

IT-314 Software Engineering

GUI Testing Using selenium



**Dhirubhai Ambani
University
Technology**

Formerly DA-IICT

1.Home page testing :

Functionality	Description of Tests Performed	Comments
Primary Navigation	Verified top navigation links: Features , How It Works , FAQs , Home logo	All primary nav links route correctly
Call-to-Action (CTA) Buttons	Tested “Get Started”, “Login”, “Back to Home”, “Sign Up Now”	All CTA buttons navigate properly;
Routing Stability	Multiple forward/back routes between: Home ↔ Auth, Home ↔ Signup, Home ↔ Login	No broken routes or crashes; navigation state consistent
Features Section	Opened each feature card (1–4), tested expand/collapse, validated “See Less”	Expand/Collapse logic works without overlap

Premium Features Cards	Opened “Premium Feature 1” and “Premium Feature 2” cards	UI elements respond correctly
FAQ Accordion	Expanded/collapsed questions 1–5, tested single click & double click	Accordion behaves correctly, no glitches
Scrolling Behavior	Programmatically scrolled to feature sections	Scroll loading and animation work properly
Footer Navigation Links	Verified footer links (Features, How It Works, Privacy Policy, etc.)	Footer routing stable & correctly linked
Mobile Menu / Hamburger Menu	Tested menu toggle, navigation items inside menu, overlay interactions	Mobile menu opens/closes properly and items route correctly
Form Entry Points	Verified all entry points to signup/login from navbar, CTA, footer, and mobile menu	All routes consistent and functional

Selenium test cases & output :

The screenshot shows the Selenium IDE interface with the following details:

- Project:** InsightStox*
- Tests:** home_page_testing*
- Search tests...:** http://localhost:5173
- Test Cases:**
 - 40: Command: link, Target: /#features, Value: linkText=How It Works?
 - 49: Command: click, Target: /#privacy, Value: linkText=Privacy Policy
 - 50: Command: click, Target: /#tos, Value: linkText=Terms Of Service
 - 51: Command: click, Target: /#features, Value: linkText=How It Works?
- Log:**

```

36. click on css=que5 img OK
37. click on css=que5 img OK
38. click on css=logotop img OK
39. click on css=github_logo OK
40. click on css=footer_first_column OK
41. click on css=footer_first_column OK
42. click on css=footer_second_column OK
43. click on css=footer_second_column OK
44. click on linkText=How It Works? OK
45. click on css=question_div OK
46. doubleClick on css=question_div OK
47. click on css=footer_second_column > nav_link:nth-child(2) > a OK
48. click on css=footer_second_column > nav_link:nth-child(3) > a OK
49. click on linkText=How It Works? OK
50. click on linkText=Privacy Policy OK
51. click on linkText=Terms Of Service OK
'home_page_testing' completed successfully

```

2.Authentication page testing :

Functionality	Generalized Actions Performed	Comments
Page Load & Basic UI	Opened /auth, validated screen size, checked visibility of Login / Signup UI elements	Page and UI rendered as expected
Login Form Inputs	Interacted with Email field, Password field, typed multiple values, toggled password visibility	Input fields responsive; visibility toggle works
Login Button Interactions	Click, mouseOver, mouseOut on .submit button	Button state transitions work correctly
Password Visibility Toggle	Clicked .pi-eye, .pi-eye-slash, toggled states repeatedly	Visibility toggle stable under rapid actions
Forgot Password Flow	Clicked “Forgot?”, entered email, submitted, entered OTP, resubmit tested	End-to-end forgot password logic working
OTP Field Testing	Entered values, double-click, mouse interactions, re-typing OTP	OTP input stable to various interactions
Multiple Window Handling	New window opened (e.g., privacy policy), window switching and closing tested	External links open correctly; window switching stable
Signup Page Navigation	Clicked “Sign Up” links from login page	Navigation to signup successful
Signup Form Validations	Typed name, email, password, confirm password, tested mismatched passwords, corrected them	Validation flow works; error handling verified
Password Validation in Signup	Entered invalid and valid passwords, toggled visibility multiple times	Signup password rules properly enforced

Privacy Policy Flow	Opened privacy policy, accepted policy	Redirects and user acceptance logic work correctly
Signup Submission	Completed form, submitted, OTP process tested (123, then correct OTP)	Signup + OTP verification flow validated
Reattempting Login & Signup	Repeated login attempts with various credentials	Auth routes stable under repeated submissions
Return to Login / Home Navigation	Navigated between Login ↔ Forgot ↔ Signup ↔ Privacy Policy ↔ Home	All internal routes functional
Mobile Menu Interactions	Toggled menu → clicked first menu item	Mobile navigation functioning
Extra UI Interactions	Clicked areas like .email-error, .auth_form to test error states	Error popups and focus behavior consistent
Overall System Stability	Full flow executed: Login → Forgot → OTP → Signup → Privacy Policy → OTP → Login	Auth module passed all GUI tests

The screenshot shows the Selenium IDE extension in Mozilla Firefox. The project is named "InsightStox". The recorded script "auth_test" contains the following steps:

- Step 57: click on css=privacy-accept-bltn
- Step 58: click on css=submit
- Step 59: mouse over on css=submit
- Step 60: mouse out on css=submit
- Step 61: click on linkText>Login

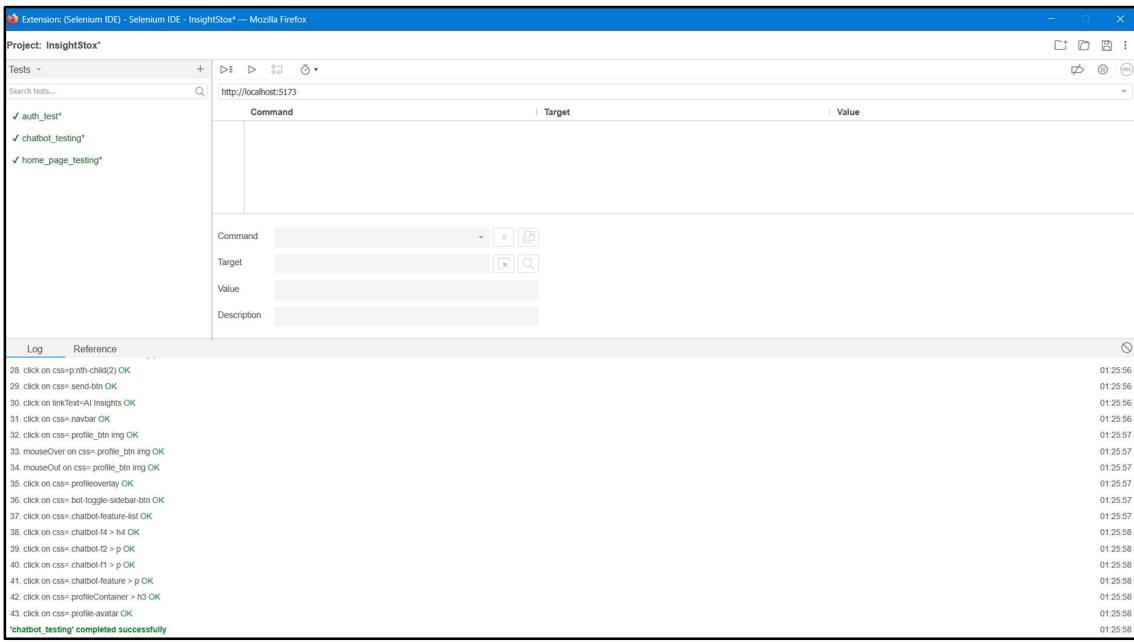
The log pane at the bottom shows the execution details:

- 46. type on id=name with value Gautam OK 01:17:22
- 47. type on id=email with value mgp932854@gmail.com OK 01:17:22
- 48. type on id=password with value Gautam2601&OK 01:17:22
- 49. type on id=confirmPassword with value Gautam2601&OK 01:17:22
- 50. click on css=privacy-policy OK 01:17:23
- 51. click on css=privacy-policy OK 01:17:23
- 52. click on css=privacy-policy OK 01:17:23
- 53. doubleClick on css=privacy-policy OK 01:17:23
- 54. click on css=auth_form OK 01:17:23
- 55. click on css=privacy-policy > input OK 01:17:23
- 56. click on linkText=privacy policy OK 01:17:23
- 57. click on css=privacy-accept-bltn OK 01:17:24
- 58. click on css=submit OK 01:17:24
- 59. mouseOver on css=submit OK 01:17:24
- 60. mouseOut on css=submit OK 01:17:28
- 61. click on linkText>Login OK 01:17:28

The message "'auth_test' completed successfully" is displayed.

3.chatbot page testing :

Functionality	Generalized Actions Performed	Comments
Navigation to Chatbot	Clicked “AI Insights” from menu	Chatbot page routing working correctly
Sidebar Toggle	Opened/closed chatbot sidebar multiple times using .bot-toggle-sidebar-btn	Sidebar state stable under repeated toggles
Chatbot Feature Cards	Clicked feature cards: chatbot-feature f1 ,f2, f3	Feature sections expand/respond correctly
Profile Menu Interactions	Opened profile menu, interacted with avatar, title, settings button → hover and unhover tested	Profile section animations & interactions OK
Overlay Handling	Clicked on ` .profileoverlay multiple times to close menus	Overlay behavior consistent
Send Button Behavior	Single click, double click, repeated clicks on .send-btn	Button works even with rapid interactions
Chat Input Testing	Clicked input, typed text (“hy”), pressed Enter, clicked send button	Message input & submission working
Chat Message Area	Clicked chat-messages & welcome message area	UI responsiveness validated
Menu Toggle (Global Header)	Opened & closed .menu_toggle repeatedly	Mobile menu system stable within chatbot page
General UI Interactions	Clicks on layout elements like chat input area, message area	No UI freeze or misbehavior



4.Preference page testing :

Functionality	Generalized Actions Performed	Comments
Page Load & Navigation	Opened My Profile page and navigated to Preferences via Sidebar	Page loads correctly, sidebar navigation works without errors
Theme Dropdown Interaction	Clicked on Theme Select dropdown and tested option visibility	Dropdown opens and responds properly
Dark Theme Selection	Selected Dark theme from dropdown	Theme updated correctly and alert displayed
Theme Update Alert	Verified alert message " Theme updated successfully! " after Dark theme selection	Alert working correctly
Light Theme Selection	Re-opened dropdown and selected Light theme	Theme switched successfully
Simple Layout Selection	Selected Simple (Essential Metrics) layout	Layout switched back correctly
Layout Re-Update Alert	Verified alert after switching to Simple layout	Alert displays correct confirmation
UI Responsiveness	Checked page stability during multiple dropdown interactions	No crashes, UI stays consistent
Overall Preference Flow	Completed full preference change cycle (Theme + Layout)	Full flow successful without errors

Selenium test cases & output:

Project: GUI_TESTING

Tests + http://localhost:5173/my-profile

HelpSupport_Page ✓ Preference_Page

Command	Target	Value
11 assert alert	Layout updated successfully!	
12 click	id=layout-select	
13 select	id=layout-select	label=Simple (Essential Metrics)
14 assert alert	Layout updated successfully!	

Command: open Target: http://localhost:5173/my-profile Value: Description:

Log	Reference	Date
2. click on css-Sidebar-item:nth-child(4) OK		14:00:26
3. click on id=theme-select OK		14:00:26
4. select on id=theme-select with value label=Dark OK		14:00:27
5. assertAlert on Theme updated successfully! OK		14:00:27
6. click on id=theme-select OK		14:00:44
7. select on id=theme-select with value label=Light OK		14:00:45
8. assertAlert on Theme updated successfully! OK		14:00:45
9. click on id=layout-select OK		14:00:45
10. select on id=layout-select with value label=Detailed (Advanced Insights) OK		14:00:46
11. assertAlert on Layout updated successfully! OK		14:00:46
12. click on id=layout-select OK		14:00:47
13. select on id=layout-select with value label=Simple (Essential Metrics) OK		14:00:47
14. assertAlert on Layout updated successfully! OK		14:00:47
'Preference_Page' completed successfully		14:00:48

5.HelpSupport page testing :

Functionality	Generalized Actions Performed	Comments
Page Load & Navigation	Opened My Profile page and navigated to HelpSupport via Sidebar	Page loads correctly, sidebar navigation works without errors
Quick Help – Getting Started Modal	Opened Getting Started Guide quick help modal and clicked Explore my Dashboard button	Modal opens successfully and button correctly routes to the Dashboard Page
Quick Help – Portfolio Guide Modal	Opened How to Build Portfolio modal and clicked Go to Portfolio button	Modal opens successfully and button correctly routes to the Portfolio Page
Quick Help – AI Suggestions Modal	Opened Understanding AI Suggestions modal and clicked on view my AI Insights button	Modal opens successfully and button correctly routes to the Insights page
Quick Help – Troubleshooting Modal	Opened Troubleshooting modal and closed using overlay	Modal overlay close works properly
Quick Help Buttons Interaction	Clicked all Quick Help buttons sequentially (1–4)	All buttons responded correctly without any UI glitches, and the modal closed successfully using the Escape key or by clicking outside the modal.
Contact Form Field	Located and clicked Message input field	Field is accessible and editable
Contact Form Submission	Clicked Submit Ticket button	Form submits successfully
Contact Form Alert	Verified alert “ Support ticket submitted successfully! ”	Success alert is displayed correctly
Feedback Form Field	Clicked Feedback input field	Input box responds correctly

Feedback Form Submission	Clicked Send Feedback button	Feedback successfully submitted
Feedback Success Alert	Verified alert “ Thank you for your feedback! ”	Success alert appears correctly

Selenium test cases & output :

Project: GUI_TESTING

Tests + | http://localhost:5173/my-profile

✓ HelpSupport_Page ✓ Preference_Page

Command	Target	Value
19 ✓ click	id=feedback-input	
20 ✓ type	id=feedback-input	i am fine.
21 ✓ click	css= help-section:nth-child(4) .btn-primary	
22 ✓ assert alert		Thank you for your feedback!

Command Target Value

Description

Log Reference

```

10. click on css= modal-overlay OK
11. click on css= quick-help-btn nth-child(1) OK
12. click on css= quick-help-btn nth-child(2) OK
13. click on css= quick-help-btn nth-child(3) OK
14. click on css= quick-help-btn nth-child(4) > quick-help-icon OK
15. Trying to find id=message-input OK
16. type on id=message-input with value what are you doing? OK
17. click on css= submit-ticket-btn OK
18. assertAlert on Support ticket submitted successfully! OK
19. click on id=feedback-input OK
20. type on id=feedback-input with value i am fine. OK
21. click on css= help-section:nth-child(4) .btn-primary OK
22. assertAlert on Thank you for your feedback! OK
'HelpSupport_Page' completed successfully

```

14:36:04
14:36:04
14:36:05
14:36:06
14:36:07
14:36:08
14:36:11
14:36:12
14:36:13
14:36:15
14:36:16
14:36:17
14:36:18
14:36:41

6. WatchList page with dashboard-header testing

Functionality	Description of Tests Performed	Comments
Add Stock via Global Header	Clicked global search bar (<code>.search-input</code>), typed "reliance", and clicked the add button (<code>.pi</code>) on the 2nd result item.	Global search integration works; stock added to watchlist successfully.
Local Search Filtering	Focused on Watchlist's local search input, typed "reliance" to filter the table rows.	Table filtered correctly to show specific stocks.
Remove Stock Action	Clicked the "Remove" button (<code>.action-btn</code>) on the specific stock row.	Stock successfully removed from the list; UI updated.
Clear Local Search	Double-clicked the local search input and cleared the text to reset the view.	Search input cleared; full watchlist restored.
Filter Modal (Daily Change)	Opened Filter Modal (<code>.filter-btn</code>), selected "Gainers" (<code>#gainers</code>), and clicked "Apply Filters".	Filter applied successfully; list updated to show gainers.
Clear Filters & Overlay	Re-opened modal, clicked "Clear All", then closed the modal by clicking the background overlay (<code>.filter-modal-overlay</code>).	Filters reset correctly; overlay click successfully closed the modal.
Filter Modal (Market Cap)	Opened modal again, selected "Large Cap" (<code>#large-cap</code>), and applied filters.	"Large Cap" filter logic applied correctly.
Stock Details Navigation (Table)	Clicked on a specific stock row (e.g., TITAN.NS) within the table.	Successfully navigated to the Stock Details page.
Stock Details Navigation (Header)	Navigated back to Watchlist, searched for "titan" in global search, and clicked the result.	Global search successfully navigated to Stock Details page.

Selenium test cases & output:

The screenshot shows the Selenium IDE interface with the following details:

- Project:** watchlistpage
- Executing:** watchlist
- URL:** http://localhost:5173
- Test Case Table:**

Command	Target	Value
1 ✓ open	/watchlist	
2 ✓ set window size	1552x832	
3 ✓ click	css=watchlist	
4 ✓ click	css=search-input	
5 ✓ type	css=popup-search-input	reliance
- Log:**
 - 20. click on css=filter-btn > img OK 05:38:35
 - 21. click on id:large-cap OK 05:38:36
 - 22. click on css=apply-filter-btn OK 05:38:37
 - 23. click on css=filter-btn OK 05:38:38
 - 24. click on css=clear-filter-btn OK 05:38:39
 - 25. click on css=apply-filter-btn OK 05:38:40
 - 26. click on css=watchlist OK 05:38:41
 - 27. click on css=table-stock:nth-child(1) > td:nth-child(2) OK 05:38:42
 - 28. click on linkText=Watchlist OK 05:38:43
 - 29. click on css=search-input OK 05:38:44
 - 30. type on css=popup-search-input with value titan OK 05:38:45
 - 31. click on css=result-item:nth-child(1) OK 05:38:46
 - 'watchlist' completed successfully** 05:38:47

7. MyProfile page with testing

The screenshot shows the Selenium IDE extension in Mozilla Firefox. The title bar reads "Extension: (Selenium IDE) - Selenium IDE - InsightStox - Group21* — Mozilla Firefox". The main interface displays a test case titled "✓ My Profile Page*". The test log table shows the following steps:

	Command	Target	Value
51	✓ click	css=.myPage_InfRow1 .myPage_EditDetails	
52	✓ click	css=input:nth-child(1)	
53	✓ type	css=input:nth-child(1)	Ayush D
54	✓ click	css=.myPage_SaveBtn	

Below the table are input fields for "Command", "Target", "Value", and "Description". At the bottom left, it says "Runs: 1 Failures: 0". The log section at the bottom shows the following entries:

```

Log Reference
50. click on css=.myPage_CancelBtn OK 01:32:03
51. click on css=.myPage_InfRow1 .myPage_EditDetails OK 01:32:04
52. click on css=input:nth-child(1) OK 01:32:04
53. type on css=input:nth-child(1) with value Ayush D OK 01:32:04
54. click on css=.myPage_SaveBtn OK 01:32:04
'My Profile Page' completed successfully 01:32:04

```

- There are a total of around 54 events that have been done as testing for the MyProfile page and all those events are triggering properly and working properly.

8. Portfolio Page with testing

The screenshot shows the Selenium IDE interface with the following details:

- Project:** InsightStox - Group21*
- Test Case:** ✓ Portfolio Page*
- URL:** https://insight-stox-portfolio-analyzer-tra-gamma.vercel.app/portfolio
- Test Log:**
 - 11. mouseOver on linkText=Portfolio OK
 - 12. click on linkText=Portfolio OK
 - 13. click on css=.profile_btn > button OK
 - 14. mouseOver on css=button > img OK
 - 15. mouseOut on css=button > img OK
 - 16. click on css=.profileoverlay OK
 - 'Portfolio Page' completed successfully

- There are a total of around 16 events that have been done as testing for the Portfolio page and all those events are triggering properly and working properly.

9. StockDetails page with test

The screenshot shows the Selenium IDE interface with the following details:

- Project:** InsightStox - Group21*
- Executing:** ✓ Stock Details
- Run all tests:** Ctrl+Shift+R | 73/
- Test Log:**

	Command	Target	Value
26	✓ click	css=input:nth-child(1)	
27	✓ type	css=input:nth-child(1)	12
28	✓ click	css=.model-confirm-btn	
29	✓ click	css=button > img	
30	✓ click	css=.profileoverlay	
- Log:**
 - 25. click on css=.model-sell OK 01:57:57
 - 26. click on css=input:nth-child(1) OK 01:57:57
 - 27. type on css=input:nth-child(1) with value 12 OK 01:57:57
 - 28. click on css=.model-confirm-btn OK 01:57:57
 - 29. click on css=button > img OK 01:57:57
 - 30. click on css=.profileoverlay OK 01:57:58
 - 'Stock Details page' completed successfully 01:57:58

- There are a total of around 30 events that have been done as testing for the StockDetail page and all those events are triggering properly and working properly.