IT314
Software Engineering

Group 5
Hostel Management System

GUI Testing

Instructor: Dr. Saurabh Tiwari and Dr. Manish Khare

Date: 26/04/2023

The tests are done using the Selenium plugin.

1.Test Case: Verify Login Functionality on the website

Test Steps:

1. Launch the web application URL
2. Verify that the login page is displayed
3. Enter valid credentials in the login fields
4. Click on the "Login" button
5. Verify that the home page is displayed
6. Logout from the application
7. Verify that the login page is displayed again

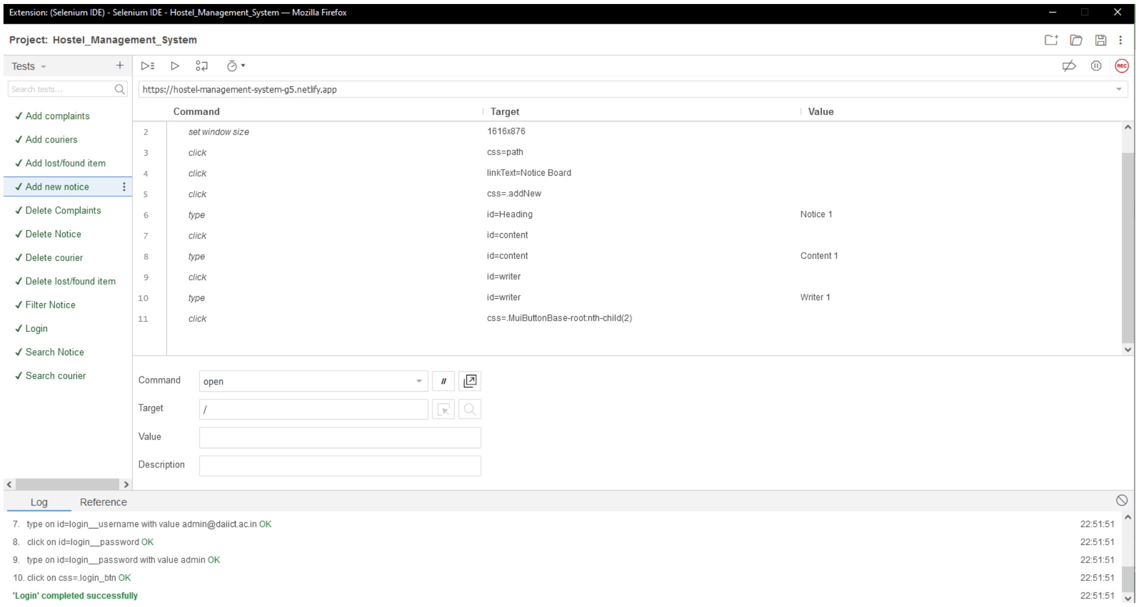
Test Data:

- Valid login credentials (username and password)

Expected Results:

- After step 1, the web application URL should be launched successfully
- After step 2, the login page should be displayed correctly
- After step 3, valid credentials should be entered successfully in the login fields
- After step 4, the home page should be displayed correctly with the user logged in
- After step 6, the user should be logged out successfully from the application
- After step 7, the login page should be displayed again correctly

Results:



Extension: (Selenium IDE) - Selenium IDE - Hostel_Management_System — Mozilla Firefox

Project: Hostel_Management_System

- ✓ Add complaints
- ✓ Add couriers
- ✓ Add lost/found item
- ✓ Add new notice
- ✓ Delete Complaints
- ✓ Delete Notice
- ✓ Delete courier
- ✓ Delete lost/found item
- ✓ Filter Notice
- ✓ Login
- ✓ Search Notice
- ✓ Search courier

https://hostel-management-system-g5.netlify.app

	Command	Target	Value
2	*set window size*	1616x876	
3	*click*	css=path	
4	*click*	linkText=Notice Board	
5	*click*	css=.addNew	
6	*type*	id=Heading	Notice 1
7	*click*	id=content	
8	*type*	id=content	Content 1
9	*click*	id=writer	
10	*type*	id=writer	Writer 1
11	*click*	css=.MuiButtonBase-root:nth-child(2)	

Command: open
Target: /
Value:
Description:

```
7.  type on id=login__username with value admin@daiict.ac.in OK          22:51:51
8.  click on id=login__password OK                                       22:51:51
9.  type on id=login__password with value admin OK                       22:51:51
10. click on css=.login_btn OK                                           22:51:51
'Login' completed successfully                                           22:51:51
```

2.Test Case: Verify Add Notice Functionality on the website

Preconditions:

- The user has a valid account and is logged in to the hostel management website.

Test Steps:

1. Launch the web application URL
2. Navigate to the notices page using the sidebar.
3. Verify the notice page is displayed.
4. Click on the "Add Notice" button.
5. Verify that the “Add Notice” window pops up.
6. Write the title, message and creator name in the respective fields
7. Verify the title, message and creator name the user wants to add is added to the respective text fields.
8. Click on the “Add” button.
9. Verify that the notice is added.

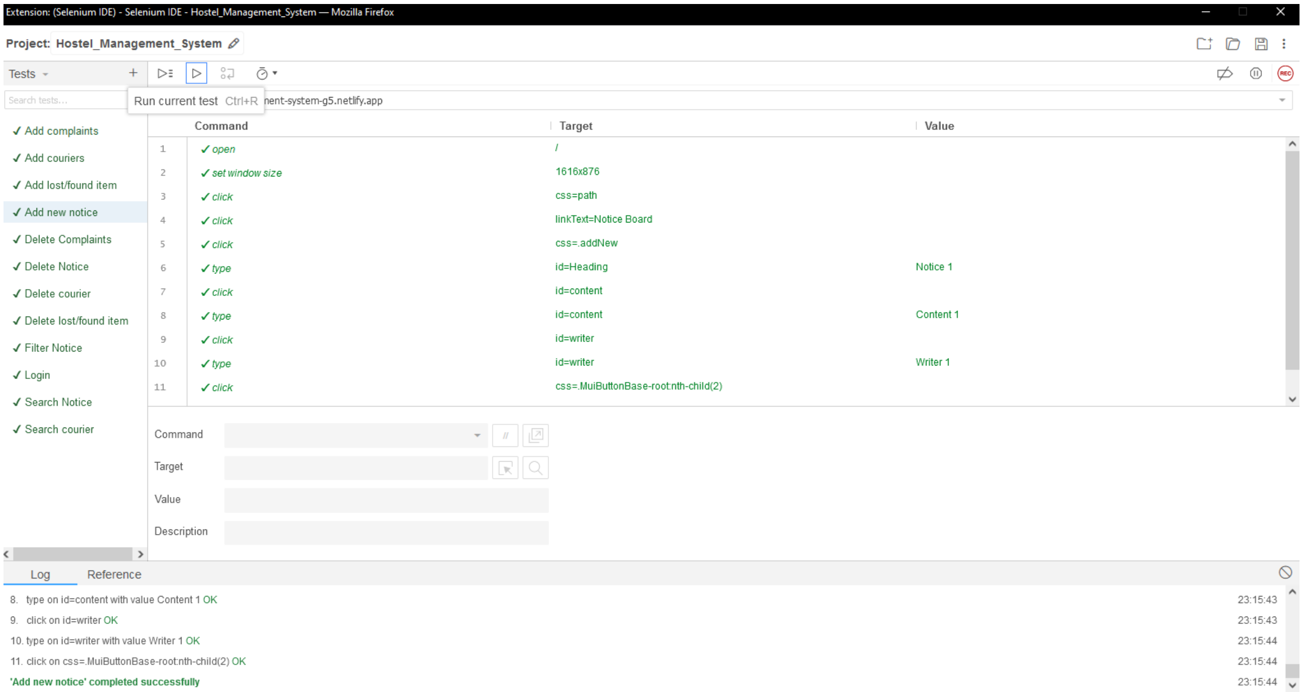
Test Data:

- Notice content- Title, message and creator name

Expected Results:

- After step 1, the web application URL should be launched successfully.
- After step 2, the user should be redirected to the notices page.
- After step 3, the notices page should be displayed correctly.
- After step 4, a window should appear for the purpose to add notice.
- After step 5, the window should be displayed correctly.
- After step 6, the data will be filled in the text fields.
- After step 7, the text fields should display correct data.
- After step 8, the notices page should be displayed.
- After step 9, the notices page should be displayed correctly with new notice added

Results:



Extension: (Selenium IDE) - Selenium IDE - Hostel_Management_System — Mozilla Firefox
Project: Hostel_Management_System
Tests
Search tests...
Run current test Ctrl+R
REC
Log
Reference

nent-system-g5.netlify.app

- ✓ Add complaints
- ✓ Add couriers
- ✓ Add lost/found item
- ✓ Add new notice
- ✓ Delete Complaints
- ✓ Delete Notice
- ✓ Delete courier
- ✓ Delete lost/found item
- ✓ Filter Notice
- ✓ Login
- ✓ Search Notice
- ✓ Search courier

	Command	Target	Value
1	✓ open	/	
2	✓ set window size	1616x876	
3	✓ click	css=path	
4	✓ click	linkText=Notice Board	
5	✓ click	css=.addNew	
6	✓ type	id=Heading	Notice 1
7	✓ click	id=content	
8	✓ type	id=content	Content 1
9	✓ click	id=writer	
10	✓ type	id=writer	Writer 1
11	✓ click	css=.MuiButtonBase-root:nth-child(2)	

Command	
Target	
Value	
Description	

```
8.  type on id=content with value Content 1 OK                 23:15:43
9.  click on id=writer OK                                      23:15:43
10. type on id=writer with value Writer 1 OK                   23:15:44
11. click on css=.MuiButtonBase-root:nth-child(2) OK           23:15:44
'Add new notice' completed successfully                        23:15:44
```

3.Test Case: Verify Delete Notice Functionality on the website

Preconditions:

- The user has a valid account and is logged in to the hostel management website.
- The user must be admin.

Test Steps:

1. Launch the web application URL
2. Navigate to the notices page using the sidebar.
3. Verify the notice page is displayed.
4. Click on the "Delete" button corresponding to the notice the user wants to delete.
5. Verify that the notice is deleted.

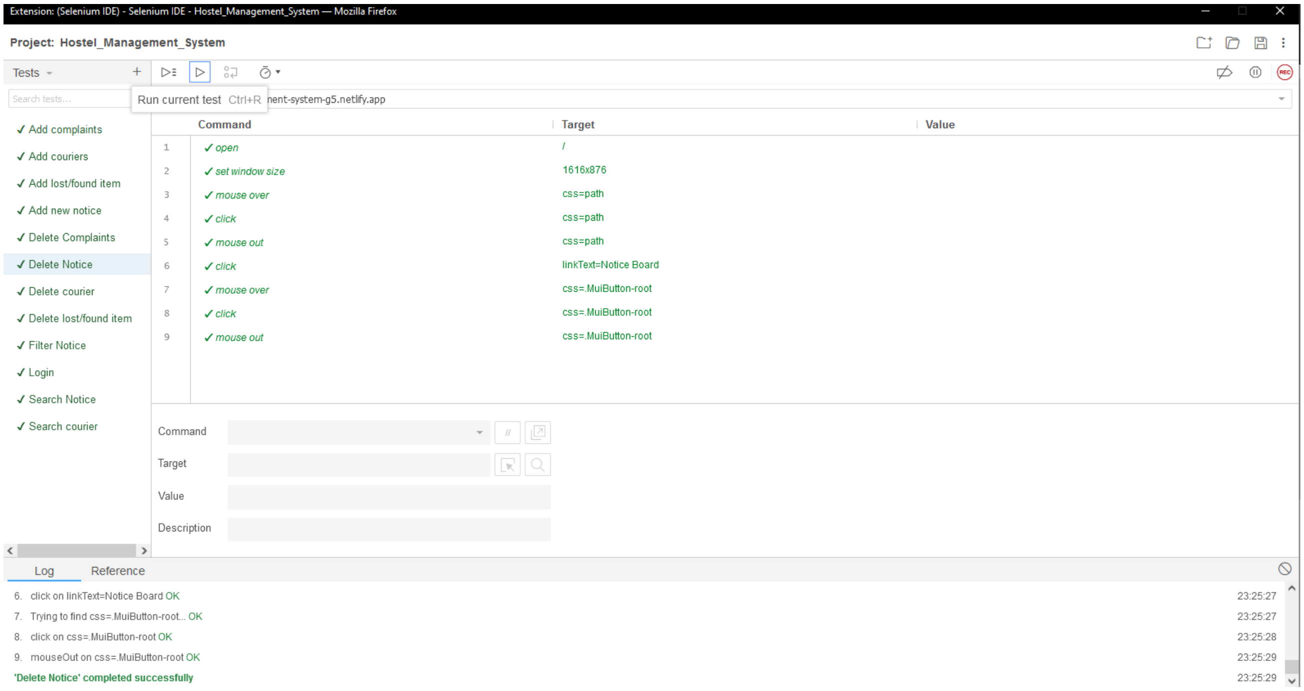
Test Data:

- Delete button

Expected Results:

- After step 1, the web application URL should be launched successfully.
- After step 2, the user should be redirected to the notices page.
- After step 3, the notices page should be displayed correctly.
- After step 4, the notices page should be displayed.
- After step 5, the notices page should be displayed correctly with the notice deleted.

Results:



Extension: (Selenium IDE) - Selenium IDE - Hostel_Management_System — Mozilla Firefox

Project: Hostel_Management_System

nent-system-g5.netlify.app

- ✓ Add complaints
- ✓ Add couriers
- ✓ Add lost/found item
- ✓ Add new notice
- ✓ Delete Complaints
- ✓ Delete Notice
- ✓ Delete courier
- ✓ Delete lost/found item
- ✓ Filter Notice
- ✓ Login
- ✓ Search Notice
- ✓ Search courier

	Command	Target	Value
1	✓ open	/	
2	✓ set window size	1616x876	
3	✓ mouse over	css=path	
4	✓ click	css=path	
5	✓ mouse out	css=path	
6	✓ click	linkText=Notice Board	
7	✓ mouse over	css=.MuiButton-root	
8	✓ click	css=.MuiButton-root	
9	✓ mouse out	css=.MuiButton-root	

```
6.  click on linkText=Notice Board OK                23:25:27
7.  Trying to find css=.MuiButton-root... OK         23:25:27
8.  click on css=.MuiButton-root OK                  23:25:28
9.  mouseOut on css=.MuiButton-root OK               23:25:29
'Delete Notice' completed successfully               23:25:29
```

4.Test Case: Verify Filter Notice Functionality on the website

Preconditions:

- The user has a valid account and is logged in to the hostel management website.

Test Steps:

1. After step 1, the web application URL should be launched successfully.
2. After step 2, the user should be redirected to the notices page.
3. After step 3, the filter should be changed to author.
4. After step 4, the author name should be selected.
5. After step 5, the notices page should be displayed correctly with filtered notices.

Test Data:

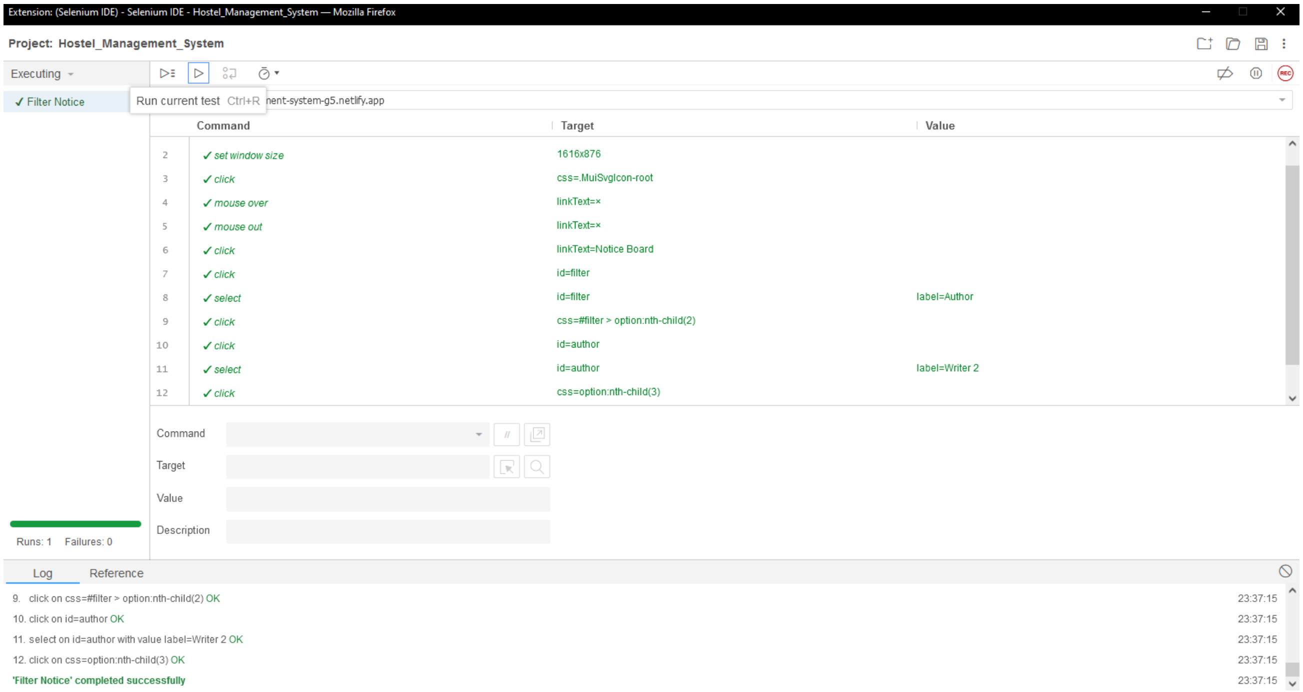
- Filter button

Expected Results:

- After step 1, the web application URL should be launched successfully.
- After step 2, the user should be redirected to the notices page.
- After step 3, the author selection button should appear.

- After step 4, the notices page should be displayed.
- After step 5, the notices page should be displayed correctly with the filtered notices.

Results:


Extension: (Selenium IDE) - Selenium IDE - Hostel_Management_System — Mozilla Firefox
Project: Hostel_Management_System
Executing
✓ Filter Notice
Run current test Ctrl+R
nent-system-g5.netlify.app
Command
Target
Value
Description
Runs: 1 Failures: 0
Log
Reference

	Command	Target	Value
2	✓ set window size	1616x876	
3	✓ click	css=.MuiSvgIcon-root	
4	✓ mouse over	linkText=×	
5	✓ mouse out	linkText=×	
6	✓ click	linkText=Notice Board	
7	✓ click	id=filter	
8	✓ select	id=filter	label=Author
9	✓ click	css=#filter > option:nth-child(2)	
10	✓ click	id=author	
11	✓ select	id=author	label=Writer 2
12	✓ click	css=option:nth-child(3)	

```
9.  click on css=#filter > option:nth-child(2) OK                    23:37:15
10. click on id=author OK                                            23:37:15
11. select on id=author with value label=Writer 2 OK                 23:37:15
12. click on css=option:nth-child(3) OK                              23:37:15
'Filter Notice' completed successfully                               23:37:15
```

5.Test Case: Verify Search Notice Functionality on the website

Preconditions:

- The user has a valid account and is logged in to the hostel management website.

Test Steps:

1. After step 1, the web application URL should be launched successfully.
2. After step 2, the user should be redirected to the notices page.
3. After step 3, write content in the search box
4. After step 4, the notices page should be displayed correctly with searched notices.

Test Data:

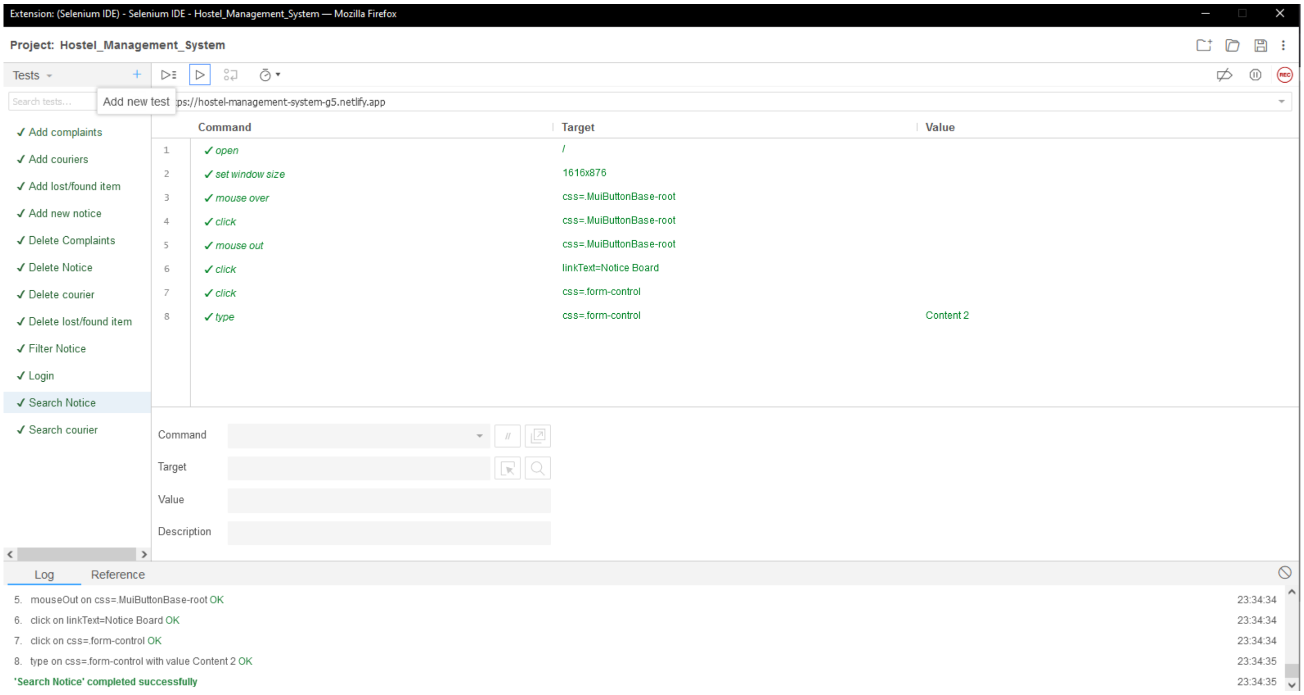
- Search button

Expected Results:

- After step 1, the web application URL should be launched successfully.
- After step 2, the user should be redirected to the notices page.
- After step 3, the notices page should be displayed.

- After step 4, the notices page should be displayed correctly with the searched notices.

Results:



Extension: (Selenium IDE) - Selenium IDE - Hostel_Management_System — Mozilla Firefox

Project: Hostel_Management_System

ps://hostel-management-system-g5.netlify.app

- ✓ Add complaints
- ✓ Add couriers
- ✓ Add lost/found item
- ✓ Add new notice
- ✓ Delete Complaints
- ✓ Delete Notice
- ✓ Delete courier
- ✓ Delete lost/found item
- ✓ Filter Notice
- ✓ Login
- ✓ Search Notice
- ✓ Search courier

	Command	Target	Value
1	✓ open	/	
2	✓ set window size	1616x876	
3	✓ mouse over	css=.MuiButtonBase-root	
4	✓ click	css=.MuiButtonBase-root	
5	✓ mouse out	css=.MuiButtonBase-root	
6	✓ click	linkText=Notice Board	
7	✓ click	css=.form-control	
8	✓ type	css=.form-control	Content 2

```
5.  mouseOut on css=.MuiButtonBase-root OK                                23:34:34
6.  click on linkText=Notice Board OK                                     23:34:34
7.  click on css=.form-control OK                                         23:34:34
8.  type on css=.form-control with value Content 2 OK                     23:34:35
'Search Notice' completed successfully                                    23:34:35
```

6.Test Case: Verify Add Complaint Functionality on the website

Preconditions:

- The user has a valid account and is logged in to the hostel management website.

Test Steps:

1. Launch the web application URL
2. Navigate to the complaints page using the sidebar.
3. Verify the complaints page is displayed.
4. Click on the "Add Complaints" button.
5. Verify that the “Add Complaint” window pops up.
6. Write the title, message and creator name in the respective fields
7. Verify the title, message and creator name the user wants to add is added to the respective text fields.
8. Click on the “Add” button.
9. Verify that the complaint is added.

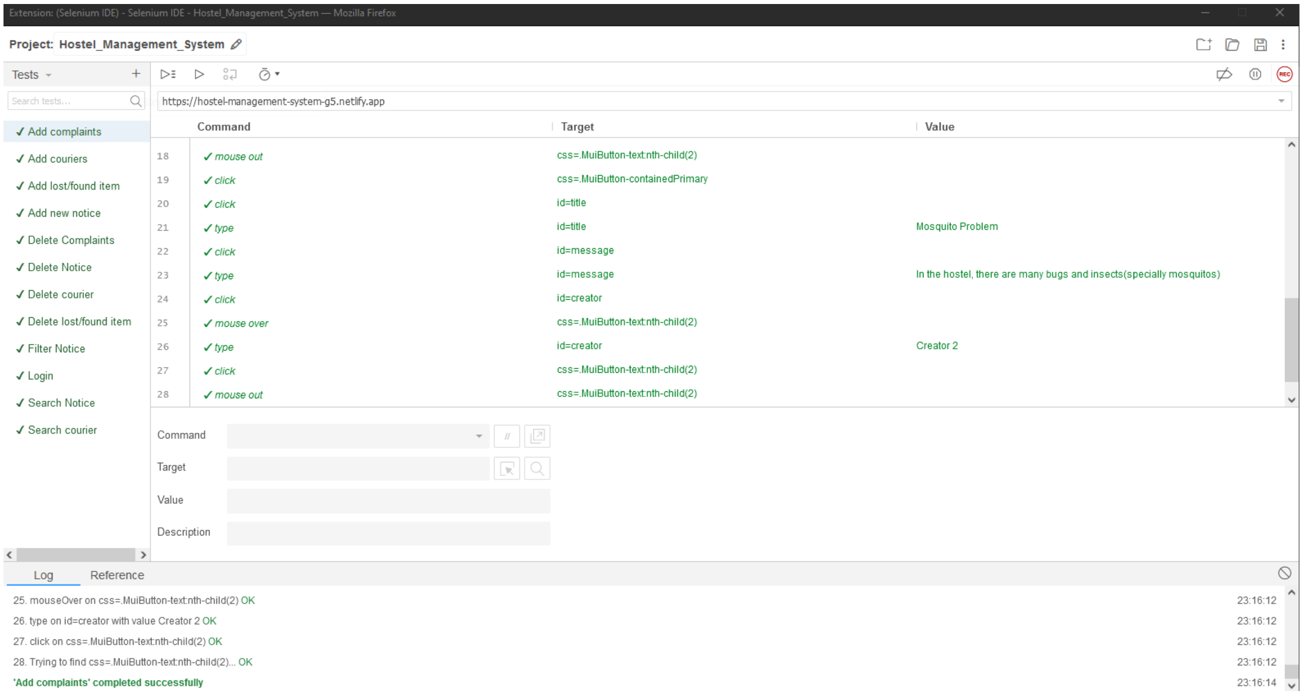
Test Data:

- Complaint content- Title, message and creator name

Expected Results:

- After step 1, the web application URL should be launched successfully.
- After step 2, the user should be redirected to the complaints page.
- After step 3, the complaints page should be displayed correctly.
- After step 4, a window should appear for the purpose to add complaints.
- After step 5, the window should be displayed correctly.
- After step 6, the data will be filled in the text fields.
- After step 7, the text fields should display correct data.
- After step 8, the complaints page should be displayed.
- After step 9, the complaints page should be displayed correctly with new complaint added

Results:



https://hostel-management-system-g5.netlify.app

	Command	Target	Value
18	✓ mouse out	css=.MuiButton-text:nth-child(2)	
19	✓ click	css=.MuiButton-containedPrimary	
20	✓ click	id=title	
21	✓ type	id=title	Mosquito Problem
22	✓ click	id=message	
23	✓ type	id=message	In the hostel, there are many bugs and insects(specially mosquitos)
24	✓ click	id=creator	
25	✓ mouse over	css=.MuiButton-text:nth-child(2)	
26	✓ type	id=creator	Creator 2
27	✓ click	css=.MuiButton-text:nth-child(2)	
28	✓ mouse out	css=.MuiButton-text:nth-child(2)	

Command
Target
Value
Description

```
25. mouseOver on css=.MuiButton-text:nth-child(2) OK                23:16:12
26. type on id=creator with value Creator 2 OK                      23:16:12
27. click on css=.MuiButton-text:nth-child(2) OK                    23:16:12
28. Trying to find css=.MuiButton-text:nth-child(2)... OK           23:16:12
'Add complaints' completed successfully                             23:16:14
```

7.Test Case: Verify Delete Complaints Functionality on the website

Preconditions:

- The user has a valid account and is logged in to the hostel management website.
- The user must be admin.

Test Steps:

1. Launch the web application URL
2. Navigate to the complaints page using the sidebar.
3. Verify the complaints page is displayed.
4. Click on the "Delete" button corresponding to the complaint the user wants to delete.
5. Verify that the complaint is deleted.

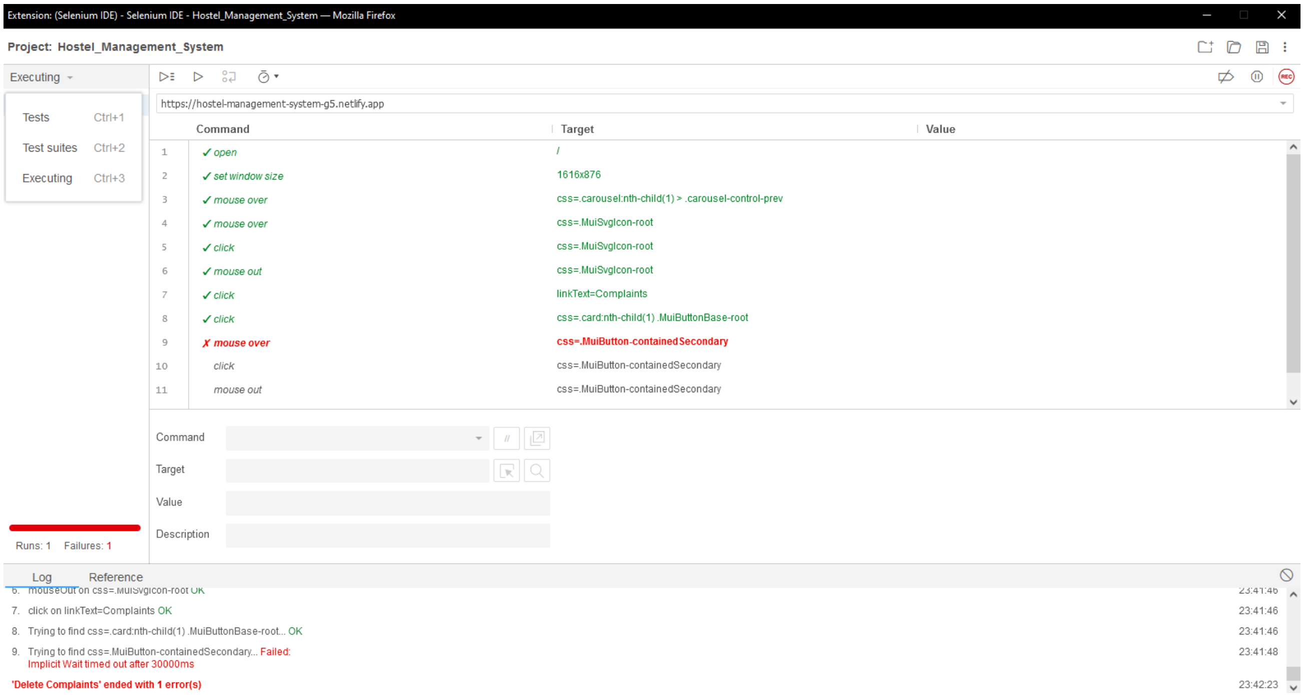
Test Data:

- Delete button

Expected Results:

- After step 1, the web application URL should be launched successfully.
- After step 2, the user should be redirected to the complaints page.
- After step 3, the complaints page should be displayed correctly.
- After step 4, the complaints page should be displayed.
- After step 5, the complaints page should be displayed correctly with the complaints deleted.

Results:



Extension: (Selenium IDE) - Selenium IDE - Hostel_Management_System — Mozilla Firefox

Project: Hostel_Management_System

https://hostel-management-system-g5.netlify.app

	Command	Target	Value
1	✓ open	/	
2	✓ set window size	1616x876	
3	✓ mouse over	css=.carousel:nth-child(1) > .carousel-control-prev	
4	✓ mouse over	css=.MuiSvgIcon-root	
5	✓ click	css=.MuiSvgIcon-root	
6	✓ mouse out	css=.MuiSvgIcon-root	
7	✓ click	linkText=Complaints	
8	✓ click	css=.card:nth-child(1) .MuiButtonBase-root	
9	✗ mouse over	css=.MuiButton-containedSecondary	
10	click	css=.MuiButton-containedSecondary	
11	mouse out	css=.MuiButton-containedSecondary	

Runs: 1 Failures: 1

```
6. mouseOut on css=.MuiSvgIcon-root OK                                    23:41:46
7. click on linkText=Complaints OK                                        23:41:46
8. Trying to find css=.card:nth-child(1) .MuiButtonBase-root... OK        23:41:46
9. Trying to find css=.MuiButton-containedSecondary... Failed:            23:41:48
   Implicit Wait timed out after 30000ms
'Delete Complaints' ended with 1 error(s)                                 23:42:23
```

8.Test Case: Verify Add Couriers Functionality on the website

Preconditions:

- The user has a valid account and is logged in to the hostel management website.

Test Steps:

1. Launch the web application URL
2. Navigate to the couriers page using the sidebar.
3. Verify the couriers page is displayed.
4. Write the student name, room number and courier ID in the respective fields
5. Verify the student name, room number and courier ID the user wants to add is added to the respective text fields.
6. Click on the “Add” button.
7. Verify that the courier is added.

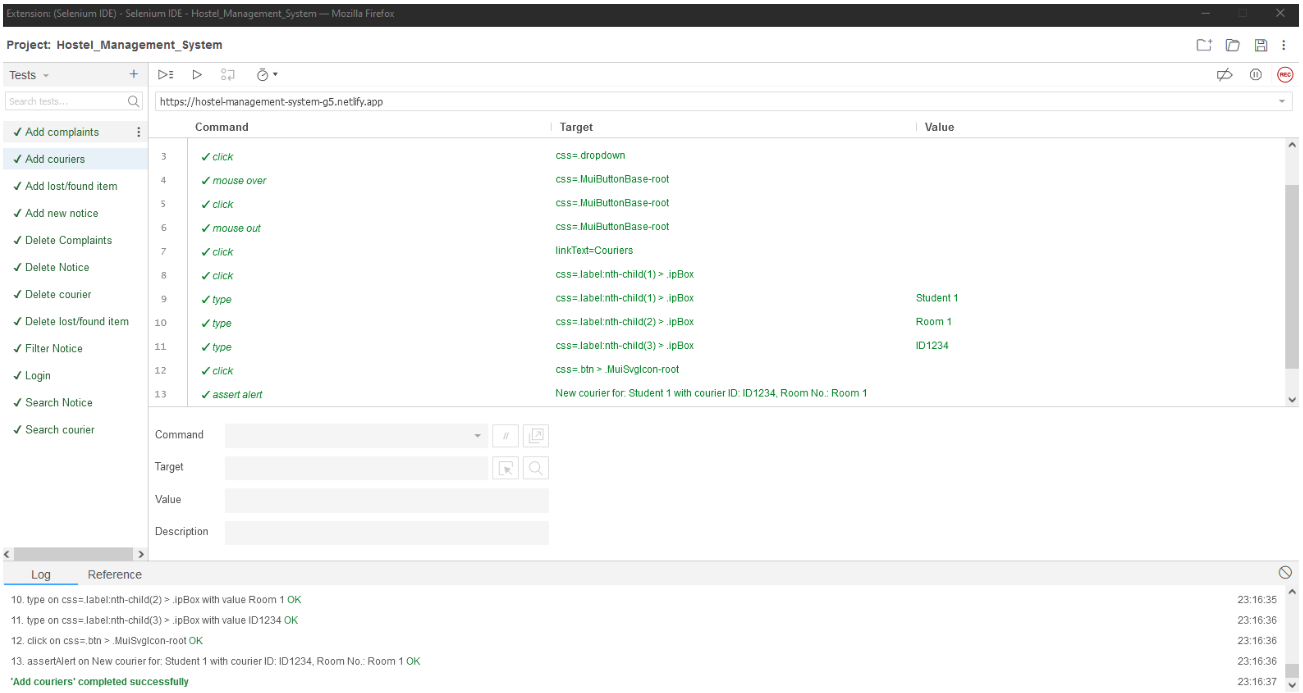
Test Data:

- Courier content- student name, room number and courier ID

Expected Results:

- After step 1, the web application URL should be launched successfully.
- After step 2, the user should be redirected to the complaints page.
- After step 3, the complaints page should be displayed correctly.
- After step 4, the data will be filled in the text fields.
- After step 5, the text fields should display correct data.
- After step 6, the couriers page should be displayed.
- After step 7, the couriers page should be displayed correctly with new couriers added

Results:



Project: Hostel_Management_System

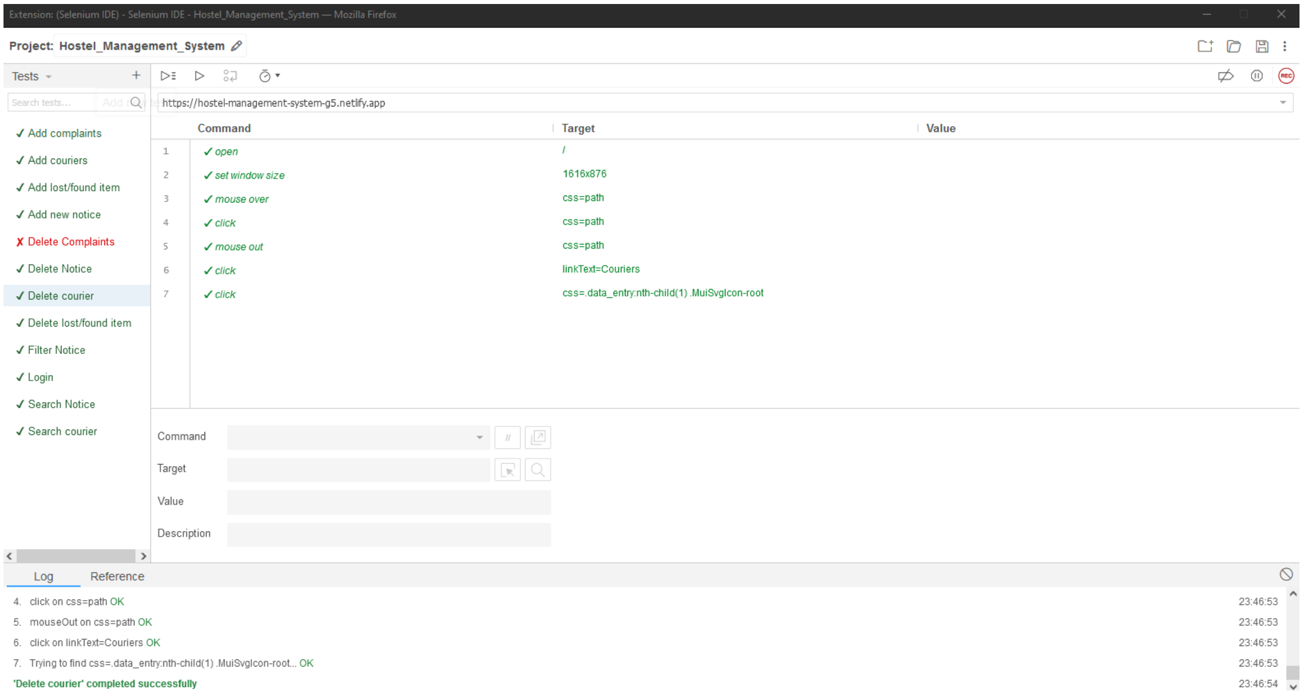
https://hostel-management-system-g5.netlify.app

	Command	Target	Value
3	✓ *click*	css=.dropdown	
4	✓ *mouse over*	css=.MuiButtonBase-root	
5	✓ *click*	css=.MuiButtonBase-root	
6	✓ *mouse out*	css=.MuiButtonBase-root	
7	✓ *click*	linkText=Couriers	
8	✓ *click*	css=.label:nth-child(1) > .ipBox	
9	✓ *type*	css=.label:nth-child(1) > .ipBox	Student 1
10	✓ *type*	css=.label:nth-child(2) > .ipBox	Room 1
11	✓ *type*	css=.label:nth-child(3) > .ipBox	ID1234
12	✓ *click*	css=.btn > .MuiSvgIcon-root	
13	✓ *assert alert*	New courier for: Student 1 with courier ID: ID1234, Room No.: Room 1	

```
10. type on css=.label:nth-child(2) > .ipBox with value Room 1 OK                              23:16:35
11. type on css=.label:nth-child(3) > .ipBox with value ID1234 OK                              23:16:36
12. click on css=.btn > .MuiSvgIcon-root OK                                                    23:16:36
13. assertAlert on New courier for: Student 1 with courier ID: ID1234, Room No.: Room 1 OK     23:16:36
'Add couriers' completed successfully                                                          23:16:37
```

9.Test Case: Verify Delete Couriers Functionality on the website

Results:



Project: Hostel_Management_System

https://hostel-management-system-g5.netlify.app

	Command	Target	Value
1	✓ *open*	/	
2	✓ *set window size*	1616x876	
3	✓ *mouse over*	css=path	
4	✓ *click*	css=path	
5	✓ *mouse out*	css=path	
6	✓ *click*	linkText=Couriers	
7	✓ *click*	css=.data_entry:nth-child(1) .MuiSvgIcon-root	

```
4.  click on css=path OK                                                23:46:53
5.  mouseOut on css=path OK                                             23:46:53
6.  click on linkText=Couriers OK                                       23:46:53
7.  Trying to find css=.data_entry:nth-child(1) .MuiSvgIcon-root... OK  23:46:53
'Delete courier' completed successfully                                 23:46:54
```

10.Test Case: Search Courier

Test Steps:

1. Navigate to the couriers page using the sidebar.
2. Verify the “Name” written in the search bar field.
3. Verify whether the couriers previewed in the couriers log are of the desired “Name”.

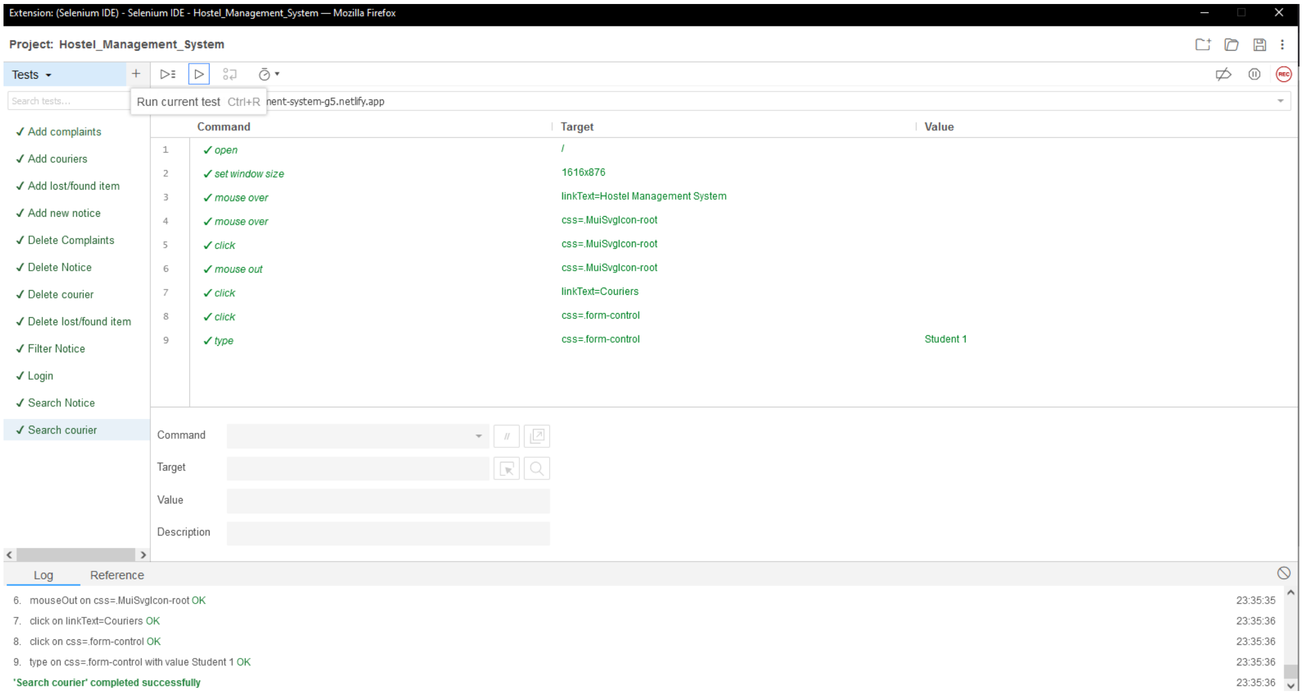
Test Data:

- The user has a valid account and is logged in to the hostel management website.
- The user wants to see if any courier has been received with their name.
- The user must have the role as **student**.

Expected Results:

- The user should be able to search the name for the couriers’ receiver.
- The couriers log should be updated according to the “Name”.

Results:



Extension: (Selenium IDE) - Selenium IDE - Hostel_Management_System — Mozilla Firefox

Project: Hostel_Management_System

nent-system-g5.netlify.app

- ✓ Add complaints
- ✓ Add couriers
- ✓ Add lost/found item
- ✓ Add new notice
- ✓ Delete Complaints
- ✓ Delete Notice
- ✓ Delete courier
- ✓ Delete lost/found item
- ✓ Filter Notice
- ✓ Login
- ✓ Search Notice
- ✓ Search courier

	Command	Target	Value
1	✓ open	/	
2	✓ set window size	1616x876	
3	✓ mouse over	linkText=Hostel Management System	
4	✓ mouse over	css=.MuiSvgIcon-root	
5	✓ click	css=.MuiSvgIcon-root	
6	✓ mouse out	css=.MuiSvgIcon-root	
7	✓ click	linkText=Couriers	
8	✓ click	css=.form-control	
9	✓ type	css=.form-control	Student 1

```
6.  mouseOut on css=.MuiSvgIcon-root OK                         23:35:35
7.  click on linkText=Couriers OK                               23:35:36
8.  click on css=.form-control OK                               23:35:36
9.  type on css=.form-control with value Student 1 OK           23:35:36
'Search courier' completed successfully                         23:35:36
```

11.Test Case: Add Lost/Found Item

Test Steps:

1. Navigate to the Lost/Found page using the sidebar.

2. Verify the item name, student ID, student contact number and item description the user wants to add is added to the respective text fields.
3. Verify the status of the item(lost or found).
4. Click on the “Submit” button
5. Verify that the item is added.
6. Repeat steps 1-5 for every role(student and admin).
7. Verify that multiple items can be added by a single user.
8. Verify that the number of items added is correct(no repetition).

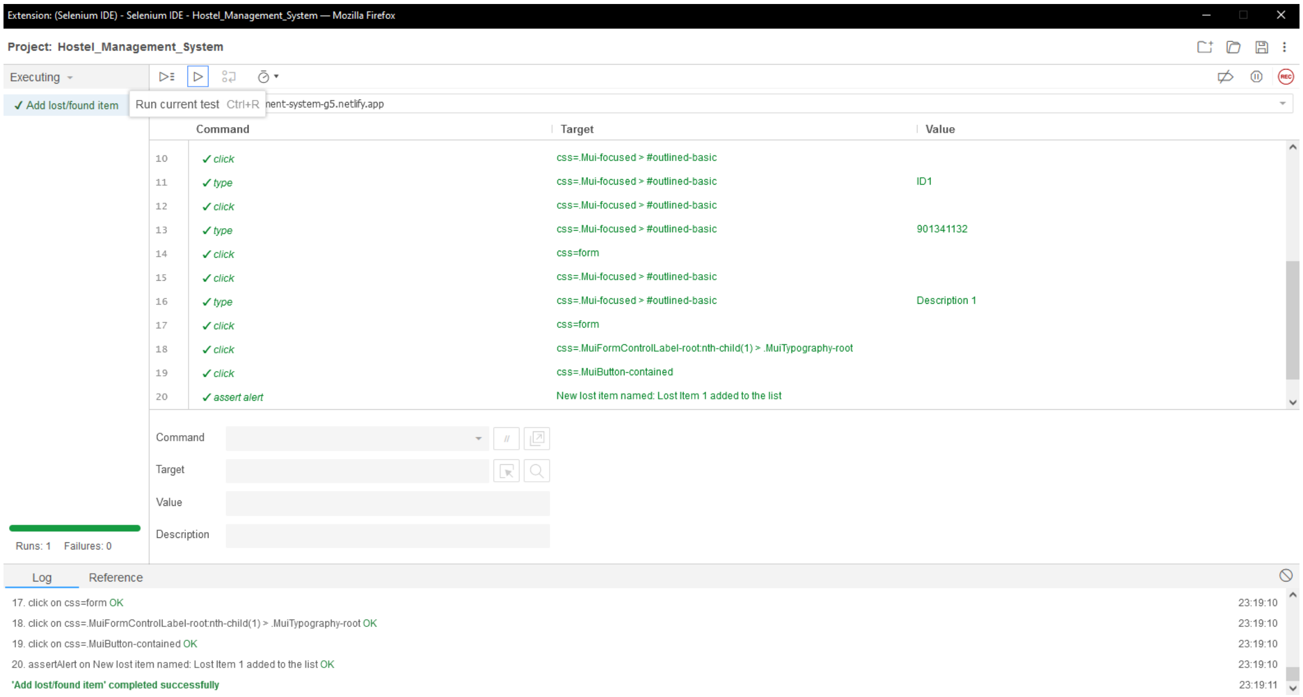
Test Data:

- The user has a valid account and is logged in to the hostel management website.
- The user loses/finds an item and wants to report it.

Expected Results:

- The user should be able to add items.
- The items log should be updated with the new added item.
- The user should be able to view the new item in the log.

Results:



Extension: (Selenium IDE) - Selenium IDE - Hostel_Management_System — Mozilla Firefox

Project: Hostel_Management_System

Executing

✓ Add lost/found item

nent-system-g5.netlify.app

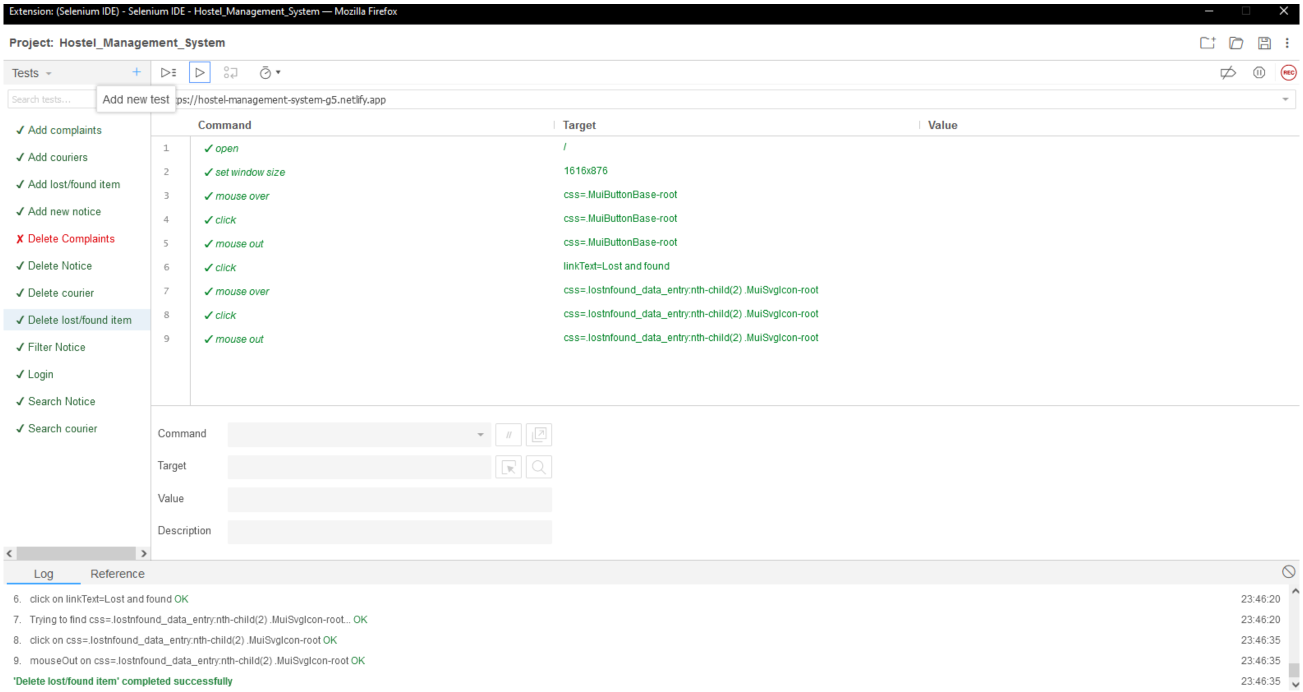
	Command	Target	Value
10	✓ click	css=.Mui-focused > #outlined-basic	
11	✓ type	css=.Mui-focused > #outlined-basic	ID1
12	✓ click	css=.Mui-focused > #outlined-basic	
13	✓ type	css=.Mui-focused > #outlined-basic	901341132
14	✓ click	css=form	
15	✓ click	css=.Mui-focused > #outlined-basic	
16	✓ type	css=.Mui-focused > #outlined-basic	Description 1
17	✓ click	css=form	
18	✓ click	css=.MuiFormControlLabel-root:nth-child(1) > .MuiTypography-root	
19	✓ click	css=.MuiButton-contained	
20	✓ assert alert	New lost item named: Lost Item 1 added to the list	

Runs: 1 Failures: 0

```
17. click on css=form OK                                                                  23:19:10
18. click on css=.MuiFormControlLabel-root:nth-child(1) > .MuiTypography-root OK          23:19:10
19. click on css=.MuiButton-contained OK                                                  23:19:10
20. assertAlert on New lost item named: Lost Item 1 added to the list OK                  23:19:10
'Add lost/found item' completed successfully                                              23:19:11
```

12.Test Case: Verify Delete Lost/Found Item Functionality on the website

Results:



Extension: (Selenium IDE) - Selenium IDE - Hostel_Management_System — Mozilla Firefox
Project: Hostel_Management_System
Tests
Search tests...
Add new test
Log
Reference

ps://hostel-management-system-g5.netlify.app

- ✓ Add complaints
- ✓ Add couriers
- ✓ Add lost/found item
- ✓ Add new notice
- ✗ Delete Complaints
- ✓ Delete Notice
- ✓ Delete courier
- ✓ Delete lost/found item
- ✓ Filter Notice
- ✓ Login
- ✓ Search Notice
- ✓ Search courier

	Command	Target	Value
1	✓ open	/	
2	✓ set window size	1616x876	
3	✓ mouse over	css=.MuiButtonBase-root	
4	✓ click	css=.MuiButtonBase-root	
5	✓ mouse out	css=.MuiButtonBase-root	
6	✓ click	linkText=Lost and found	
7	✓ mouse over	css=.lostnfound_data_entry:nth-child(2) .MuiSvgIcon-root	
8	✓ click	css=.lostnfound_data_entry:nth-child(2) .MuiSvgIcon-root	
9	✓ mouse out	css=.lostnfound_data_entry:nth-child(2) .MuiSvgIcon-root	

Command

Target

Value

Description

```
6.  click on linkText=Lost and found OK                                          23:46:20
7.  Trying to find css=.lostnfound_data_entry:nth-child(2) .MuiSvgIcon-root... OK 23:46:20
8.  click on css=.lostnfound_data_entry:nth-child(2) .MuiSvgIcon-root OK          23:46:35
9.  mouseOut on css=.lostnfound_data_entry:nth-child(2) .MuiSvgIcon-root OK       23:46:35
'Delete lost/found item' completed successfully                                   23:46:35
```