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:

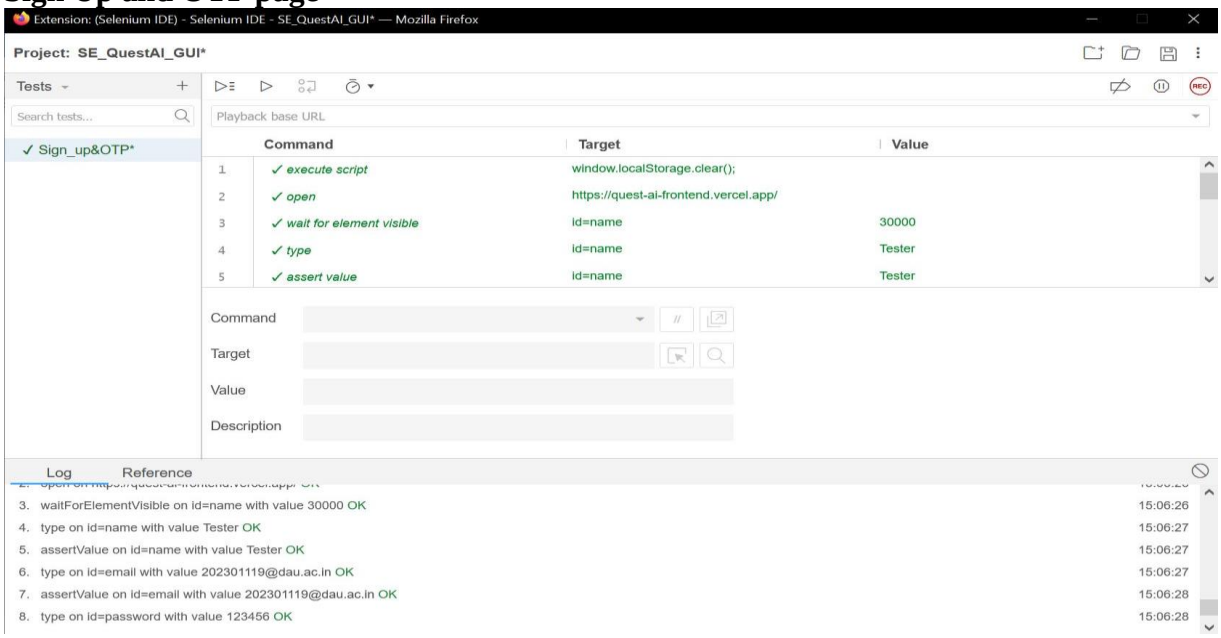
- **Navbar (Logged Out)**
- **Navbar (Logged In)**
- **About**
- **Edit Profile**
- **Create Story Page & Chatbox**
- **Public Stories Page**
- **Home Page(manually)**

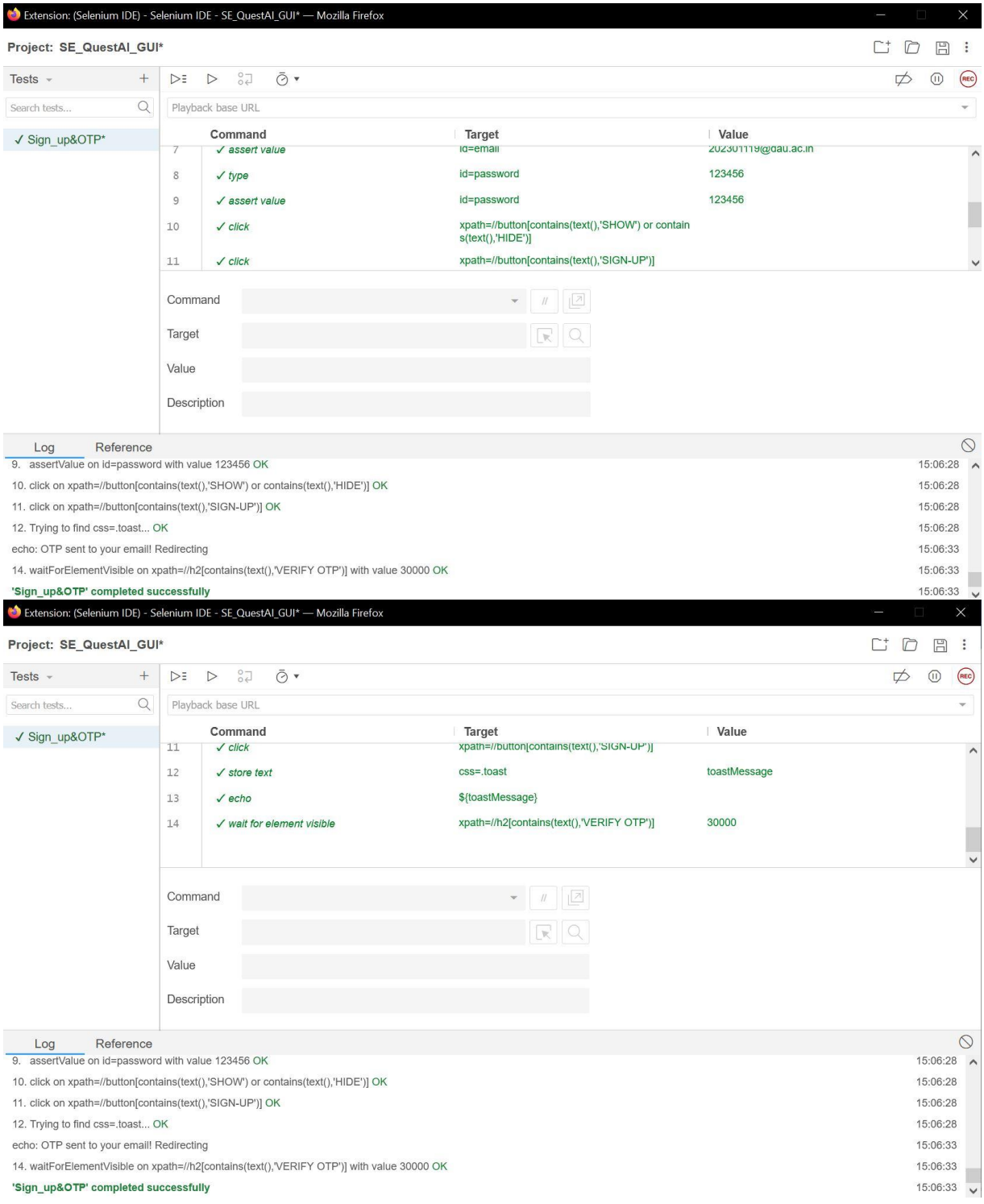
For these pages:

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



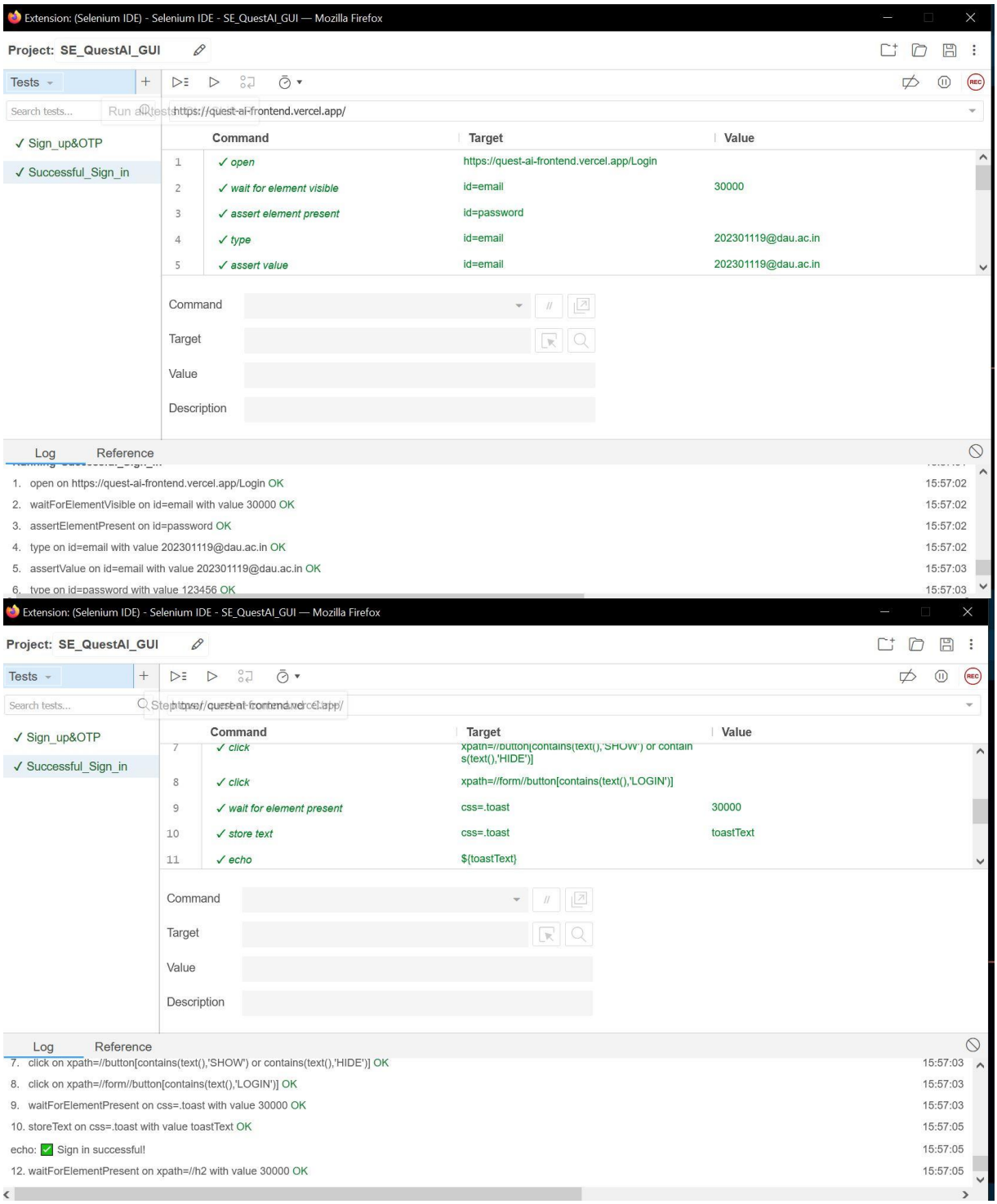


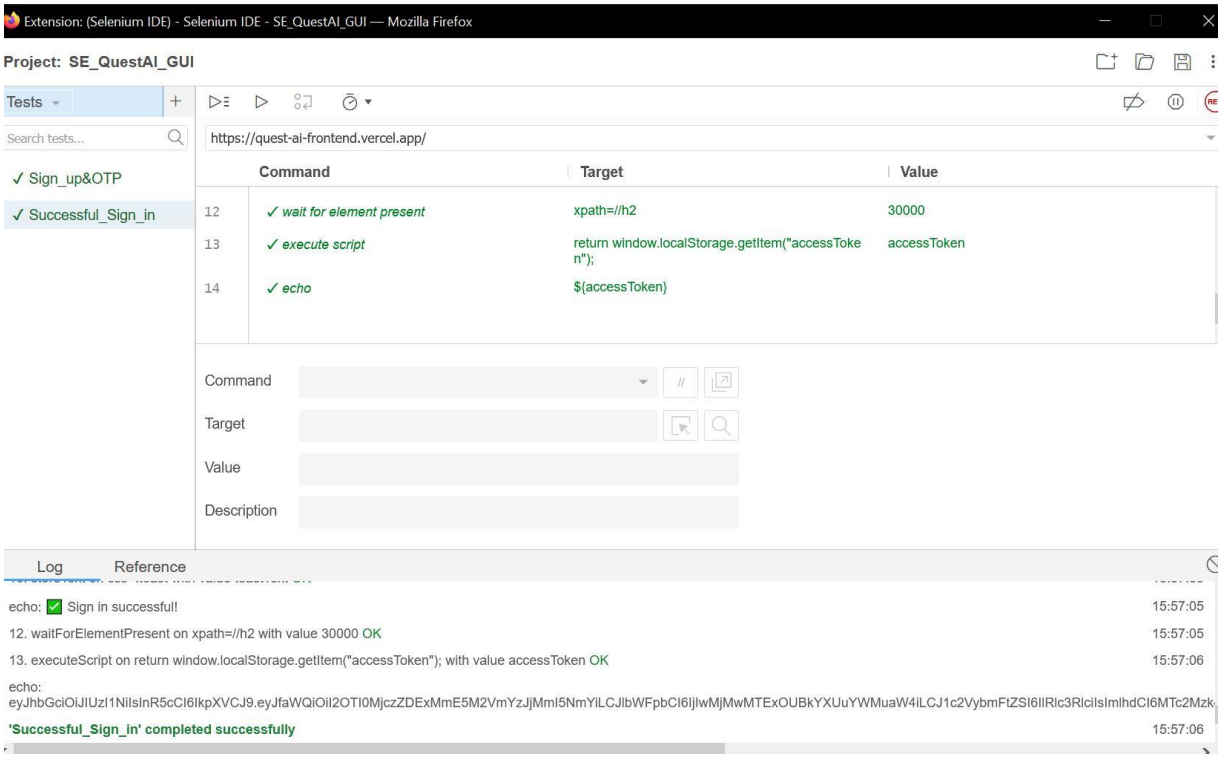
The test confirmed that the Sign-Up page loads properly, takes all the inputs properly, and successfully sends the OTP on the provided email. A toast appears, and the app correctly redirects to the OTP page, so the Sign-Up - OTP flow works as expected.

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.

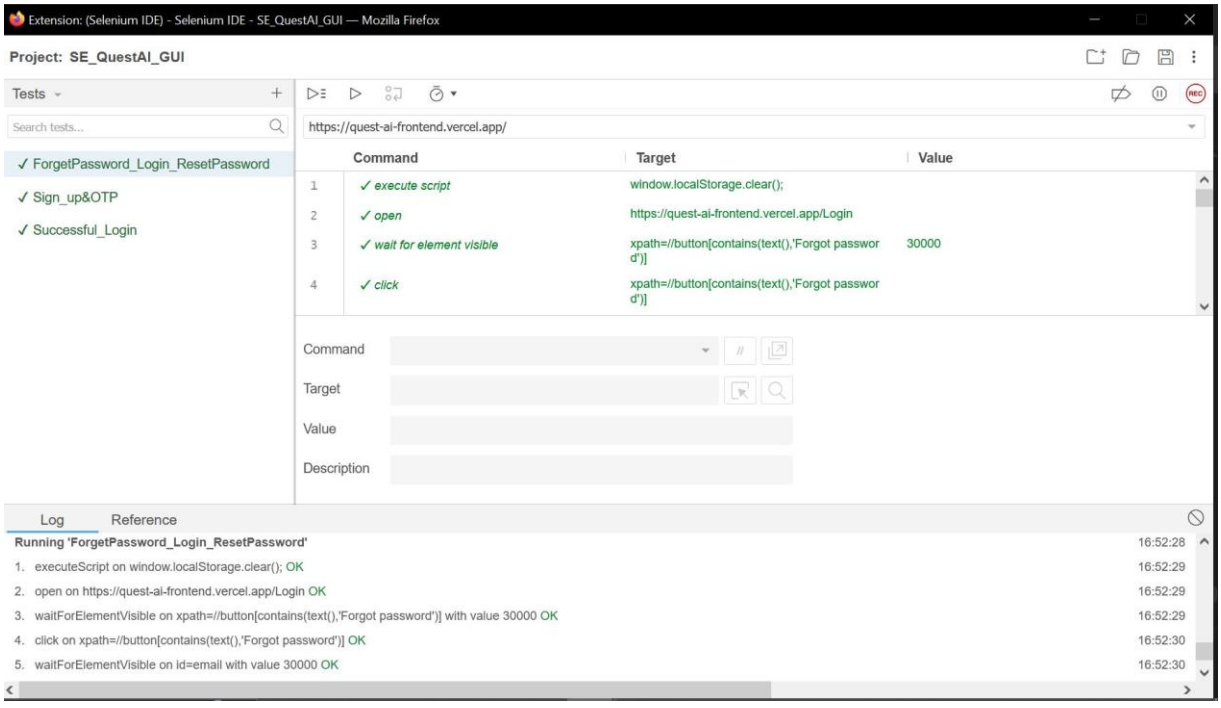
Successful Login

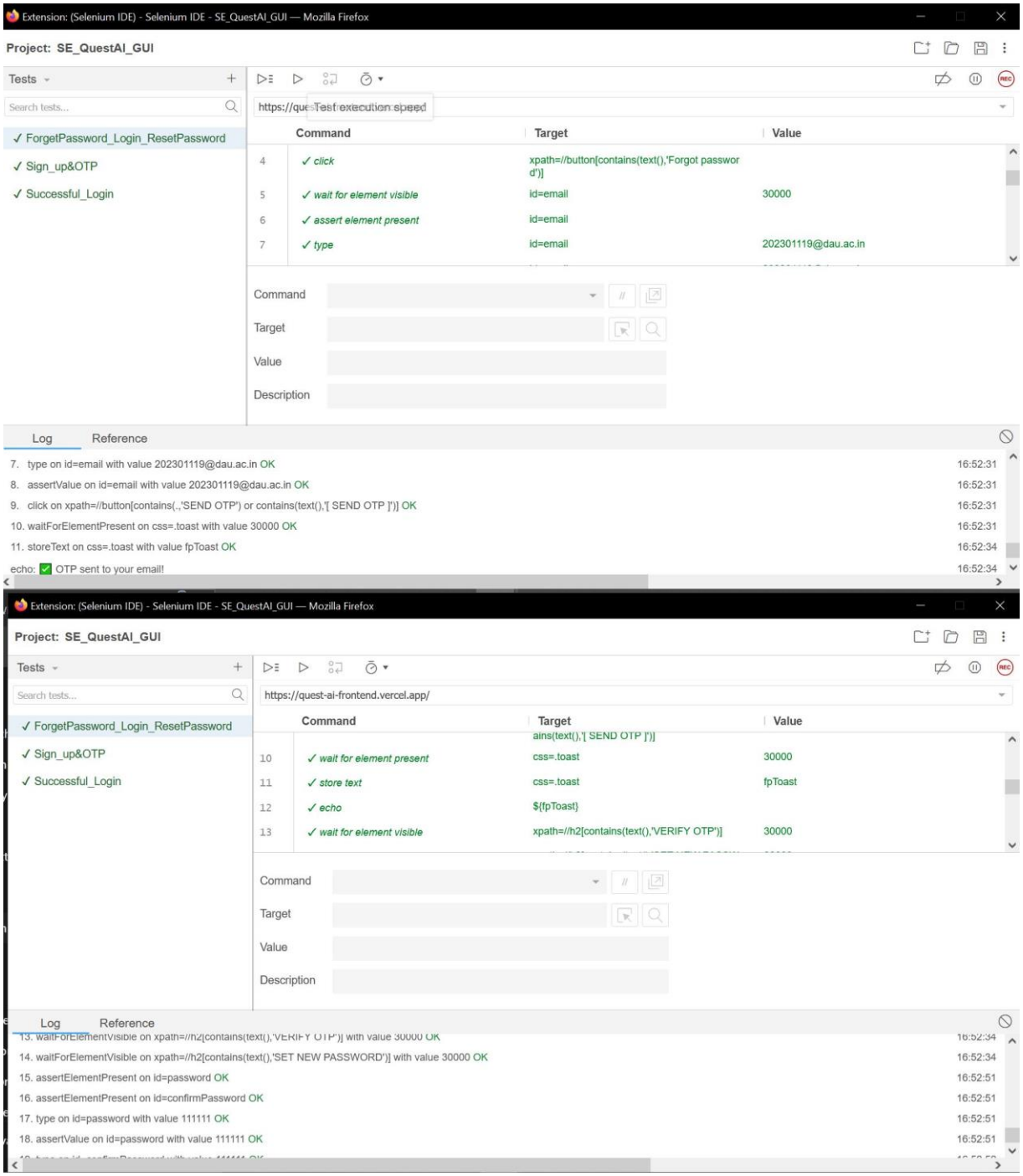




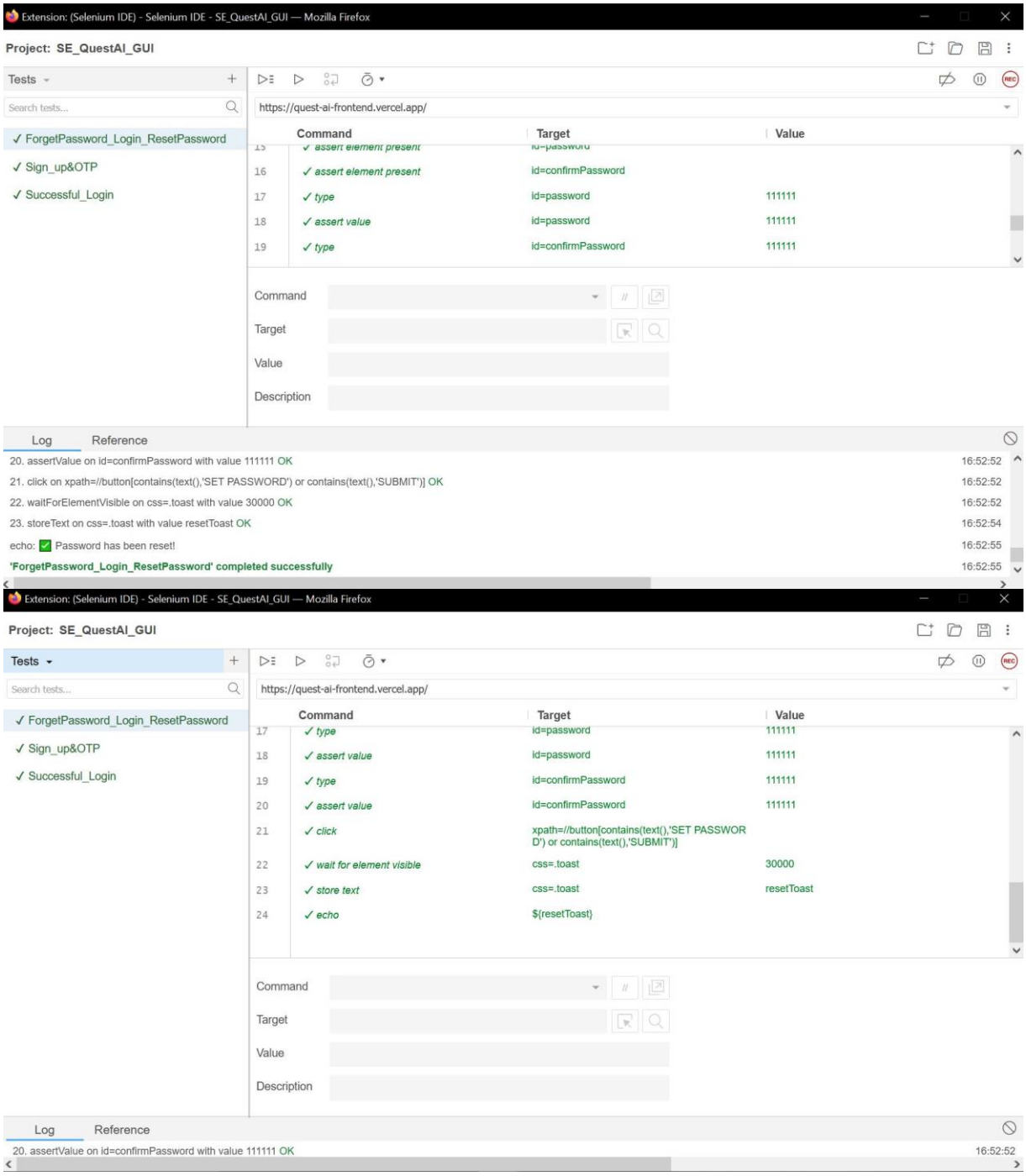
This test checks that the Login page loads correctly and that both the email and password fields are present and can accepts the correct input. After entering the correct credentials, the test clicks the Login button and waits for the success toast message. It then confirms the login and the user has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

ForgetPassword_Login_ResetPassword




Extension: (Selenium IDE) - Selenium IDE - SE_QuestAI_GUI — Mozilla Firefox
Project: SE_QuestAI_GUI
Tests
Search tests...
✓ ForgetPassword_Login_ResetPassword
✓ Sign_up&OTP
✓ Successful_Login
Test execution speed
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

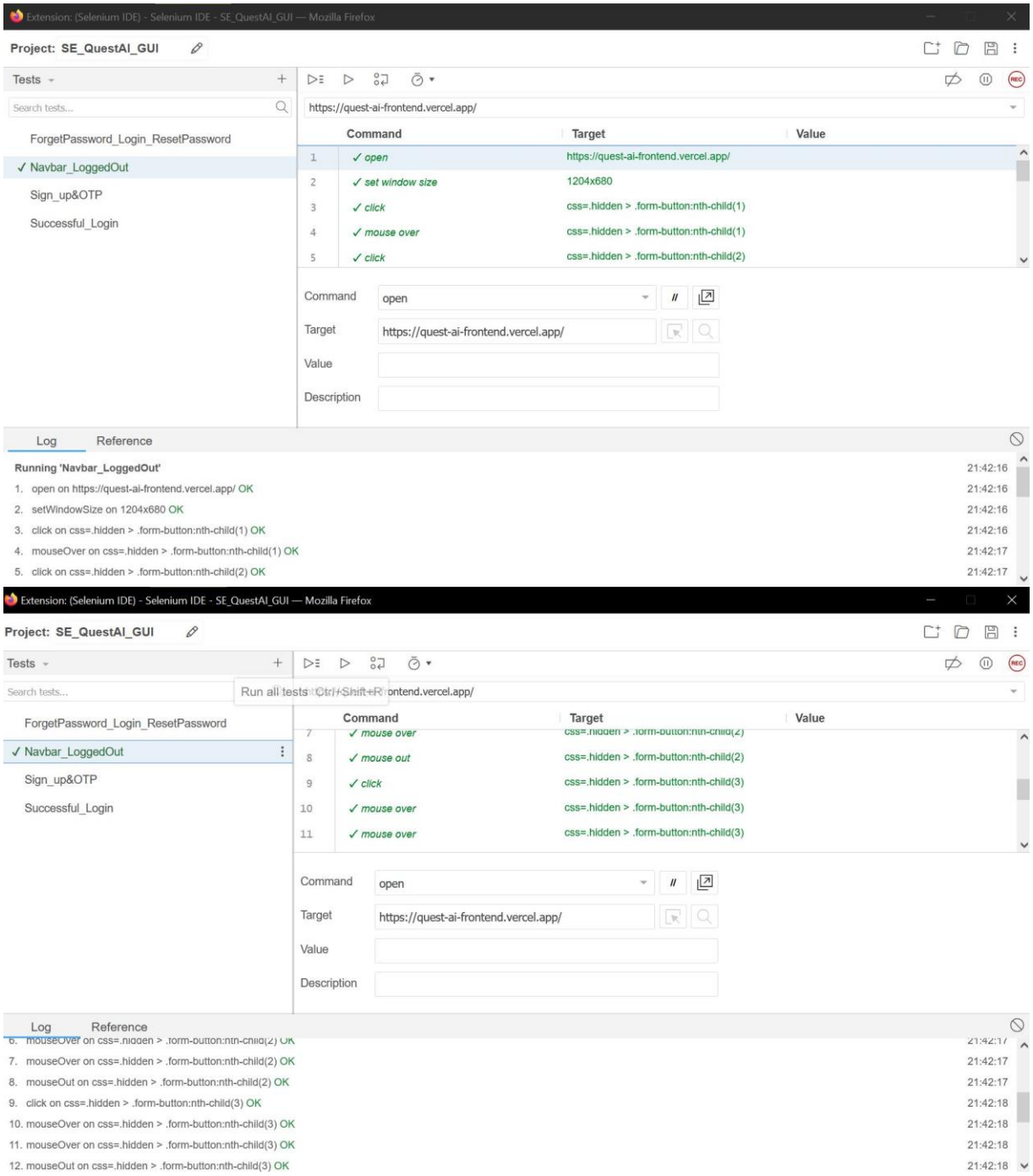
Extension: (Selenium IDE) - Selenium IDE - SE_QuestAI_GUI — Mozilla Firefox
Project: SE_QuestAI_GUI
Tests
Search tests...
✓ ForgetPassword_Login_ResetPassword
✓ Sign_up&OTP
✓ Successful_Login
https://quest-ai-frontend.vercel.app/
Command
Target
Value
10 ✓ wait for element present css=.toast 30000
11 ✓ 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
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

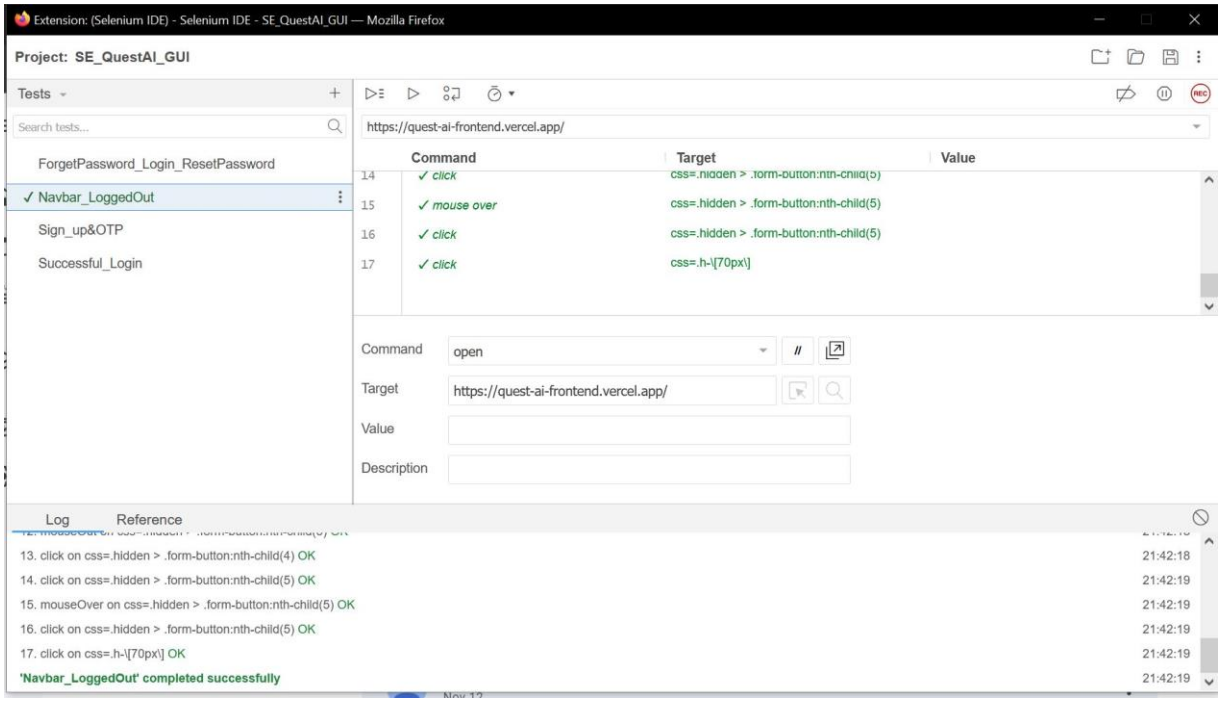


This test covers the full Forgot Password flow starting from the Login page. First, it checks that clicking the “Forgot password?” button correctly opens the Forgot Password page and that the email field accepts a valid email. When the backend sends an OTP, the test also verifies the toast message that 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.

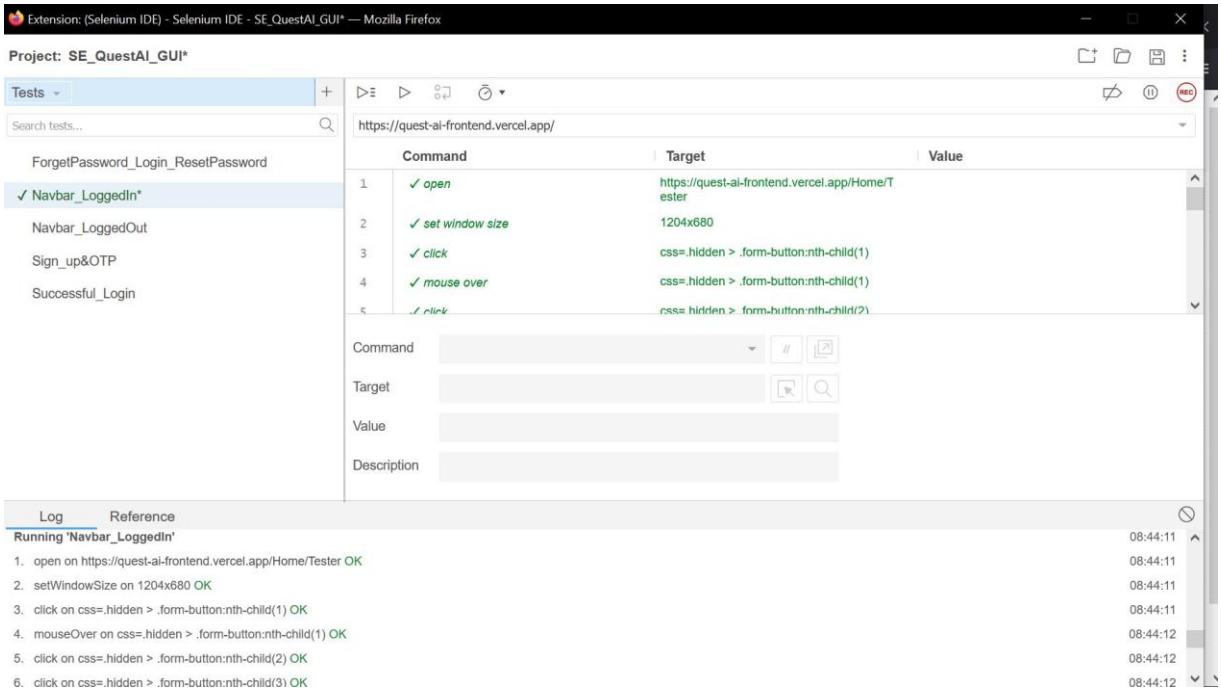
Navbar_LoggedOut

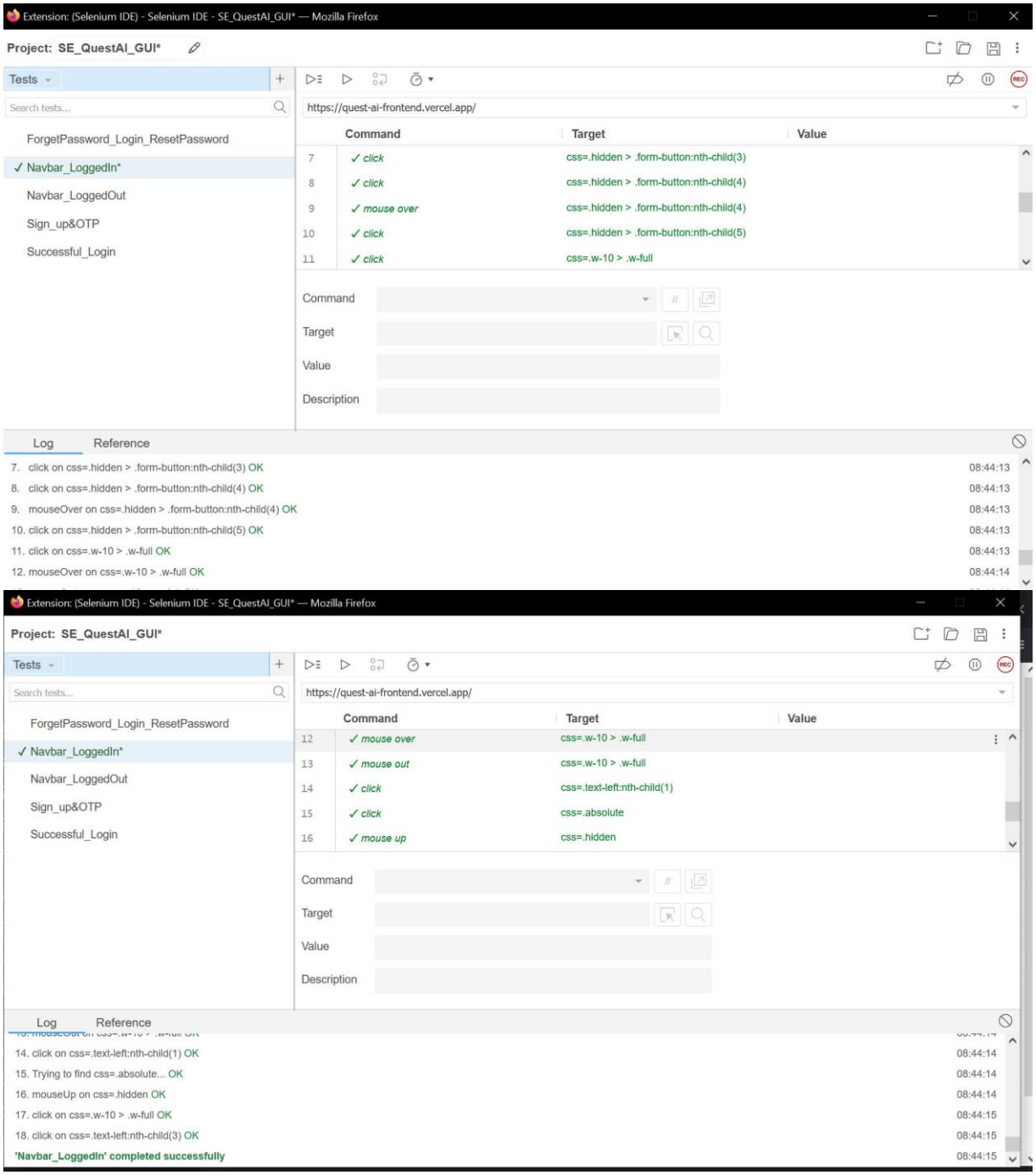




This test checked whether all the Navbar buttons work properly when the user isn't logged in. Starting from the Login page, the about,sign up,login and public stories buttons each opened the pages they were supposed to, and all the expected elements loaded fine on those pages. The theme toggle also worked 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



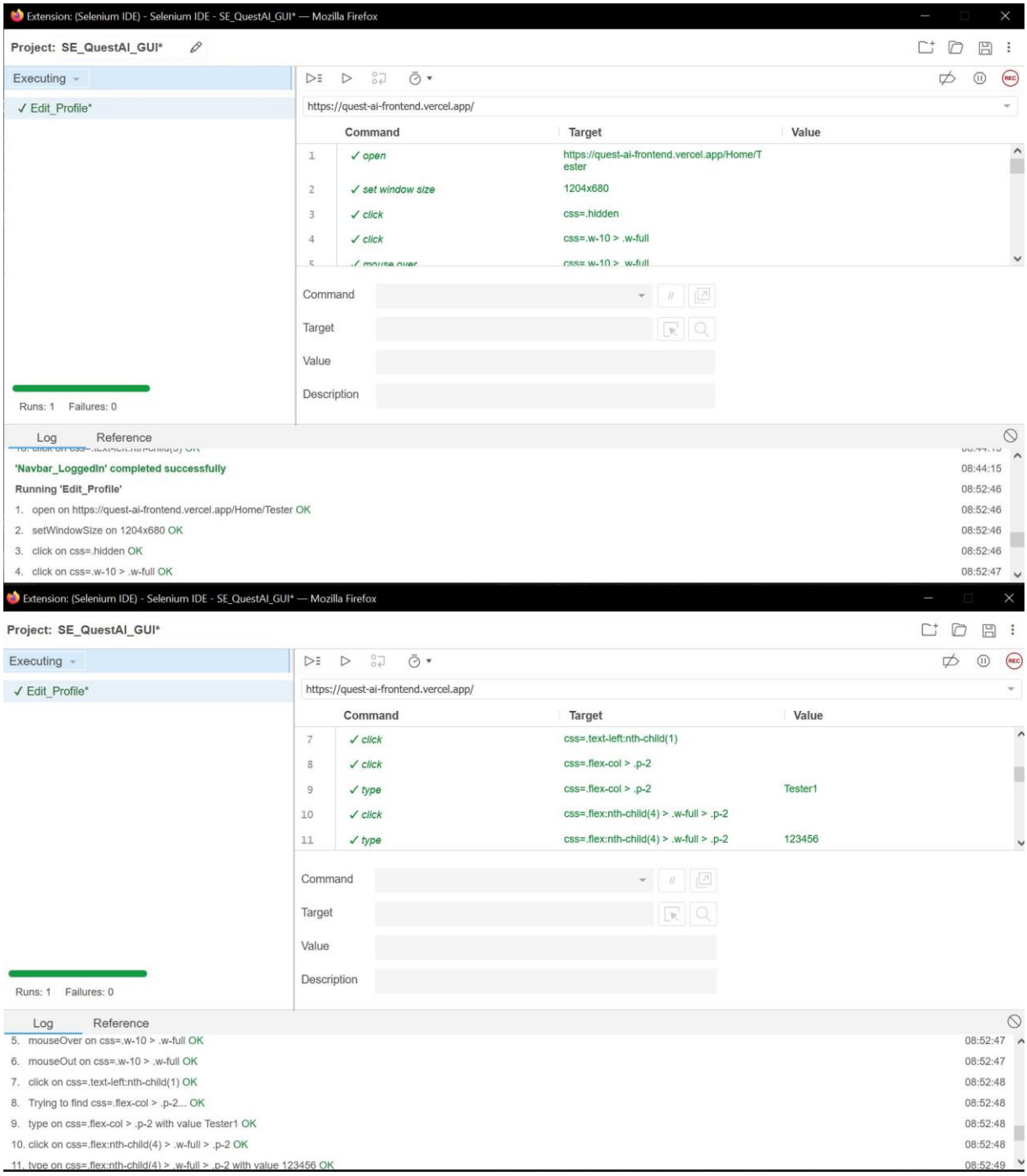


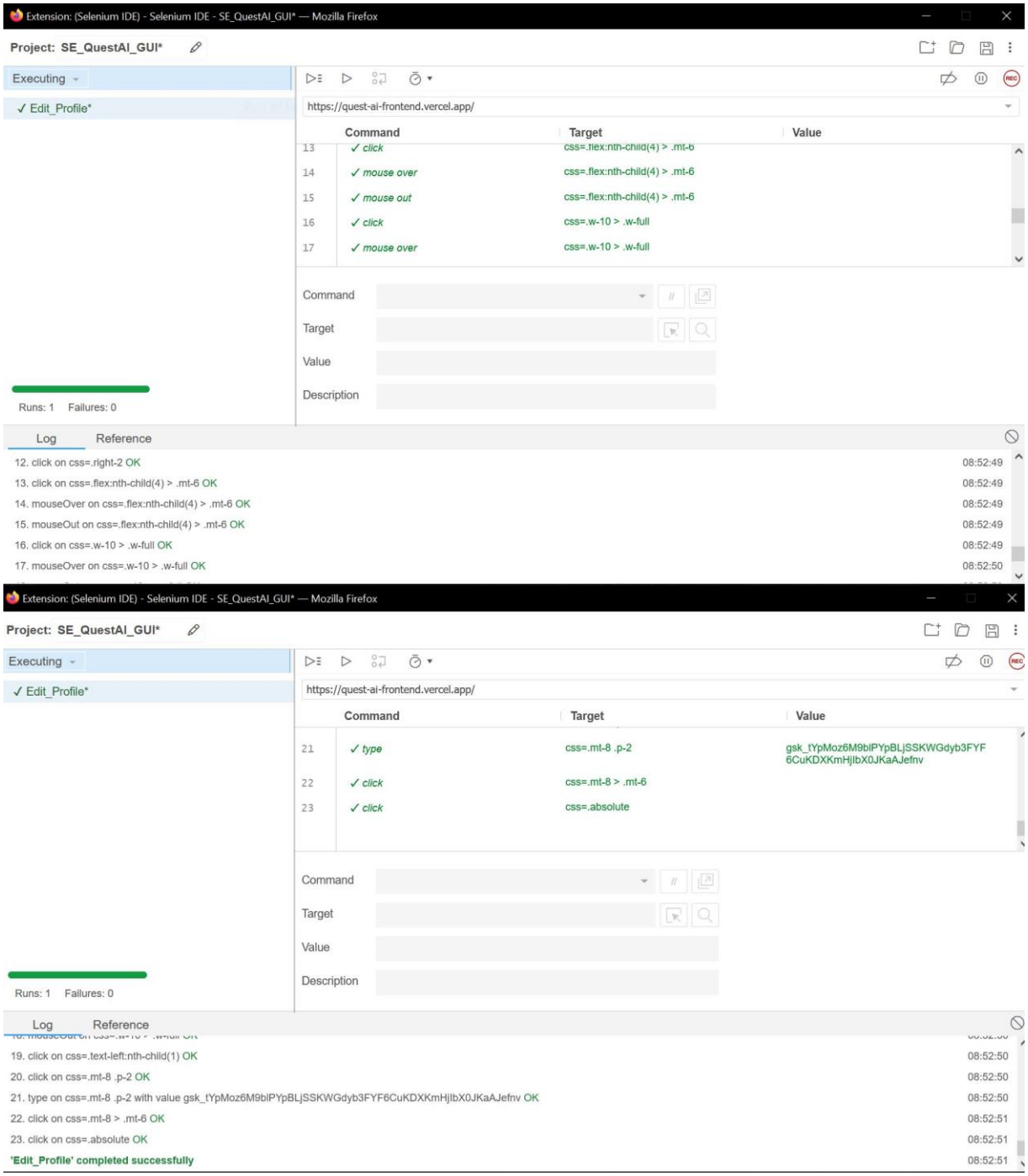
The logged-in navbar was tested using simple click-based Selenium IDE commands.
Each navbar item (Home, Create Story, Public Stories, Theme Toggle, Profile Panel, Logout) responded correctly during the test.
Navigation transitions worked smoothly without failures.
Overall, the logged-in navbar flow is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

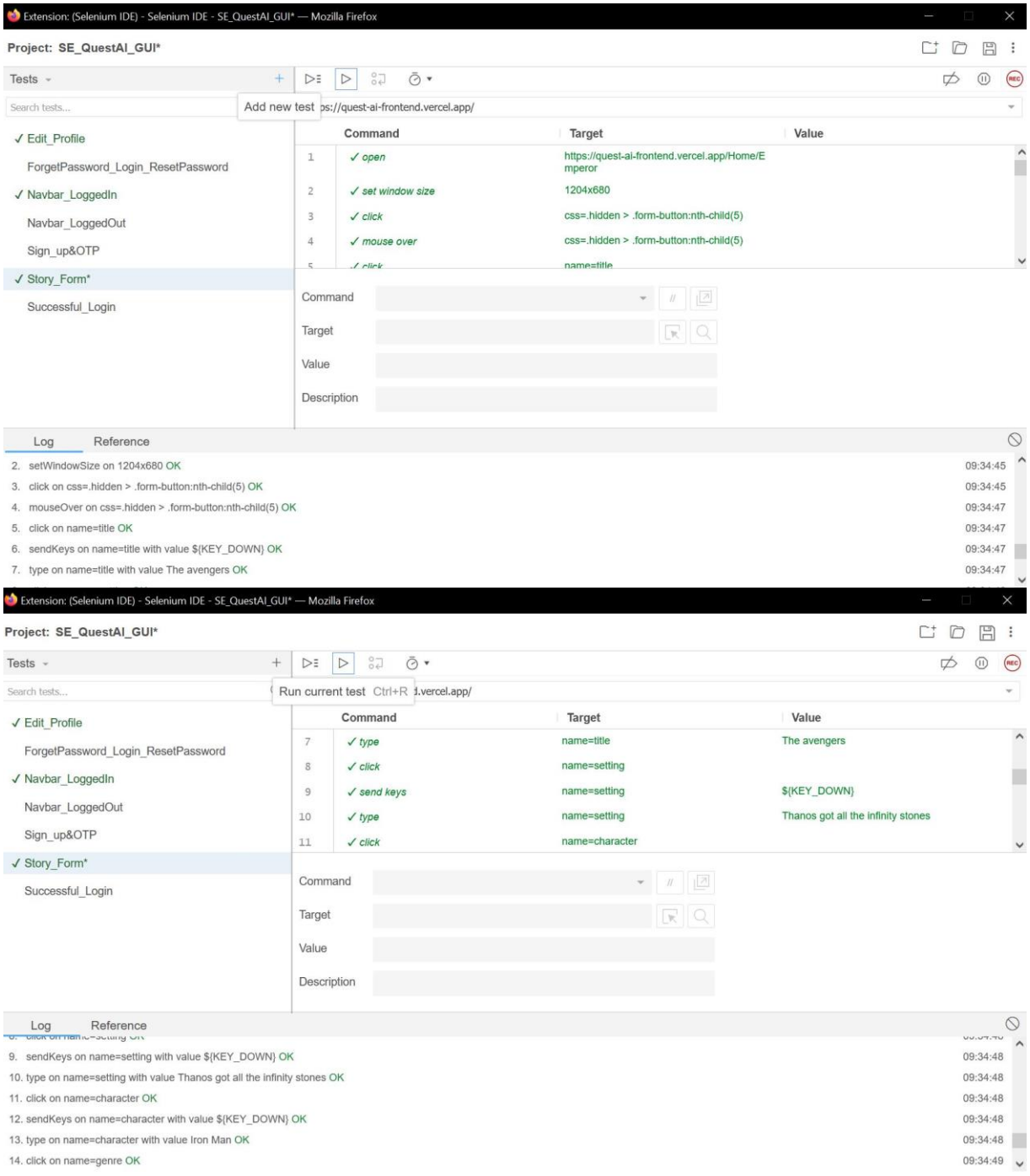




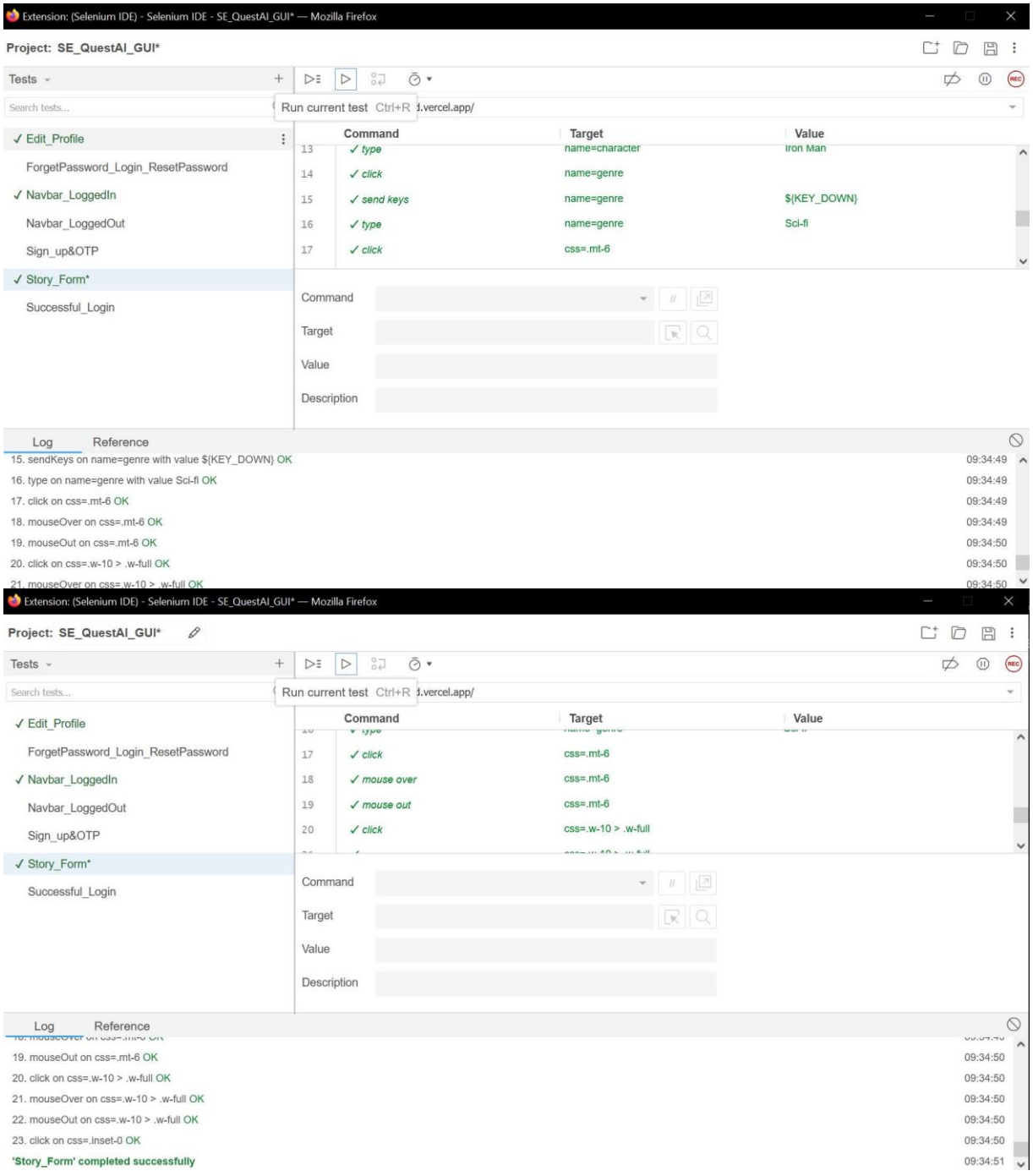
The Edit Profile flow was tested successfully using Selenium IDE through the recording method.

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.

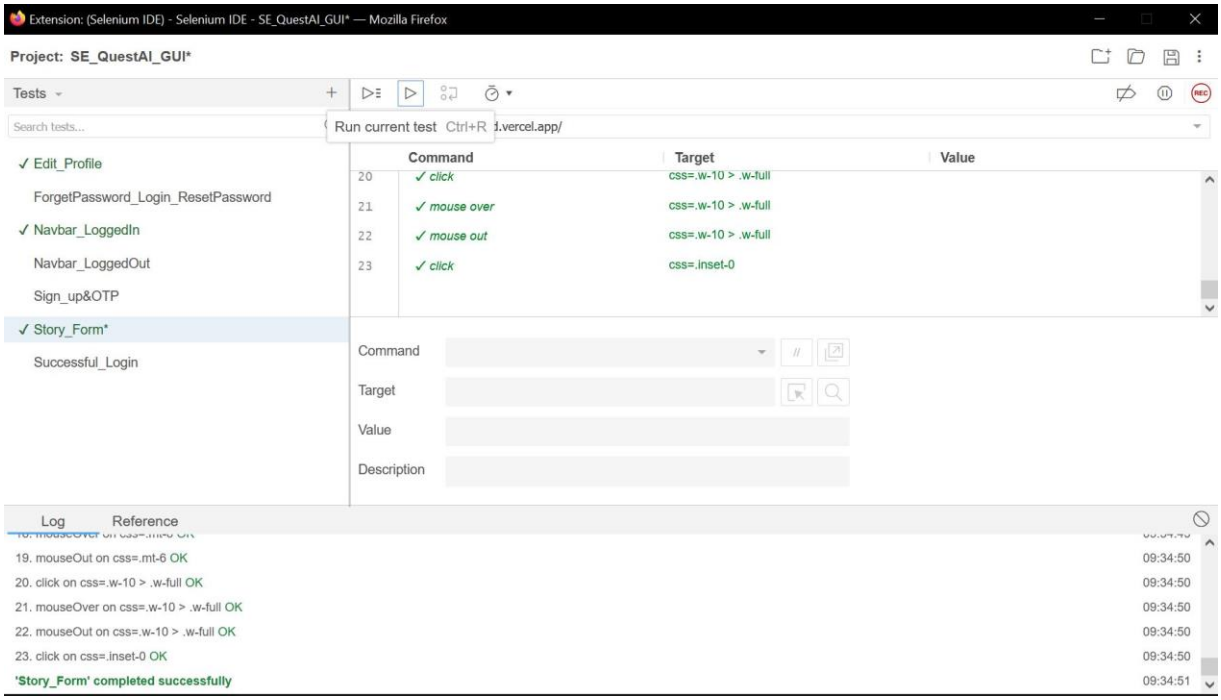
Story form and Chatbox


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/
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
.vercel.app/
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

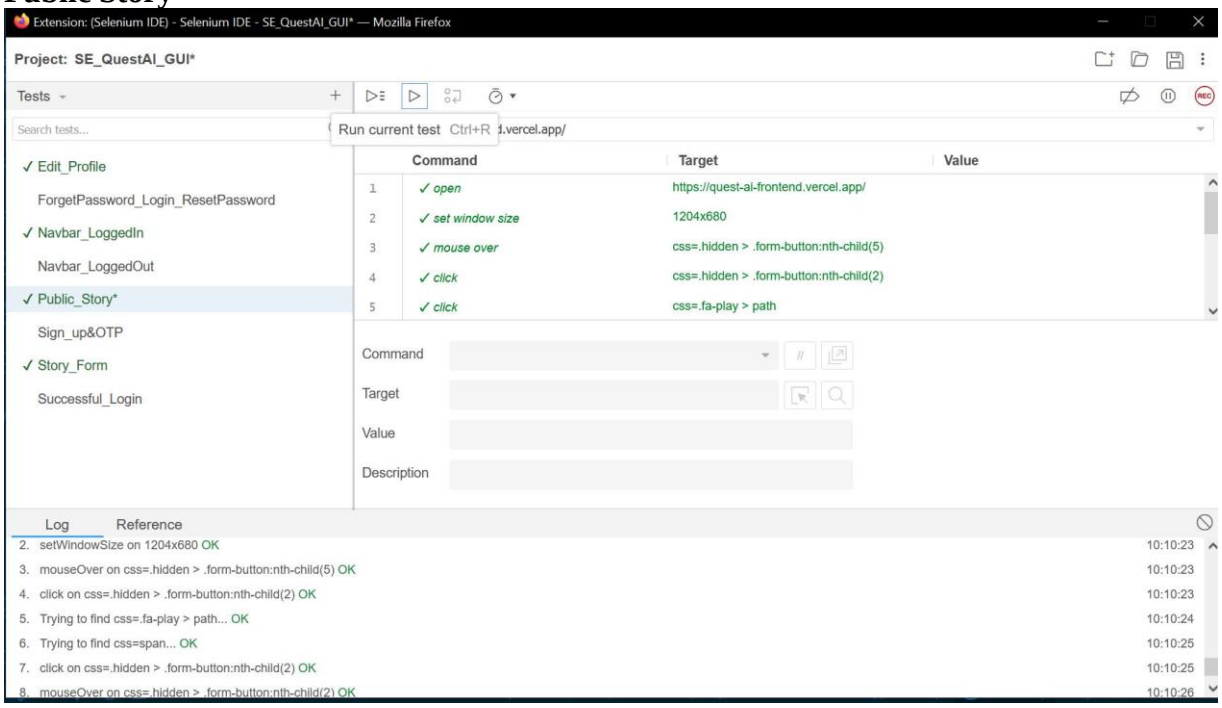

Extension: (Selenium IDE) - Selenium IDE - SE_QuestAI_GUI* — Mozilla Firefox
Project: SE_QuestAI_GUI*
Tests
Search tests...
Run current test Ctrl+R
Edit_Profile
ForgetPassword_Login_ResetPassword
Navbar_LoggedIn
Navbar_LoggedOut
Sign_up&OTP
Story_Form*
Successful_Login
Command
Target
Value
13 type name=character Iron Man
14 click name=genre
15 send keys name=genre ${KEY_DOWN}
16 type name=genre Sci-fi
17 click css=.mt-6
Command
Target
Value
Description
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

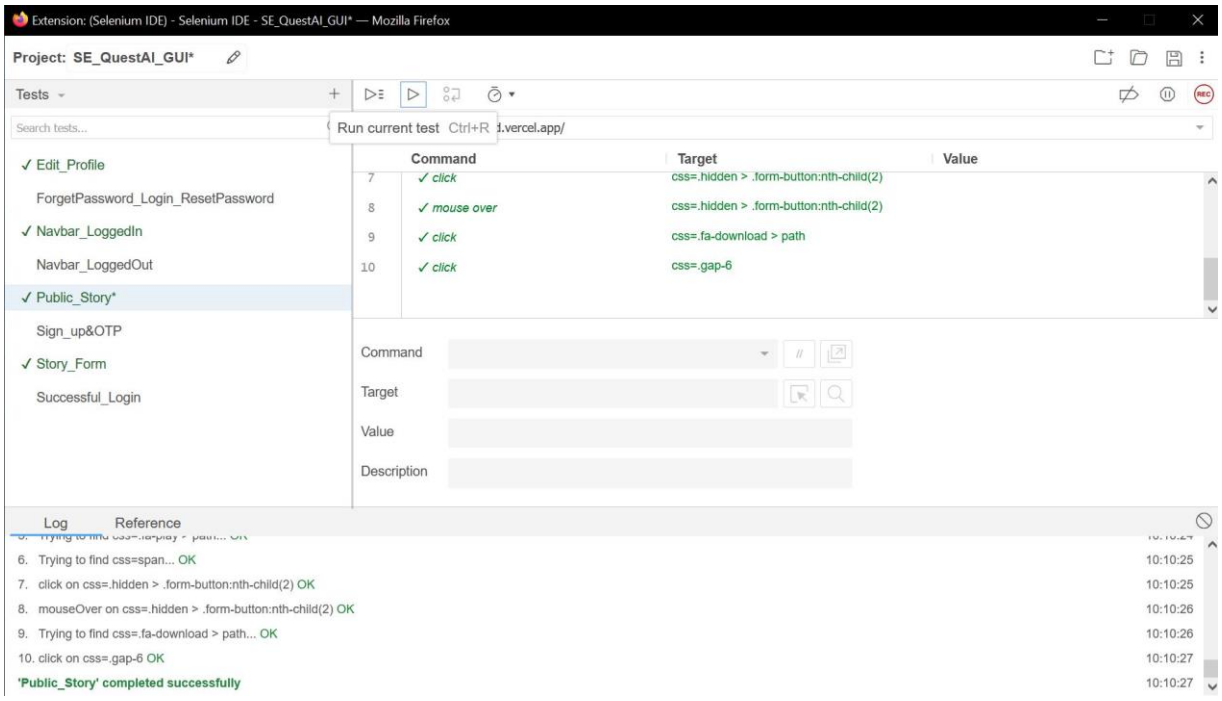
Extension: (Selenium IDE) - Selenium IDE - SE_QuestAI_GUI* — Mozilla Firefox
Project: SE_QuestAI_GUI*
Tests
Search tests...
Run current test Ctrl+R
Edit_Profile
ForgetPassword_Login_ResetPassword
Navbar_LoggedIn
Navbar_LoggedOut
Sign_up&OTP
Story_Form*
Successful_Login
Command
Target
Value
17 click css=.mt-6
18 mouse over css=.mt-6
19 mouse out css=.mt-6
20 click css=.w-10 > .w-full
Command
Target
Value
Description
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.

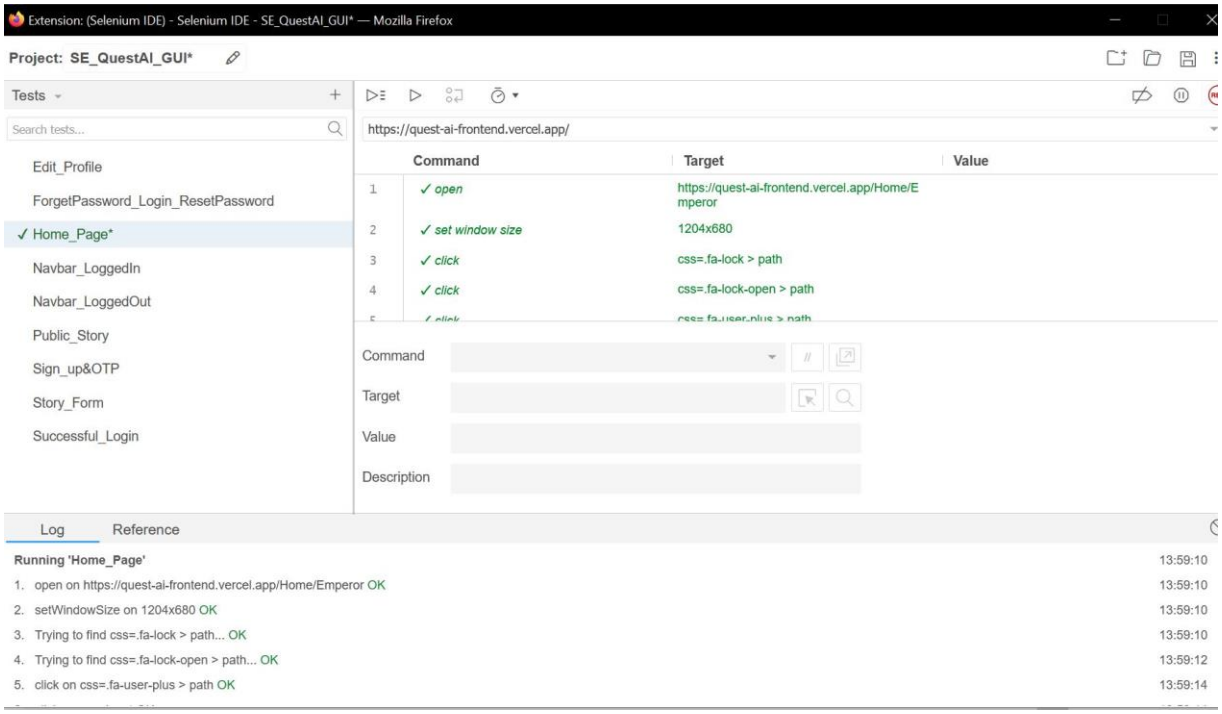
Public Story

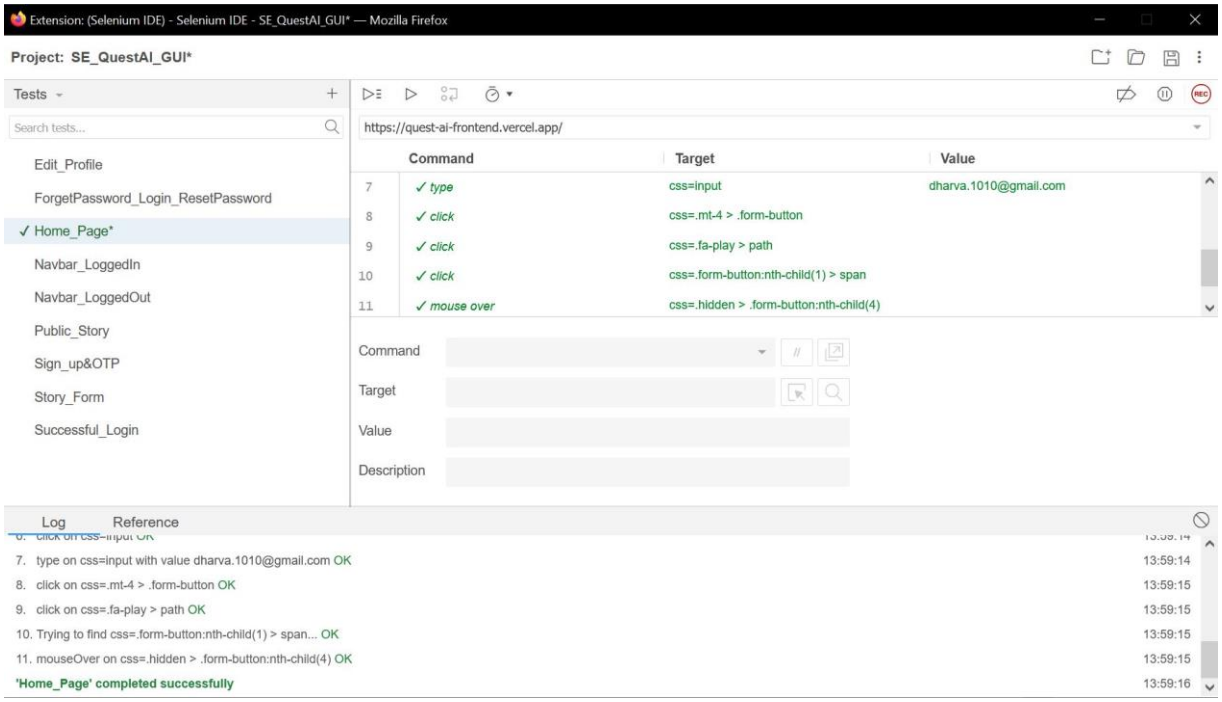




The Public Stories page was tested using Selenium IDE through the recording method, and all recorded interactions worked correctly.
Clicking on a Public Story successfully navigates to the Chatbox in public mode, confirming that story viewing without login works as intended.
The Download Story button 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

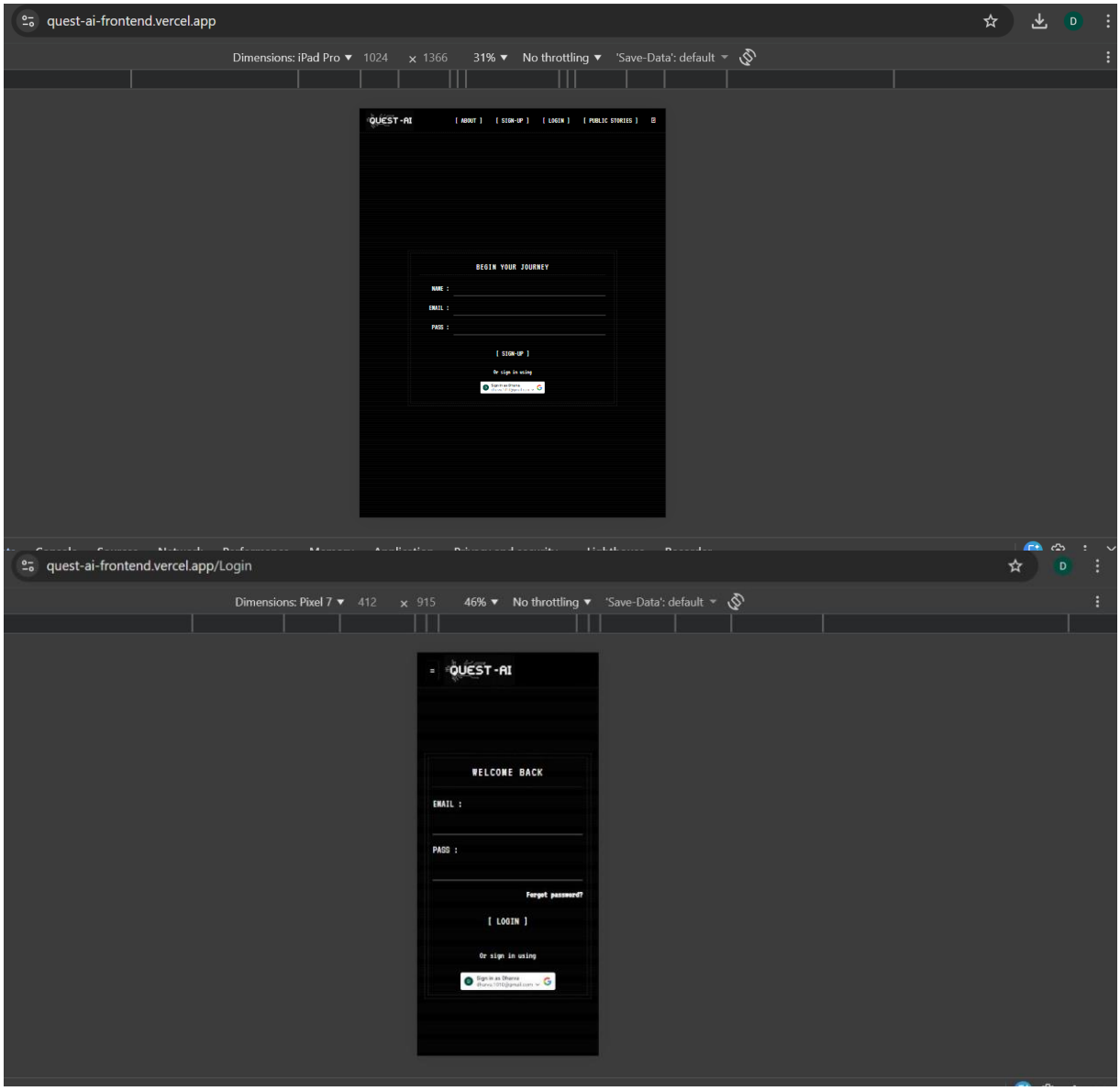


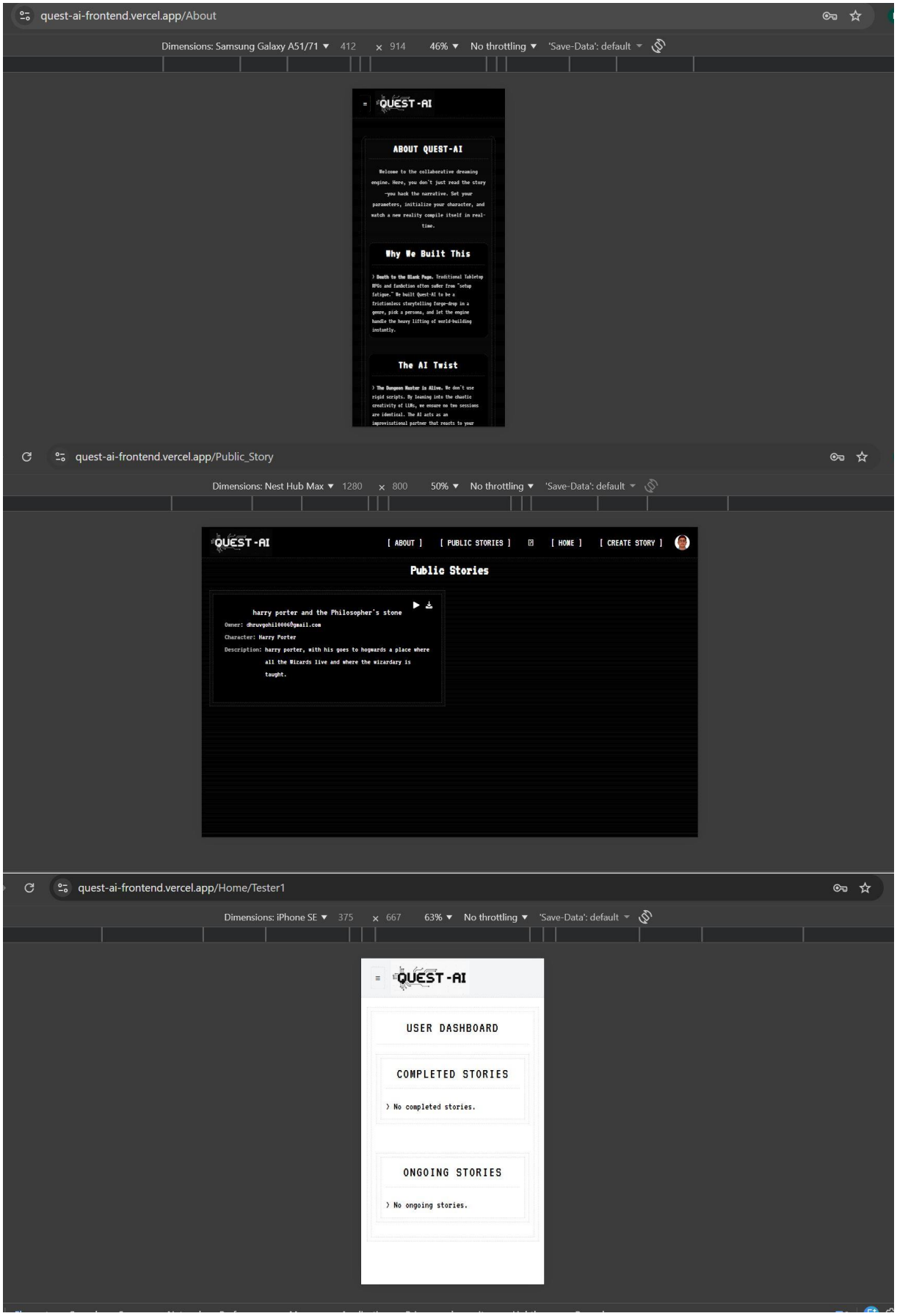


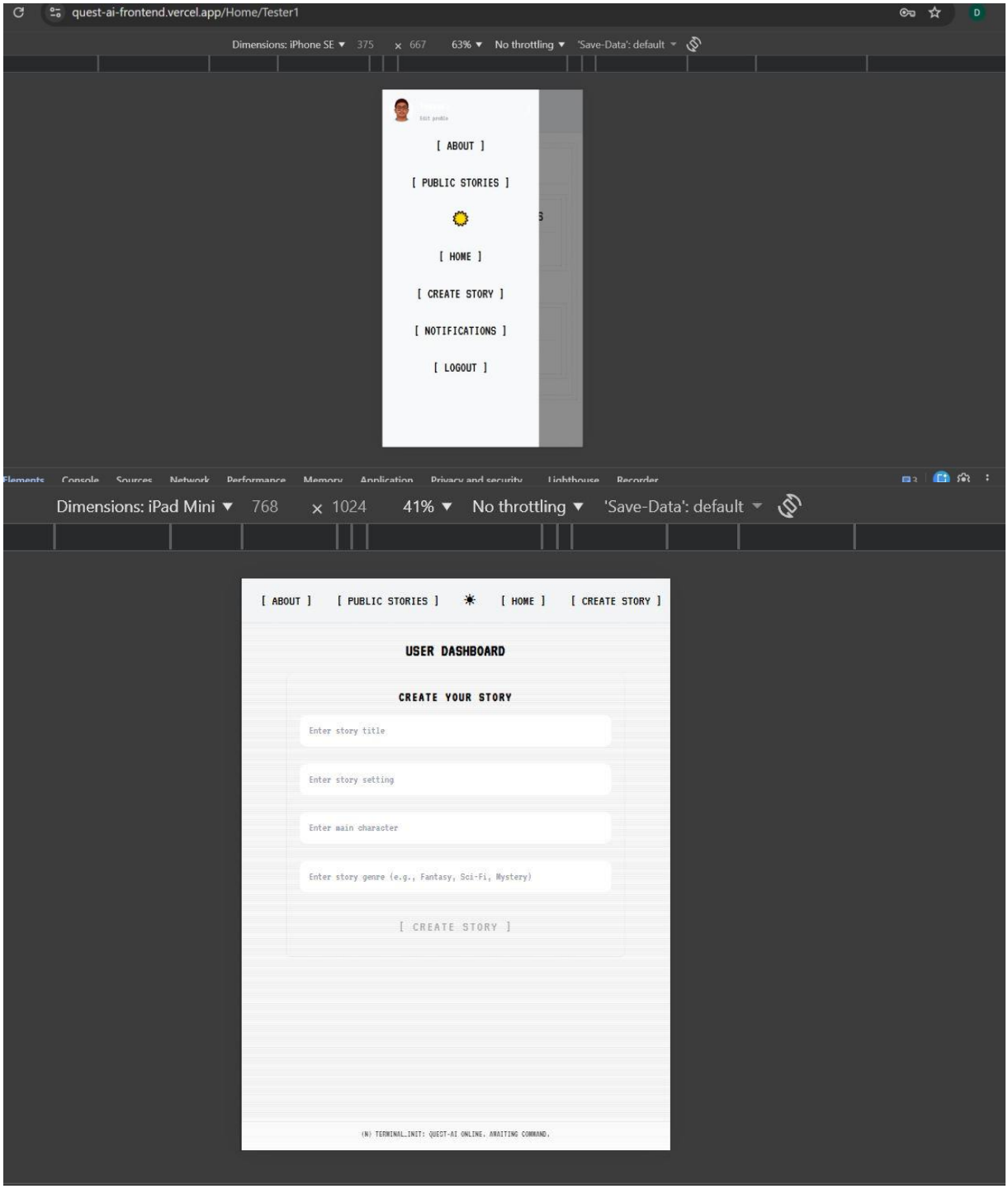
Some buttons on the Home page, especially the ones that open the ChatBox, had to be tested manually because Selenium wasn't able to detect or interact with the ChatBox properly. The rest of the Home page buttons and flows that worked with Selenium were tested using the normal recording method. Both manual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

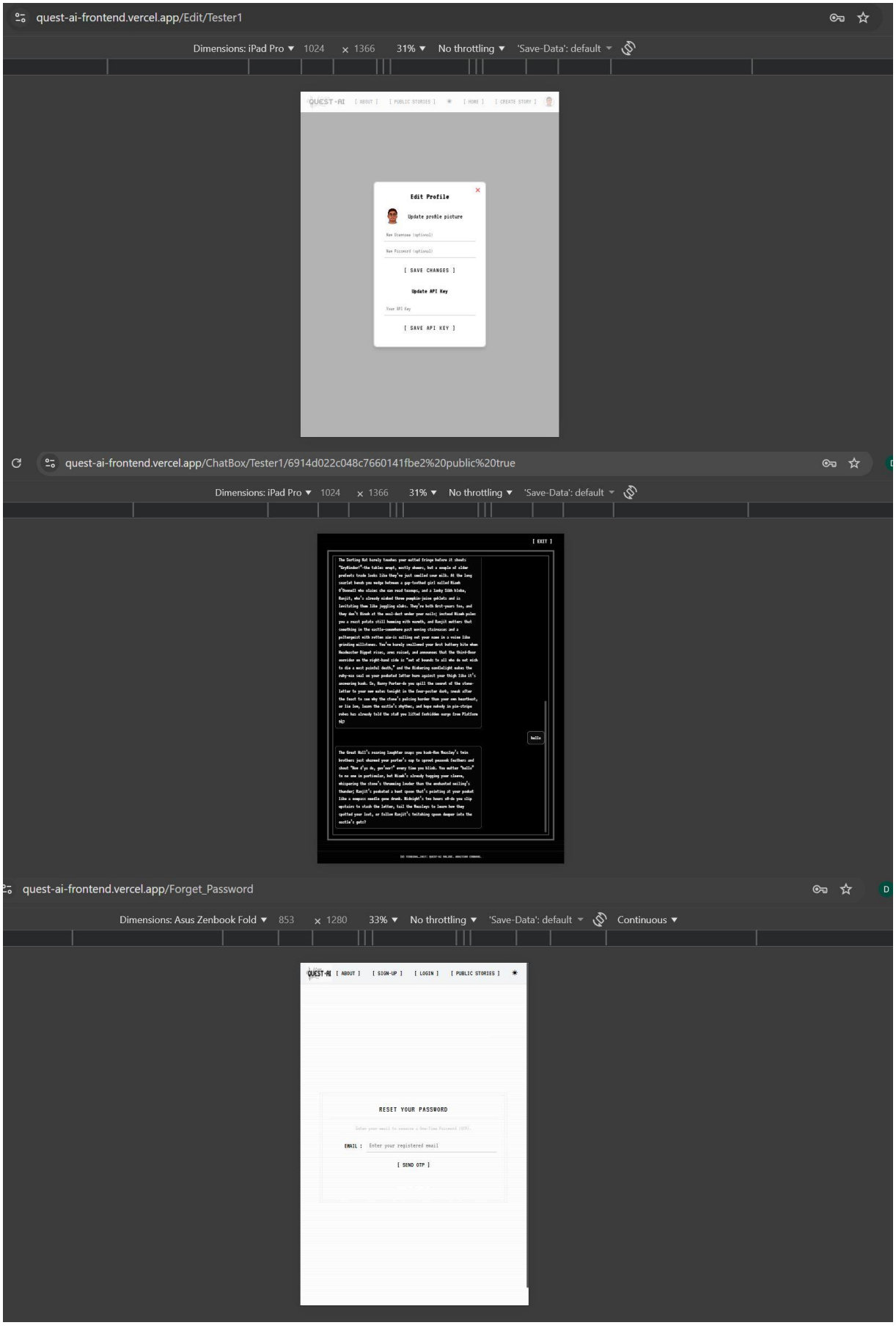
Responsive testing

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:




quest-ai-frontend.vercel.app/About
Dimensions: Samsung Galaxy A51/71 412 x 914 46% No throttling 'Save-Data': default
QUEST-AI
ABOUT QUEST-AI
Welcome to the collaborative dreaming engine. Here, you don't just read the story -you hack the narrative. Set your parameters, initialize your character, and watch a new reality compile itself in real-time.
Why We Built This
> Death to the Blank Page. Traditional Tabletop RPGs and fanfiction often suffer from "setup fatigue." We built Quest-AI to be a frictionless storytelling forge-drop in a genre, pick a persona, and let the engine handle the heavy lifting of world-building instantly.
The AI Twist
> The Dungeon Master is Alive. We don't use rigid scripts. By leaning into the chaotic creativity of LLMs, we ensure no two sessions are identical. The AI acts as an improvisational partner that reacts to your
quest-ai-frontend.vercel.app/Public_Story
Dimensions: Nest Hub Max 1280 x 800 50% No throttling 'Save-Data': default
QUEST-AI
[ABOUT] [PUBLIC STORIES] [HOME] [CREATE STORY]
Public Stories
harry porter and the Philosopher's stone
Owner: dhruvgohil0006@gmail.com
Character: Harry Porter
Description: harry porter, with his goes to hogwards a place where all the Wizards live and where the wizardary is taught.
quest-ai-frontend.vercel.app/Home/Tester1
Dimensions: iPhone SE 375 x 667 63% No throttling 'Save-Data': default
QUEST-AI
USER DASHBOARD
COMPLETED STORIES
> No completed stories.
ONGOING STORIES
> No ongoing stories.


quest-ai-frontend.vercel.app/Home/Tester1
Dimensions: iPhone SE 375 x 667 63% No throttling 'Save-Data': default
[ABOUT]
[PUBLIC STORIES]
[HOME]
[CREATE STORY]
[NOTIFICATIONS]
[LOGOUT]
Dimensions: iPad Mini 768 x 1024 41% No throttling 'Save-Data': default
[ABOUT] [PUBLIC STORIES] [HOME] [CREATE STORY]
USER DASHBOARD
CREATE YOUR STORY
Enter story title
Enter story setting
Enter main character
Enter story genre (e.g., Fantasy, Sci-Fi, Mystery)
[CREATE STORY]
(N) TERMINAL_INIT: QUEST-AI ONLINE. AWAITING COMMAND.


quest-ai-frontend.vercel.app/Edit/Tester1
Dimensions: iPad Pro 1024 x 1366 31% No throttling 'Save-Data': default
QUEST-AI [ABOUT] [PUBLIC STORIES] [HOME] [CREATE STORY]
Edit Profile
Update profile picture
New Username (optional)
New Password (optional)
[SAVE CHANGES]
Update API Key
Your API Key
[SAVE API KEY]
quest-ai-frontend.vercel.app/ChatBox/Tester1/6914d022c048c7660141fbe2%20public%20true
Dimensions: iPad Pro 1024 x 1366 31% No throttling 'Save-Data': default
[EXIT]
Hello
quest-ai-frontend.vercel.app/Forget_Password
Dimensions: Asus Zenbook Fold 853 x 1280 33% No throttling 'Save-Data': default Continuous
QUEST-AI [ABOUT] [SIGN-UP] [LOGIN] [PUBLIC STORIES]
RESET YOUR PASSWORD
EMAIL : Enter your registered email
[SEND OTP]