video:

- **Navbar (Logged Out)**
- **Navbar (Logged In)**
- **About**
- **Edit Profile**
- **Create Story Page & Chatbox**
- **Public Stories Page**
- **Home Page(manually)**

For these pages:

Only clicks + navigation flow

Goal: validate user journey

Everything created through recording for natural flow testing

Testing of each page is shown below.

### Sign Up and OTP page

Extension: (Selenium IDE) - Selenium IDE - SE\_QuestAI\_GUI\* — Mozilla Firefox

Project: SE\_QuestAI\_GUI\*

Tests +

Search tests...

Playback base URL

	Command	Target	Value
1.	✓ execute script	window.localStorage.clear();	
2.	✓ open	https://quest-ai-frontend.vercel.app/	
3.	✓ wait for element visible	id=name	30000
4.	✓ type	id=name	Tester
5.	✓ assert value	id=name	Tester

Command

Target

Value

Description

Log	Reference	
2. open on https://quest-ai-frontend.vercel.app/ OK		15:06:26
3. waitForElementVisible on id=name with value 30000 OK		15:06:26
4. type on id=name with value Tester OK		15:06:27
5. assertValue on id=name with value Tester OK		15:06:27
6. type on id=email with value 202301119@dau.ac.in OK		15:06:27
7. assertValue on id=email with value 202301119@dau.ac.in OK		15:06:28
8. type on id=password with value 123456 OK		15:06:28

Extension: (Selenium IDE) - Selenium IDE - SE\_QuestAI\_GUI\* — Mozilla Firefox

Project: SE\_QuestAI\_GUI\*

Tests +

Search tests...

✓ Sign\_up&OTP\*

Playback base URL

	Command	Target	Value
7	✓ assert value	id=email	2023011119@oau.ac.in
8	✓ type	id=password	123456
9	✓ assert value	id=password	123456
10	✓ click	xpath=//button[contains(text(),'SHOW') or contains(text(),'HIDE')]	
11	✓ click	xpath=//button[contains(text(),'SIGN-UP')]	

Command

Target

Value

Description

Log

Reference

9. assertValue on id=password with value 123456 OK

10. click on xpath=//button[contains(text(),'SHOW') or contains(text(),'HIDE')] OK

11. click on xpath=//button[contains(text(),'SIGN-UP')] OK

12. Trying to find css=.toast... OK

echo: OTP sent to your email! Redirecting

14. waitForElementVisible on xpath=//h2[contains(text(),'VERIFY OTP')] with value 30000 OK

'Sign\_up&OTP' completed successfully

15:06:28

15:06:28

15:06:28

15:06:28

15:06:33

15:06:33

15:06:33

Extension: (Selenium IDE) - Selenium IDE - SE\_QuestAI\_GUI\* — Mozilla Firefox

Project: SE\_QuestAI\_GUI\*

Tests +

Search tests...

✓ Sign\_up&OTP\*

Playback base URL

	Command	Target	Value
11	✓ click	xpath=//button[contains(text(),'SIGN-UP')]	
12	✓ store text	css=.toast	toastMessage
13	✓ echo	\${toastMessage}	
14	✓ wait for element visible	xpath=//h2[contains(text(),'VERIFY OTP')]	

Command

Target

Value

Description

Log

Reference

9. assertValue on id=password with value 123456 OK

10. click on xpath=//button[contains(text(),'SHOW') or contains(text(),'HIDE')] OK

11. click on xpath=//button[contains(text(),'SIGN-UP')] OK

12. Trying to find css=.toast... OK

echo: OTP sent to your email! Redirecting

14. waitForElementVisible on xpath=//h2[contains(text(),'VERIFY OTP')] with value 30000 OK

'Sign\_up&OTP' completed successfully

15:06:28

15:06:28

15:06:28

15:06:28

15:06:33

15:06:33

15:06:33

The test confirmed that the Sign-Up page loads properly, takes all the inputs properly, and successfully sends the OTP on the provided email. A toast appears, and the app correctly redirects to the OTP page, so the Sign-Up - OTP flow works as expected.

Since Selenium IDE can't read real email OTPs, the OTP submission and "Resend OTP" features were tested manually. These worked fine, and after entering the correct OTP, the sign up was successful.

Also, Google Sign-In can't be automated because Google blocks automated OAuth popups, so this feature was manually tested as well.

## Successful Login

The screenshot displays the Selenium IDE interface within a Mozilla Firefox browser window. The project is named 'SE\_QuestAI\_GUI'. The test suite 'Sign\_up&OTP' contains a test 'Successful\_Sign\_in'. The test script is as follows:

Step	Command	Target	Value
1	open	https://quest-ai-frontend.vercel.app/Login	
2	wait for element visible	id=email	30000
3	assert element present	id=password	
4	type	id=email	202301119@dau.ac.in
5	assert value	id=email	202301119@dau.ac.in
7	click	xpath=//button[contains(text(),'SHOW') or contains(text(),'HIDE')]	
8	click	xpath=//form//button[contains(text(),'LOGIN')]	
9	wait for element present	css=.toast	30000
10	store text	css=.toast	toastText
11	echo	\${toastText}	

The log shows the execution of the test steps, with the final echo command displaying 'Sign in successful!'.

Log

- 1. open on https://quest-ai-frontend.vercel.app/Login OK 15:57:02
- 2. waitForElementVisible on id=email with value 30000 OK 15:57:02
- 3. assertElementPresent on id=password OK 15:57:02
- 4. type on id=email with value 202301119@dau.ac.in OK 15:57:02
- 5. assertValue on id=email with value 202301119@dau.ac.in OK 15:57:03
- 6. type on id=password with value 123456 OK 15:57:03
- 7. click on xpath=//button[contains(text(),'SHOW') or contains(text(),'HIDE')] OK 15:57:03
- 8. click on xpath=//form//button[contains(text(),'LOGIN')] OK 15:57:03
- 9. waitForElementPresent on css=.toast with value 30000 OK 15:57:03
- 10. storeText on css=.toast with value toastText OK 15:57:05
- echo: Sign in successful! 15:57:05
- 12. waitForElementPresent on xpath=//h2 with value 30000 OK 15:57:05

Extension: (Selenium IDE) - Selenium IDE - SE\_QuestAI\_GUI — Mozilla Firefox

Project: SE\_QuestAI\_GUI

Tests: ☒ Sign\_up&OTP ☒ Successful\_Sign\_in

Search tests...

https://quest-ai-frontend.vercel.app/

	Command	Target	Value
12	✓ wait for element present	xpath=//h2	30000
13	✓ execute script	return window.localStorage.getItem("accessToken");	accessToken
14	✓ echo	\${accessToken}	

Command  //

Target

Value

Description

Log Reference

echo: ✓ Sign in successful! 15:57:05

12. waitForElementPresent on xpath=//h2 with value 30000 OK 15:57:05

13. executeScript on return window.localStorage.getItem("accessToken"); with value accessToken OK 15:57:06

echo: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2OTI0Mjc2ZDEzMmE5M2VmYzJjMmI1NmYiLCJlbWVpbiI6IjwvMjJwMTExOUYyYUwuaW4iLCJ1c2VybmFIZSI6IiRlc3RlcilslmhdCi6MTc2Mzk... 15:57:06

'Successful\_Sign\_in' completed successfully 15:57:06

This test checks that the Login page loads correctly and that both the email and password fields are present and can accept the correct input. After entering the correct credentials, the test clicks the Login button and waits for the success toast message. It then confirms the login and the user has successfully logged in.

## ForgetPassword\_Login\_ResetPassword

Extension: (Selenium IDE) - Selenium IDE - SE\_QuestAI\_GUI — Mozilla Firefox

Project: SE\_QuestAI\_GUI

Tests: ☒ ForgetPassword\_Login\_ResetPassword ☒ Sign\_up&OTP ☒ Successful\_Login

Search tests...

https://quest-ai-frontend.vercel.app/

	Command	Target	Value
1	✓ execute script	window.localStorage.clear();	
2	✓ open	https://quest-ai-frontend.vercel.app/Login	
3	✓ wait for element visible	xpath=//button[contains(text(),'Forgot password')]	30000
4	✓ click	xpath=//button[contains(text(),'Forgot password')]	

Command  //

Target

Value

Description

Log Reference

Running 'ForgetPassword\_Login\_ResetPassword' 16:52:28

1. executeScript on window.localStorage.clear(); OK 16:52:29

2. open on https://quest-ai-frontend.vercel.app/Login OK 16:52:29

3. waitForElementVisible on xpath=//button[contains(text(),'Forgot password')] with value 30000 OK 16:52:29

4. click on xpath=//button[contains(text(),'Forgot password')] OK 16:52:30

5. waitForElementVisible on id=email with value 30000 OK 16:52:30

Extension: (Selenium IDE) - Selenium IDE - SE\_QuestAI\_GUI — Mozilla Firefox

Project: SE\_QuestAI\_GUI

Tests

Search tests...

✓ ForgetPassword\_Login\_ResetPassword

✓ Sign\_up&OTP

✓ Successful\_Login

https://quest-ai-frontent.vercel.app/

	Command	Target	Value
4	✓ click	xpath=//button[contains(text(),'Forgot password')]	
5	✓ wait for element visible	id=email	30000
6	✓ assert element present	id=email	
7	✓ type	id=email	202301119@dau.ac.in

Command

Target

Value

Description

Log

Reference

7. type on id=email with value 202301119@dau.ac.in OK 16:52:31

8. assertValue on id=email with value 202301119@dau.ac.in OK 16:52:31

9. click on xpath=//button[contains(.,'SEND OTP') or contains(text(),'[ SEND OTP ]')] OK 16:52:31

10. waitForElementPresent on css=.toast with value 30000 OK 16:52:31

11. storeText on css=.toast with value fpToast OK 16:52:34

echo: ✓ OTP sent to your email! 16:52:34

Extension: (Selenium IDE) - Selenium IDE - SE\_QuestAI\_GUI — Mozilla Firefox

Project: SE\_QuestAI\_GUI

Tests

Search tests...

✓ ForgetPassword\_Login\_ResetPassword

✓ Sign\_up&OTP

✓ Successful\_Login

https://quest-ai-frontent.vercel.app/

	Command	Target	Value
10	✓ wait for element present	css=.toast	30000
11	✓ store text	css=.toast	fpToast
12	✓ echo	\$(fpToast)	
13	✓ wait for element visible	xpath=//h2[contains(text(),'VERIFY OTP')]	30000

Command

Target

Value

Description

Log

Reference

13. waitForElementVisible on xpath=//h2[contains(text(),'VERIFY OTP')] with value 30000 OK 16:52:34

14. waitForElementVisible on xpath=//h2[contains(text(),'SET NEW PASSWORD')] with value 30000 OK 16:52:34

15. assertElementPresent on id=password OK 16:52:51

16. assertElementPresent on id=confirmPassword OK 16:52:51

17. type on id=password with value 111111 OK 16:52:51

18. assertValue on id=password with value 111111 OK 16:52:51

19. assertElementPresent on id=confirmPassword with value 111111 OK 16:52:51

The top screenshot shows the Selenium IDE interface for a test project named 'SE\_QuestAI\_GUI'. The test steps are as follows:

Step	Command	Target	Value
15	✓ assert element present	id=password	
16	✓ assert element present	id=confirmPassword	
17	✓ type	id=password	111111
18	✓ assert value	id=password	111111
19	✓ type	id=confirmPassword	111111

The bottom screenshot shows the continuation of the test steps:

Step	Command	Target	Value
17	✓ type	id=password	111111
18	✓ assert value	id=password	111111
19	✓ type	id=confirmPassword	111111
20	✓ assert value	id=confirmPassword	111111
21	✓ click	xpath=//button[contains(text(),'SET PASSWORD') or contains(text(),'SUBMIT')]	
22	✓ wait for element visible	css=.toast	30000
23	✓ store text	css=.toast	resetToast
24	✓ echo	\$(resetToast)	

Both screenshots include a 'Log' tab at the bottom showing the execution results of the test steps.

This test covers the full Forgot Password flow starting from the Login page. First, it checks that clicking the “Forgot password?” button correctly opens the Forgot Password page and that the email field accepts a valid email. When the backend sends an OTP, the test also verifies the toast message that appears. After that, it confirms that the user is navigated to the OTP page and then automatically redirects to the Reset Password page.

Since Selenium IDE cannot read or enter real OTPs, the OTP step was tested manually. From the Reset Password page onward, Selenium continues the flow by making sure the New Password and Confirm Password fields are visible and working. Finally, it enters matching passwords, submits the form, verifies the success toast, and confirms that the password reset is completed properly.



## Navbar\_LoggedOut

Extension: (Selenium IDE) - Selenium IDE - SE\_QuestAI\_GUI - Mozilla Firefox

Project: SE\_QuestAI\_GUI

Tests

Search tests...

ForgetPassword\_Login\_ResetPassword

✓ Navbar\_LoggedOut

Sign\_up&OTP

Successful\_Login

https://quest-ai-frontend.vercel.app/

	Command	Target	Value
1	✓ open	https://quest-ai-frontend.vercel.app/	
2	✓ set window size	1204x680	
3	✓ click	css=.hidden > .form-button:nth-child(1)	
4	✓ mouse over	css=.hidden > .form-button:nth-child(1)	
5	✓ click	css=.hidden > .form-button:nth-child(2)	

Command: open

Target: https://quest-ai-frontend.vercel.app/

Value:

Description:

Log Reference

Running 'Navbar\_LoggedOut'

1. open on https://quest-ai-frontend.vercel.app/ OK 21:42:16
2. setWindowSize on 1204x680 OK 21:42:16
3. click on css=.hidden > .form-button:nth-child(1) OK 21:42:16
4. mouseOver on css=.hidden > .form-button:nth-child(1) OK 21:42:17
5. click on css=.hidden > .form-button:nth-child(2) OK 21:42:17

Extension: (Selenium IDE) - Selenium IDE - SE\_QuestAI\_GUI - Mozilla Firefox

Project: SE\_QuestAI\_GUI

Tests

Search tests...

Run all tests (Ctrl+Shift+R) on https://quest-ai-frontend.vercel.app/

ForgetPassword\_Login\_ResetPassword

✓ Navbar\_LoggedOut

Sign\_up&OTP

Successful\_Login

	Command	Target	Value
7	✓ mouse over	css=.hidden > .form-button:nth-child(2)	
8	✓ mouse out	css=.hidden > .form-button:nth-child(2)	
9	✓ click	css=.hidden > .form-button:nth-child(3)	
10	✓ mouse over	css=.hidden > .form-button:nth-child(3)	
11	✓ mouse over	css=.hidden > .form-button:nth-child(3)	

Command: open

Target: https://quest-ai-frontend.vercel.app/

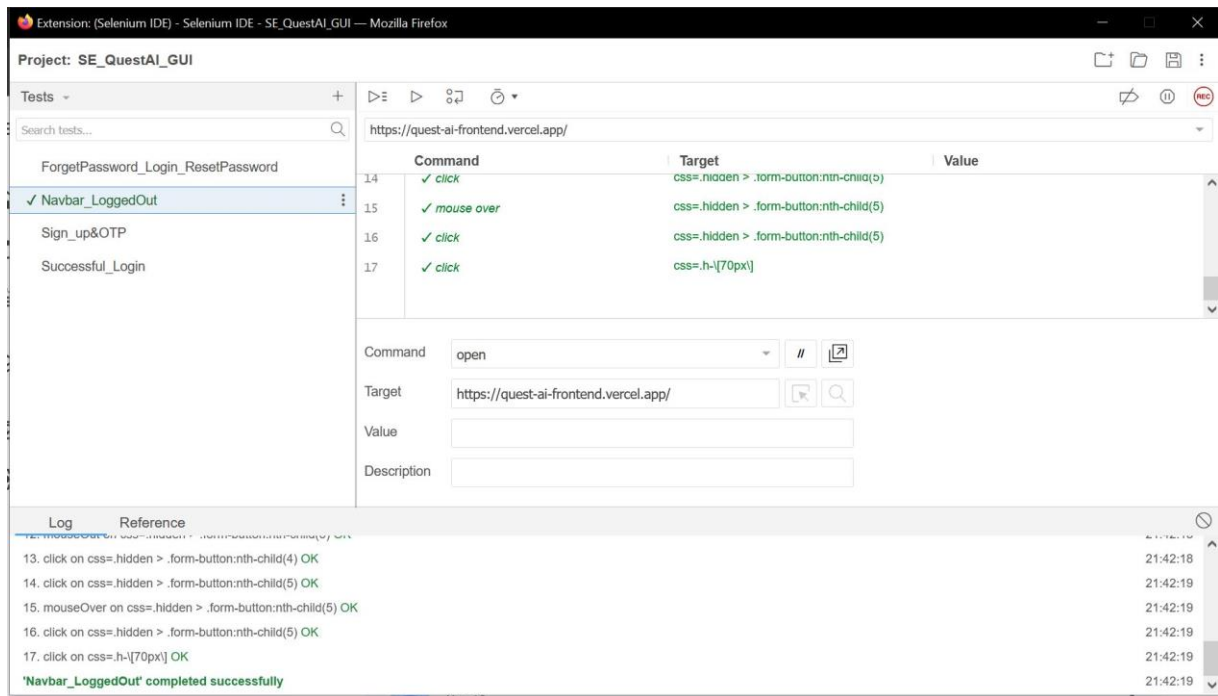
Value:

Description:

Log Reference

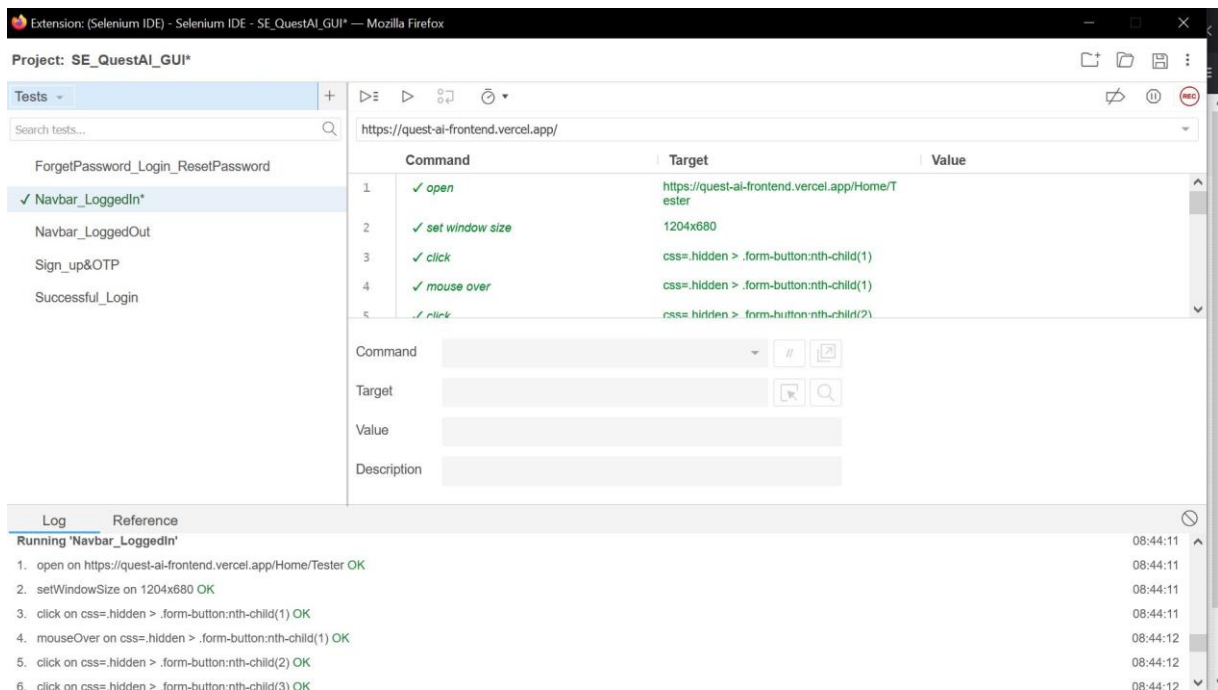
6. mouseOver on css=.hidden > .form-button:nth-child(2) OK 21:42:17
7. mouseOver on css=.hidden > .form-button:nth-child(2) OK 21:42:17
8. mouseOut on css=.hidden > .form-button:nth-child(2) OK 21:42:17
9. click on css=.hidden > .form-button:nth-child(3) OK 21:42:18
10. mouseOver on css=.hidden > .form-button:nth-child(3) OK 21:42:18
11. mouseOver on css=.hidden > .form-button:nth-child(3) OK 21:42:18
12. mouseOut on css=.hidden > .form-button:nth-child(3) OK 21:42:18

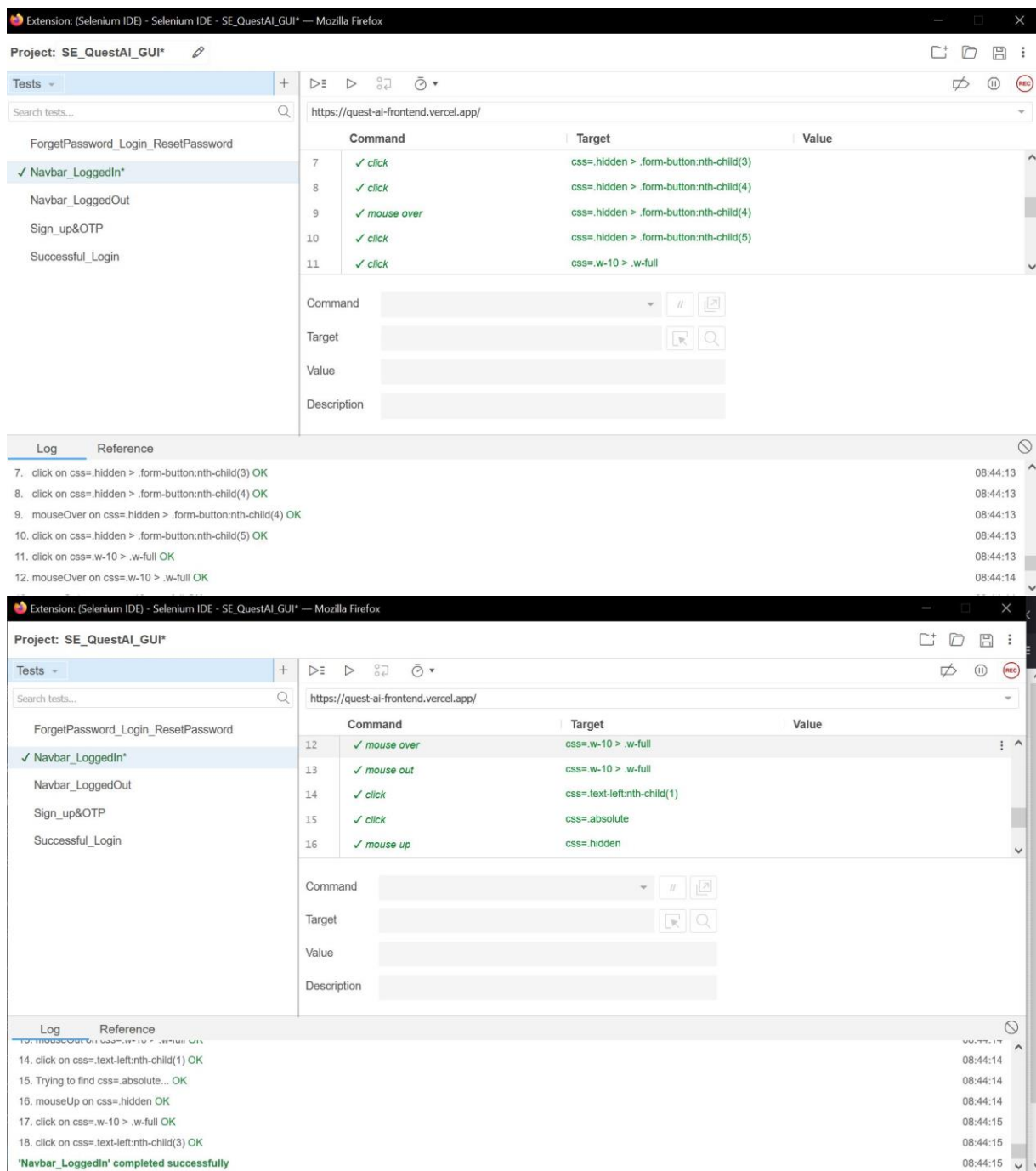




This test checked whether all the Navbar buttons work properly when the user isn't logged in. Starting from the Login page, the about,sign up,login and public stories buttons each opened the pages they were supposed to, and all the expected elements loaded fine on those pages. The theme toggle also worked without any issues, and clicking the Quest AI logo brought me back to the Login page as it should. Overall, the navigation flow for a logged-out user works smoothly.

## Navbar\_LoggedIn





The logged-in navbar was tested using simple click-based Selenium IDE commands. Each navbar item (Home, Create Story, Public Stories, Theme Toggle, Profile Panel, Logout) responded correctly during the test. Navigation transitions worked smoothly without failures. Overall, the logged-in navbar flow is functioning as expected.

## About Page

The About page was successfully reached through the Navbar, and navigation worked correctly. There are no interactive buttons or inputs on the About page, so no further testing is needed.

## Edit Profile Page

Extension: (Selenium IDE) - Selenium IDE - SE\_QuestAI\_GUI\* — Mozilla Firefox

Project: SE\_QuestAI\_GUI\*

Executing ▾

✓ Edit\_Profile\*

https://quest-ai-frontend.vercel.app/

Command	Target	Value
1 ✓ open	https://quest-ai-frontend.vercel.app/Home/Tester	
2 ✓ set window size	1204x680	
3 ✓ click	css=.hidden	
4 ✓ click	css=.w-10 > .w-full	
5 ✓ mouse over	css=.w-10 > .w-full	

Command:  //

Target:

Value:

Description:

Runs: 1 Failures: 0

Log Reference

08:44:15 'Navbar\_LoggedIn' completed successfully

08:52:46 Running 'Edit\_Profile'

08:52:46 1. open on https://quest-ai-frontend.vercel.app/Home/Tester OK

08:52:46 2. setWindowSize on 1204x680 OK

08:52:46 3. click on css=.hidden OK

08:52:46 4. click on css=.w-10 > .w-full OK

08:52:47 5. mouseOver on css=.w-10 > .w-full OK

08:52:47 6. mouseOut on css=.w-10 > .w-full OK

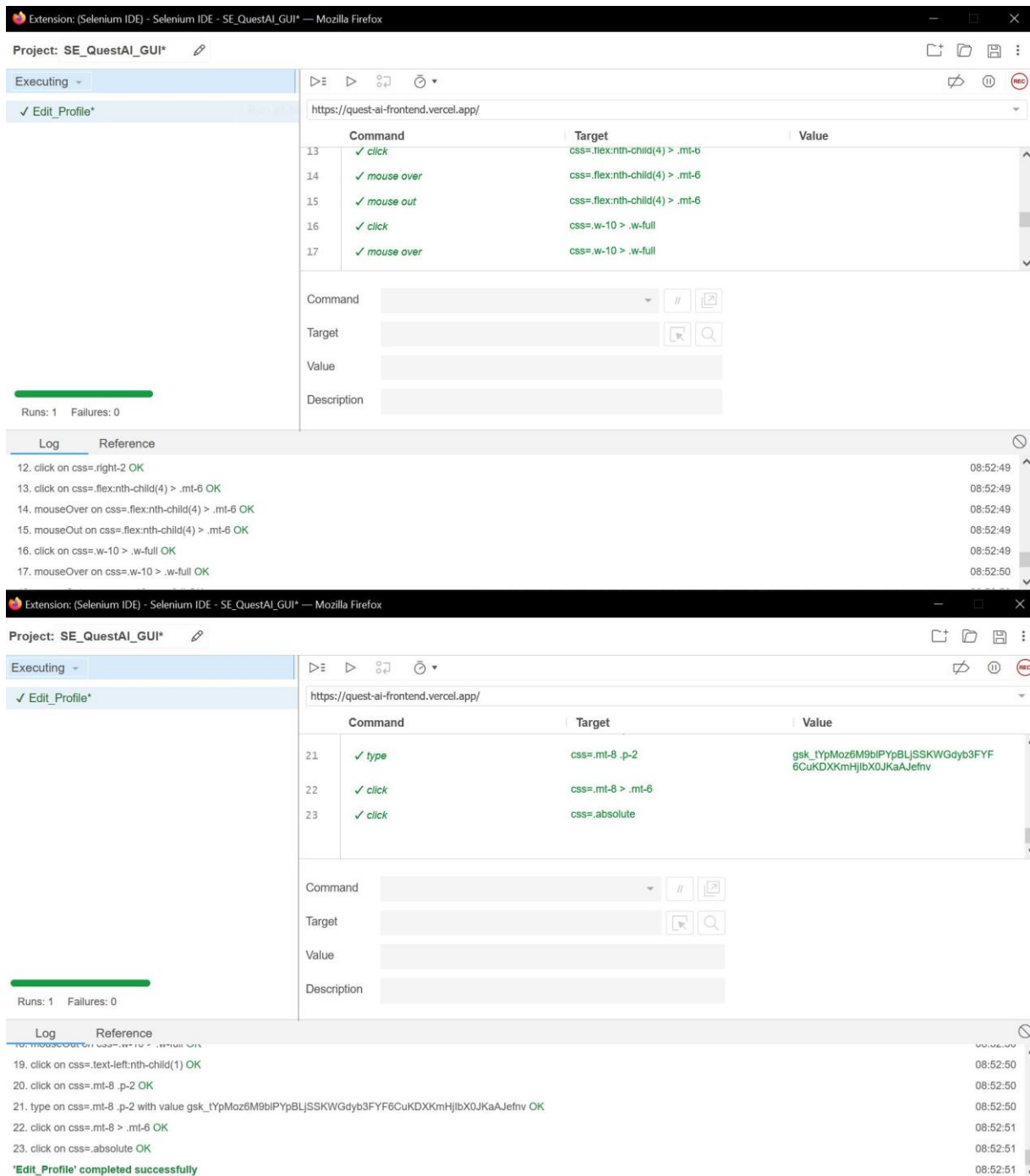
08:52:48 7. click on css=.text-left:nth-child(1) OK

08:52:48 8. Trying to find css=.flex-col > .p-2... OK

08:52:48 9. type on css=.flex-col > .p-2 with value Tester1 OK

08:52:48 10. click on css=.flex:nth-child(4) > .w-full > .p-2 OK

08:52:49 11. type on css=.flex:nth-child(4) > .w-full > .p-2 with value 123456 OK



The Edit Profile flow was tested successfully using Selenium IDE through the recording method.

The profile-picture change feature was also triggered in Selenium, but since Selenium IDE cannot verify image previews or file rendering, the profile-picture update was manually tested and confirmed to work properly.

Overall, all Edit Profile functionalities behaved as expected.

## Story form and Chatbox

Extension: (Selenium IDE) - Selenium IDE - SE\_QuestAI\_GUI\* — Mozilla Firefox

Project: SE\_QuestAI\_GUI\*

Tests ▾ + ▶ ⌵ ⌚ ▾

Search tests... Add new test ps://quest-ai-frontend.vercel.app/

	Command	Target	Value
✓ Edit_Profile			
ForgetPassword_Login_ResetPassword			
✓ Navbar_LoggedIn			
Navbar_LoggedOut			
Sign_up&OTP			
✓ Story_Form*			
Successful_Login			

Command	Target	Value
1 ✓ open	https://quest-ai-frontend.vercel.app/Home/Emperor	
2 ✓ set window size	1204x680	
3 ✓ click	css=.hidden > .form-button:nth-child(5)	
4 ✓ mouse over	css=.hidden > .form-button:nth-child(5)	
5 ✓ click	name=title	

Command  //

Target

Value

Description

Log	Reference
2. setWindowSize on 1204x680 OK	09:34:45
3. click on css=.hidden > .form-button:nth-child(5) OK	09:34:45
4. mouseOver on css=.hidden > .form-button:nth-child(5) OK	09:34:47
5. click on name=title OK	09:34:47
6. sendKeys on name=title with value \${KEY_DOWN} OK	09:34:47
7. type on name=title with value The avengers OK	09:34:47

Extension: (Selenium IDE) - Selenium IDE - SE\_QuestAI\_GUI\* — Mozilla Firefox

Project: SE\_QuestAI\_GUI\*

Tests ▾ + ▶ ⌵ ⌚ ▾

Search tests... Run current test Ctrl+R 1.vercel.app/

	Command	Target	Value
✓ Edit_Profile			
ForgetPassword_Login_ResetPassword			
✓ Navbar_LoggedIn			
Navbar_LoggedOut			
Sign_up&OTP			
✓ Story_Form*			
Successful_Login			

Command	Target	Value
7 ✓ type	name=title	The avengers
8 ✓ click	name=setting	
9 ✓ send keys	name=setting	\$(KEY_DOWN)
10 ✓ type	name=setting	Thanos got all the infinity stones
11 ✓ click	name=character	

Command  //

Target

Value

Description

Log	Reference
9. sendKeys on name=setting with value \$(KEY_DOWN) OK	09:34:48
10. type on name=setting with value Thanos got all the infinity stones OK	09:34:48
11. click on name=character OK	09:34:48
12. sendKeys on name=character with value \$(KEY_DOWN) OK	09:34:48
13. type on name=character with value Iron Man OK	09:34:48
14. click on name=genre OK	09:34:49

Extension: (Selenium IDE) - Selenium IDE - SE\_QuestAI\_GUI\* — Mozilla Firefox

Project: SE\_QuestAI\_GUI\*

Tests + [Run] [Stop] [Refresh] [Reset]

Search tests... Run current test Ctrl+R 1.vercel.app/

Test Case	Step	Command	Target	Value
✓ Edit_Profile	13	✓ type	name=character	Iron Man
	14	✓ click	name=genre	
	15	✓ send keys	name=genre	\$(KEY_DOWN)
	16	✓ type	name=genre	Sci-fi
✓ Story_Form*	17	✓ click	css=.mt-6	

Log Reference

- 15. sendKeys on name=genre with value \$(KEY\_DOWN) OK 09:34:49
- 16. type on name=genre with value Sci-fi OK 09:34:49
- 17. click on css=.mt-6 OK 09:34:49
- 18. mouseOver on css=.mt-6 OK 09:34:49
- 19. mouseOut on css=.mt-6 OK 09:34:50
- 20. click on css=.w-10 > .w-full OK 09:34:50
- 21. mouseOver on css=.w-10 > .w-full OK 09:34:50

Extension: (Selenium IDE) - Selenium IDE - SE\_QuestAI\_GUI\* — Mozilla Firefox

Project: SE\_QuestAI\_GUI\*

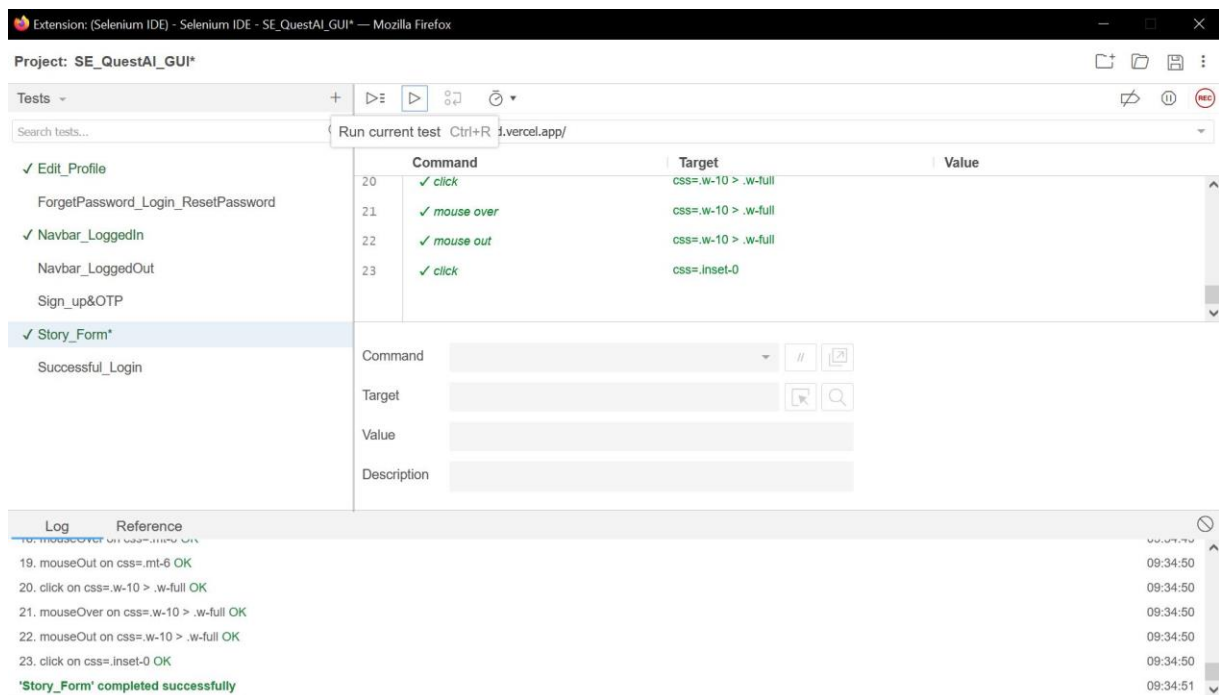
Tests + [Run] [Stop] [Refresh] [Reset]

Search tests... Run current test Ctrl+R 1.vercel.app/

Test Case	Step	Command	Target	Value
✓ Edit_Profile	17	✓ click	css=.mt-6	
	18	✓ mouse over	css=.mt-6	
	19	✓ mouse out	css=.mt-6	
	20	✓ click	css=.w-10 > .w-full	
✓ Story_Form*				

Log Reference

- 19. mouseOut on css=.mt-6 OK 09:34:50
- 20. click on css=.w-10 > .w-full OK 09:34:50
- 21. mouseOver on css=.w-10 > .w-full OK 09:34:50
- 22. mouseOut on css=.w-10 > .w-full OK 09:34:50
- 23. click on css=.inset-0 OK 09:34:50
- \*Story\_Form\* completed successfully 09:34:51



The Story Form page was tested successfully using Selenium IDE through the recording method.

Entering the title, setting, character, and genre worked perfectly, and the form submission flow was captured without failures.

However, the Chatbox page could not be automated in Selenium, since it requires AI responses, dynamic message rendering, and scrolling—elements that Selenium IDE cannot reliably detect.

Therefore, the Chatbox was manually tested end-to-end, including sending prompts, receiving responses, copying messages, and completing/continuing a story.

All Chatbox functionalities worked correctly during manual testing.



## Public Story

Extension: (Selenium IDE) - Selenium IDE - SE\_QuestAI\_GUI\* — Mozilla Firefox

Project: SE\_QuestAI\_GUI\*

Tests ▾ + ▶ ⌂ ⌚

Search tests... Run current test Ctrl+R 1.vercel.app/

Command	Target	Value
✓ open	https://quest-ai-frontend.vercel.app/	
✓ set window size	1204x680	
✓ mouse over	css=.hidden > .form-button:nth-child(5)	
✓ click	css=.hidden > .form-button:nth-child(2)	
✓ click	css=.fa-play > path	

Log Reference

Log	Reference
2. setWindowSize on 1204x680 OK	10:10:23
3. mouseOver on css=.hidden > .form-button:nth-child(5) OK	10:10:23
4. click on css=.hidden > .form-button:nth-child(2) OK	10:10:23
5. Trying to find css=.fa-play > path... OK	10:10:24
6. Trying to find css=.span... OK	10:10:25
7. click on css=.hidden > .form-button:nth-child(2) OK	10:10:25
8. mouseOver on css=.hidden > .form-button:nth-child(2) OK	10:10:26

Extension: (Selenium IDE) - Selenium IDE - SE\_QuestAI\_GUI\* — Mozilla Firefox

Project: SE\_QuestAI\_GUI\*

Tests ▾ + ▶ ⌂ ⌚

Search tests... Run current test Ctrl+R 1.vercel.app/

Command	Target	Value
7. ✓ click	css=.hidden > .form-button:nth-child(2)	
8. ✓ mouse over	css=.hidden > .form-button:nth-child(2)	
9. ✓ click	css=.fa-download > path	
10. ✓ click	css=.gap-6	

Log Reference

Log	Reference
6. Trying to find css=.span... OK	10:10:25
7. click on css=.hidden > .form-button:nth-child(2) OK	10:10:25
8. mouseOver on css=.hidden > .form-button:nth-child(2) OK	10:10:26
9. Trying to find css=.fa-download > path... OK	10:10:26
10. click on css=.gap-6 OK	10:10:27
*Public_Story* completed successfully	

The Public Stories page was tested using Selenium IDE through the recording method, and all recorded interactions worked correctly.

Clicking on a Public Story successfully navigates to the Chatbox in public mode, confirming that story viewing without login works as intended.

The Download Story button was also tested and verified to generate and download the text file properly.

Hover effects, navigation, and icon interactions (Play, Download) were captured and executed successfully in Selenium.

Overall, the Public Stories feature works smoothly.

## Home Page

The screenshot shows the Selenium IDE interface with the project 'SE\_QuestAI\_GUI\*' and the test 'Home\_Page\*' selected. The test script is as follows:

Command	Target	Value
open	https://quest-ai-frontend.vercel.app/Home/Emperor	
set window size	1204x680	
click	css=fa-lock > path	
click	css=fa-lock-open > path	
click	css=fa-user-plus > path	

The Log tab shows the following execution details:

Log	Reference
Running 'Home_Page'	13:59:10
1. open on https://quest-ai-frontend.vercel.app/Home/Emperor OK	13:59:10
2. setWindowSize on 1204x680 OK	13:59:10
3. Trying to find css=fa-lock > path... OK	13:59:10
4. Trying to find css=fa-lock-open > path... OK	13:59:12
5. click on css=fa-user-plus > path OK	13:59:14

The screenshot shows the continuation of the Selenium IDE interface. The test script continues with the following commands:

Command	Target	Value
type	css=input	dharva.1010@gmail.com
click	css=.mt-4 > .form-button	
click	css=fa-play > path	
click	css=.form-button:nth-child(1) > span	
mouse over	css=.hidden > .form-button:nth-child(4)	

The Log tab shows the following execution details:

Log	Reference
6. click on css=input OK	13:59:14
7. type on css=input with value dharva.1010@gmail.com OK	13:59:14
8. click on css=.mt-4 > .form-button OK	13:59:15
9. click on css=fa-play > path OK	13:59:15
10. Trying to find css=.form-button:nth-child(1) > span... OK	13:59:15
11. mouseOver on css=.hidden > .form-button:nth-child(4) OK	13:59:15
'Home_Page' completed successfully	13:59:16

Some buttons on the Home page, especially the ones that open the ChatBox, had to be tested manually because Selenium wasn't able to detect or interact with the ChatBox properly. The rest of the Home page buttons and flows that worked with Selenium were tested using the normal recording method. Both manual and recorded tests were used to make sure the whole Home page works correctly.

## Responsive testing

The responsiveness of the application was tested manually using the browser's Inspect → Device Toolbar. All the main pages were checked across multiple device to ensure proper layout, alignment, and usability.. Screenshots of key pages on different device sizes have been captured and included. The pages are shown below:

