**Problem Description:**

Billboards Website Testing Requirement Specification

**Title:** ***D H Advertising Connect: Bridging Hoarding Vendors and Advertisers***

**Purpose:**

"D H Advertising Connect" serves as the go-to platform for hoarding vendors and advertisers seeking seamless collaboration in the world of outdoor advertising. By providing a straightforward and efficient interface, the platform enables advertisers to effortlessly connect with hoarding spaces that align with their campaign goals. Vendors, in turn, gain increased exposure and access to a broader audience of potential advertisers. With transparent communication tools and real-time booking features, D H Advertising Connect simplifies the outdoor advertising process, saving time and effort for both parties. This innovative platform is set to redefine the landscape, making outdoor advertising more accessible and impactful for businesses of all sizes.

**Testing Requirements for the Application**

***User Login Functionality:***

*Functional Requirements*

1. The application shall have a page for User Authentication:
2. The page shall verify that users can successfully authenticate using valid credentials(email and password).
3. the page shall ensure that login fails with incorrect credentials
4. The page shall allow to registration of a property only when the login is successful and if the usertype is either an agent or owner.
5. Verify the system behavior when a user attempts to log in without providing required email.
6. password is masked when the user enters the password
7. The page shall have a successful logout

*Password Recovery*

1. Test the password recovery process to ensure users can reset their passwords securely.
2. The user should provide his/her registered email and password to reset the password.
3. Ensure that login credentials are transmitted securely using encryption.

***Admin Login Functionality***

*Functional Requirements*

1. The application shall have a page for User Authentication:
2. The page shall verify that admin can successfully authenticate using valid credentials(username and password).
3. the page shall ensure that login fails with incorrect credentials
4. The page shall allow to access the dashboard only when the login is successful.
5. Verify the system behavior when a user attempts to log in without providing required username.
6. password is masked when the user enters the password
7. The page shall have a successful logout

*Password Recovery:*

1. Test the password recovery process to ensure users can reset their passwords securely.
2. The user should provide his/her registered email and password to reset the password.
3. Secure Transmission: Ensure that login credentials are transmitted securely using encryption.

***User signup Functionality***

*Functional Requirements*

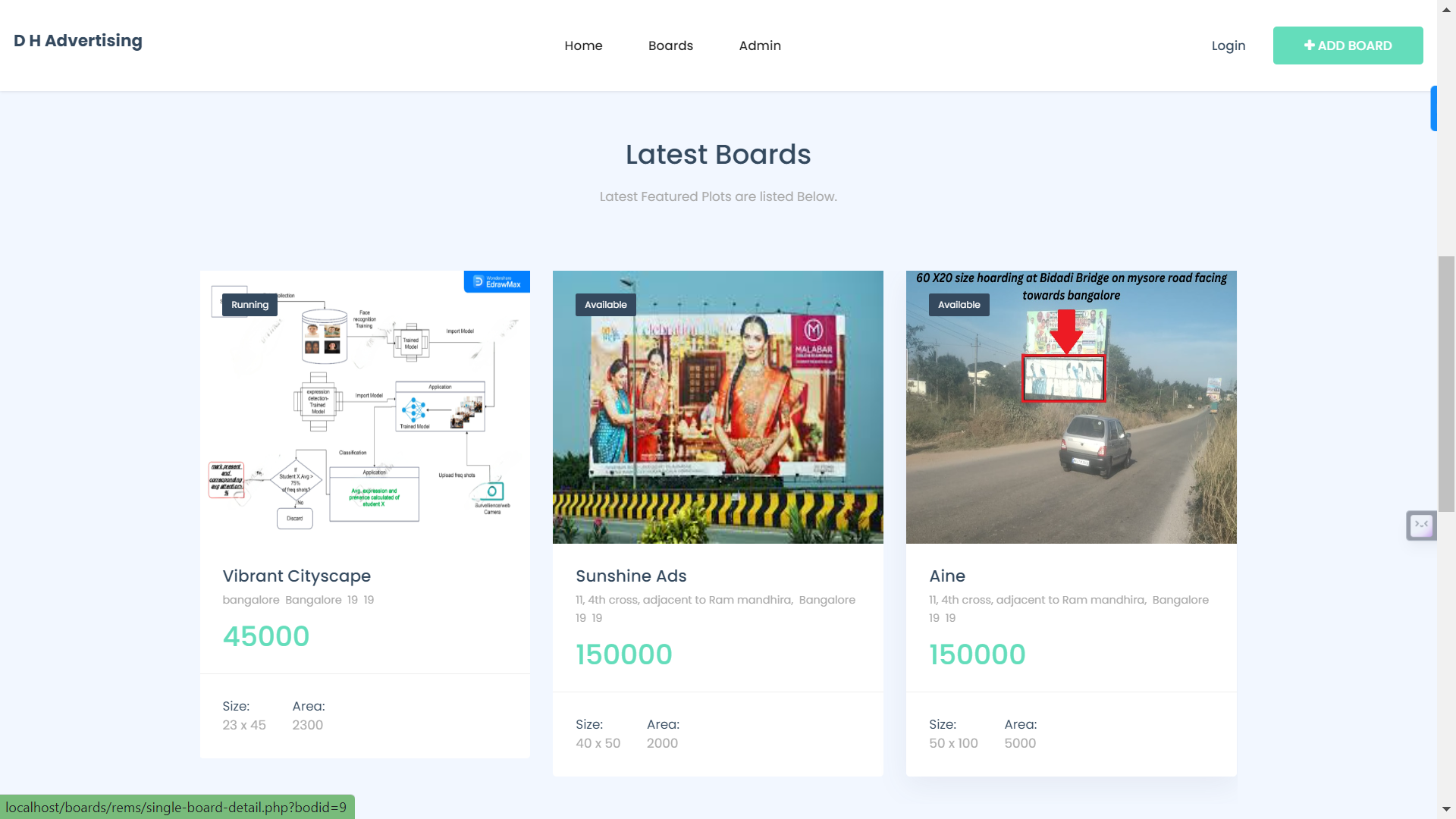
1. The application shall have a page for User signup
2. Verify that all fields on the page are required and validated correctly.
3. Verify that the "Email Address" field accepts valid email addresses only.
4. Verify that the "Mobile Number" field accepts valid phone numbers only.
5. Verify that the "Agent/Owner/Business" field is selected and is mandatory.
6. Password should be masked
7. Verify that the "REGISTER" button is enabled only when all required fields are filled in correctly.
8. Verify that after clicking the "REGISTER" button, the user is registered successfully and redirected to the login page.
9. Existing associated email and phone numbers cannot be used to register a new account.

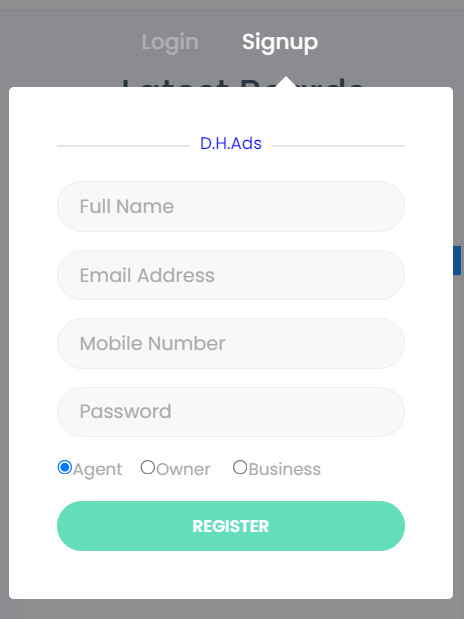
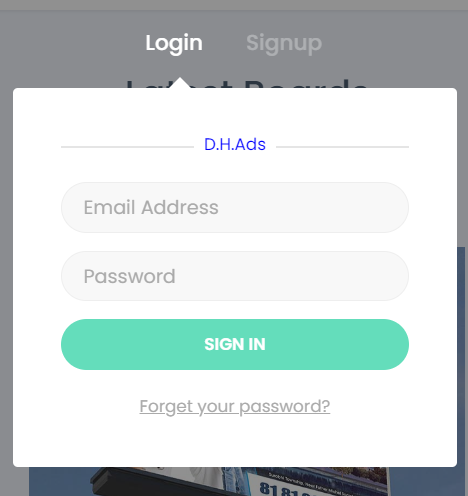
***Board registration Functionality***

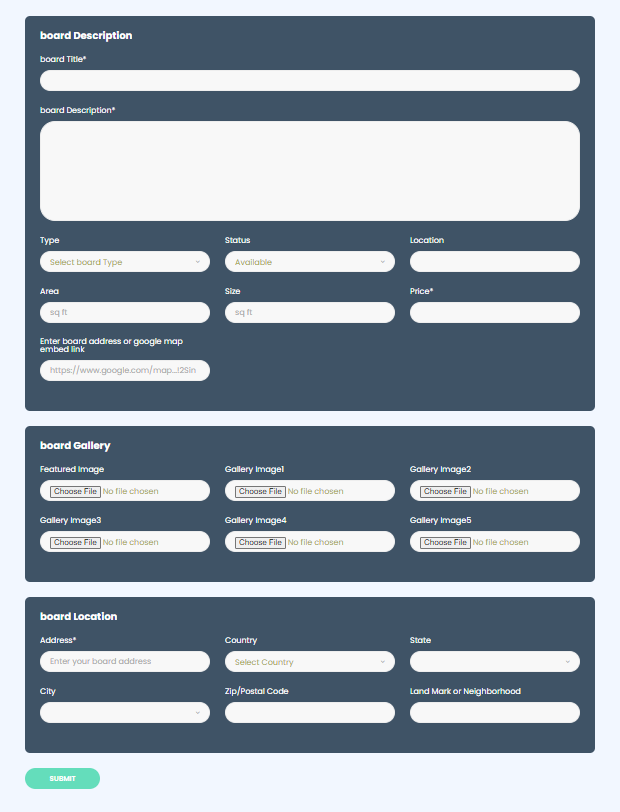
Functional Testing

1. Verify that all fields on the page are required and validated correctly.
2. Verify that the "board Title" field accepts valid board titles only.
3. Verify that the "board Description" field is optional .
4. Verify that the "Type" field is valid.
5. Verify that the "Status" field is a selection.
6. Verify that the "Location" field is valid.
7. Verify that the "Area" field is alphanumeric.
8. Verify that the "Size" field is numeric.
9. Verify that the "Price" field is numeric.
10. Verify that the "board address or google map embed link" field is optional.
11. Verify that the "Enter" button is enabled only when all required fields are filled in correctly.
12. Verify that after clicking the "Enter" button, the board is registered successfully and the user is redirected to the board list page.
13. Based on the screenshot of the board registration page for DH Advertising, here are some testing requirements:
14. Verify that users can upload featured images and gallery images of valid file types and sizes.
15. Verify that users can select the board location from a list of valid countries, states, cities, and zip codes.
16. Verify that the "SUBMIT" button is enabled only when all required fields are filled in correctly.
17. Verify that after clicking the "SUBMIT" button, the board is registered successfully and the user is redirected to a confirmation page.

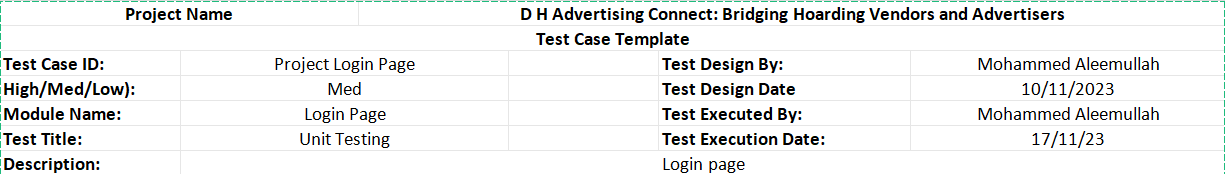
**Case study features and screenshots**



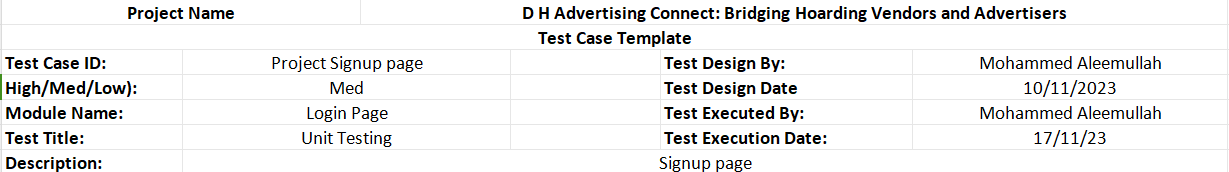




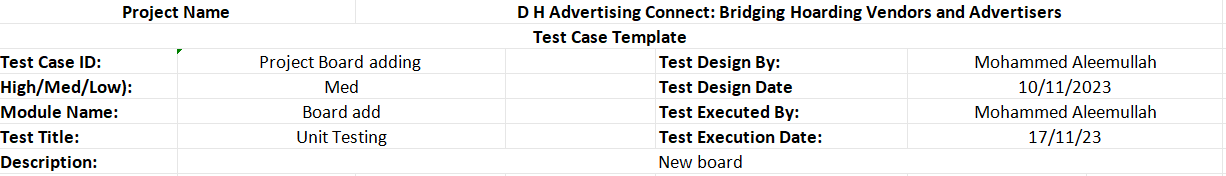
**Manual testing**

****

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Login page test cases** | | | | | | | |  |
| **Test Case ID** | **Test Description** | **Test Data** | **Input** | **Expected Outcome** | **Actual Outcome** | **Pass/Fail** | | Remarks |
| 1 | Valid Credentials | username:admin pass:admin | Valid username and password | Successful login redirected to homepage | Successful login redirected to homepage | | pass | fine |
| 2 | Invalid Username | username:admin2 pass:admin | Invalid username | Unsuccessful login, "Invalid details" | Unsuccessful login, "Invalid details" | | pass | fine |
| 3 | Invalid Password | username:admin pass:admin2 | Valid username, invalid password | Unsuccessful login, "Invalid details" | Unsuccessful login, "Invalid details" | | pass | fine |
| 4 | Empty Username | username: pass:admin | Empty username, valid password | "please fill out this field" | "please fill out this field" | | pass | fine |
| 5 | Empty Password | username:admin pass: | Valid username, empty password | "please fill out this field" | "please fill out this field" | | pass | fine |
| 6 | Logout | username:admin pass:admin | Logout and access again | Access denied, redirected to login | Access denied, redirected to login | | pass | fine |
| 7 | special characters not allowed | username:++++ password:++++ | special characters | not allowed to type | llowed | | fail | fine |
| 8 | SQL injection detect | username:"OR '1+1-- password:random | sql injection attack check | does not allow | does allow | | fail | fine |
| 9 | Blankspaces both | both credentials blank | blankspace check | prompt to fill it particularly | doesnot particularly prompt | | fail | fine |

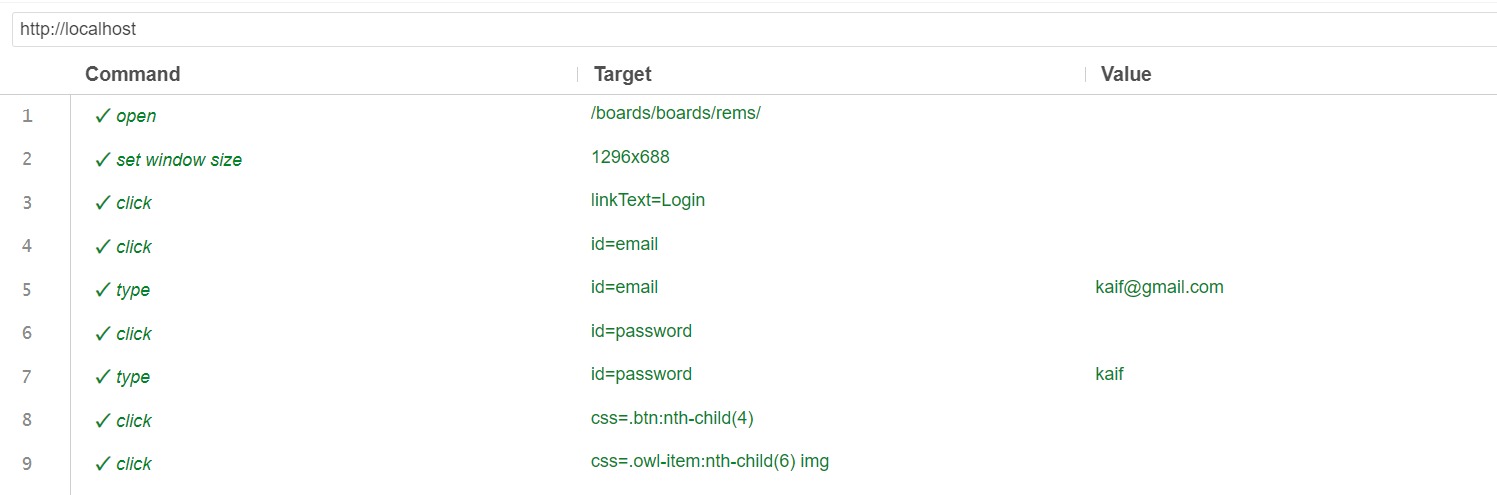
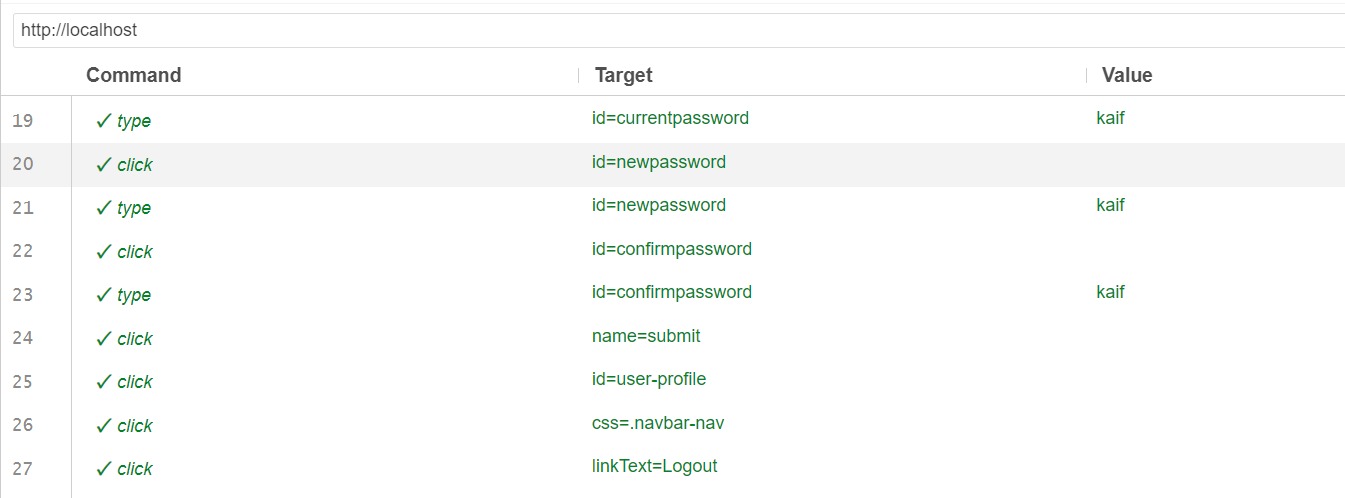
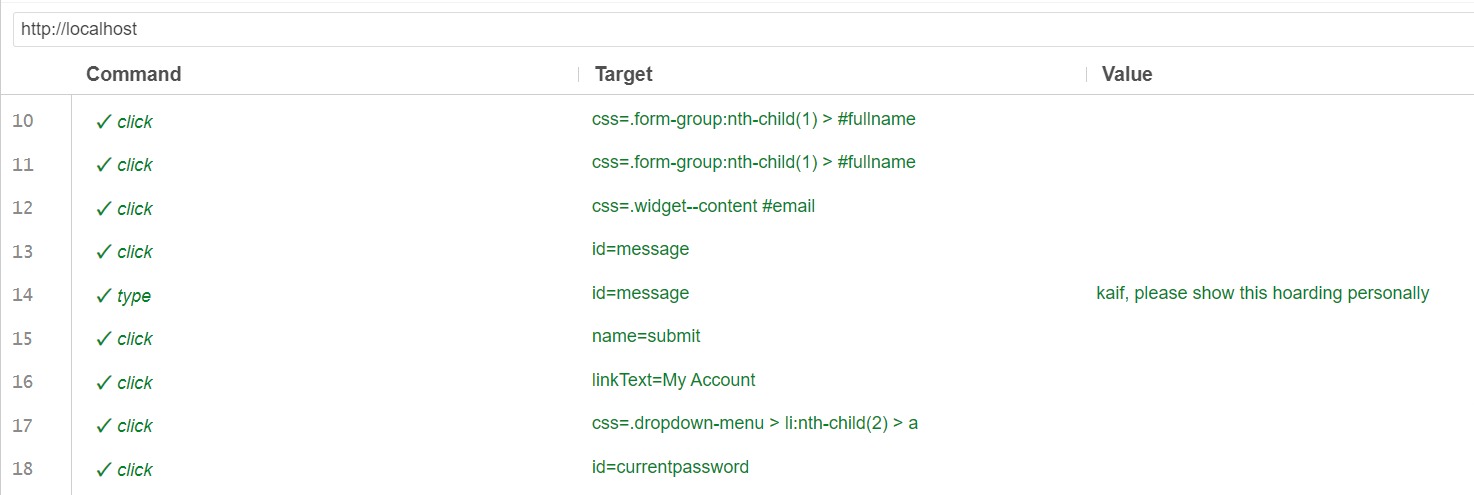


|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
| **Signup** | | | | | | |  |
| **Test Case ID** | **Test Description** | **Test Data** | **Input** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** |
| 7 | Verify that the user is able to register with a valid phone number. | Valid phone number (e.g., +1234567890) | Enter a valid phone number in the phone number field. | The user should be able to successfully register for an account and be redirected to the login page. | The user is able to successfully register for an account and is redirected to the login page. | Pass | done |
| 8 | Verify that the user is able to select their preferred role (Agent/Owner/Business). | None | Select a role from the drop-down menu. | The user should be able to select their preferred role from the drop-down menu. | The user is able to select their preferred role from the drop-down menu. | Pass | done |
| 9 | Verify that the user is able to see a confirmation message after registering successfully. | Valid email address and password | Valid email address and password | After submitting the registration form, the user should see a confirmation message indicating that they have successfully registered for an account. | The user sees a confirmation message indicating that they have successfully registered for an account. | Pass | done |
| 10 | Verify that the user is able to see error messages if they enter invalid data in any of the fields. | Invalid email address, invalid phone number, invalid password | Enter invalid data in any of the fields. | The user should see an error message indicating that the data they entered is invalid. | The user sees an error message indicating that the data they entered is invalid. | Pass | done |
| 11 | Verify that the registration form is responsive and looks good on different devices and screen sizes. | Resize the browser window to simulate different device sizes. | The registration form should resize and adapt to the different screen sizes without any issues. | The registration form resizes and adapts to the different screen sizes without any issues. | doesnot happen expectedly | fail | done |
| 7 | special characters not allowed | username:++++ password:++++ | special characters | not allowed to type | llowed | fail | fine |
| 8 | SQL injection detect | username:"OR '1+1-- password:random | sql injection attack check | does not allow | does allow | fail | fine |
| 9 | Blankspaces both | both credentials blank | blankspace check | prompt to fill it particularly | doesnot particularly prompt | fail | fine |



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Description | Test Data | Input | Expected Result | Actual Result | Pass/Fail | Remark |
| 1 | Upload valid JPEG | JPEG file | Select file | Image uploaded | Image displayed | Pass | done |
| 2 | Upload valid PNG | PNG file | Select file | Image uploaded | Image displayed | Pass | done |
| 3 | Upload invalid file format | PDF file | Select file | Error displayed | Error message shown | Pass | done |
| 4 | Upload multiple files | 2 JPGs | Select files | Images uploaded | Both images displayed | Fail | done |
| 5 | Remove uploaded image | Click button | Click button | Image removed | Image no longer displayed | Pass | done |
| 6 | Preview uploaded images | Click button | Click button | Preview opens | Images shown in preview | Fail | done |
| 7 | Large image resize | High-resolution JPG | Select image | Image resized | Image displayed at smaller size | Fail | done |
| 8 | Secure image storage | Upload image | Check server permissions | Restricted permissions | Image access limited | Pass | done |
| 10 | JPEG, PNG, GIF upload | Different formats | Select files | All uploaded | All formats displayed | Pass | done |
| 11 | Max image size limit | Large file | Select file | Error displayed | Size limit enforced | Pass | done |
| 12 | Repeated file upload | Same image twice | Select file | Duplicate detected | Warning message shown | Pass | done |
| 13 | Empty image upload | Blank file | Select file | Error displayed | Invalid file detected | Pass | done |
| 14 | Corrupted image upload | Damaged file | Select file | Error displayed | uploaded | Fail | done |
| 15 | Upload on slow connection | Limited bandwidth | Select file | Upload progress indicator | Image uploaded eventually | Pass | done |
| 16 | Upload from mobile device | Smartphone browser | Select file | Mobile-optimized experience | Image uploaded successfully | Fail | done |
| 17 | Upload from different browsers | Chrome, Firefox, Safari | Select file | Functionally consistent | Image uploaded regardless of browser | Pass | done |
| 18 | Image quality after upload | Compression level | Check visual fidelity | Minimal quality loss | Acceptable image quality | Fail | done |
| 19 | Automatic image optimization | Metadata removal | Analyze uploaded image | Metadata stripped | Unnecessary info removed | Pass | done |
| 20 | Image format integrity | File format conversion | Verify format post-upload | Original format preserved | Correct format maintained | Pass | done |
| 21 | Image orientation correction | Automatic rotation | Analyze uploaded image | Orientation corrected | Image displayed upright | Pass | done |
| 22 | Image caption functionality | Add captions | Enter text | Captions displayed | Captions shown alongside images | Fail | done |
| 23 | Image sorting and filtering | Sort by date uploaded | Select option | Images ordered correctly | Date order applied | Pass | done |

**Selenium IDE screenshot and Python Script**

**IDE Code:**

# Generated by Selenium IDE

import pytest

import time

import json

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.action\_chains import ActionChains

from selenium.webdriver.support import expected\_conditions

from selenium.webdriver.support.wait import WebDriverWait

from selenium.webdriver.common.keys import Keys

from selenium.webdriver.common.desired\_capabilities import DesiredCapabilities

class TestDefaultSuite():

def setup\_method(self):

self.driver = webdriver.Chrome()

self.vars = {}

def teardown\_method(self):

self.driver.quit()

def test\_aleem(self):

self.driver.get("http://localhost/boards/boards/rems/")

self.driver.set\_window\_size(1296, 688)

self.driver.find\_element(By.LINK\_TEXT, "Login").click()

self.driver.find\_element(By.ID, "email").click()

self.driver.find\_element(By.ID, "email").send\_keys("kaif@gmail.com")

self.driver.find\_element(By.ID, "password").click()

self.driver.find\_element(By.ID, "password").send\_keys("kaif")

self.driver.find\_element(By.CSS\_SELECTOR, ".btn:nth-child(4)").click()

self.driver.find\_element(By.CSS\_SELECTOR, ".owl-item:nth-child(6) img").click()

self.driver.find\_element(By.CSS\_SELECTOR, ".form-group:nth-child(1) > #fullname").click()

self.driver.find\_element(By.CSS\_SELECTOR, ".form-group:nth-child(1) > #fullname").click()

self.driver.find\_element(By.CSS\_SELECTOR, ".widget--content #email").click()

self.driver.find\_element(By.ID, "message").click()

self.driver.find\_element(By.ID, "message").send\_keys("kaif, please show this hoarding personally")

self.driver.find\_element(By.NAME, "submit").click()

self.driver.find\_element(By.LINK\_TEXT, "My Account").click()

self.driver.find\_element(By.CSS\_SELECTOR, ".dropdown-menu > li:nth-child(2) > a").click()

self.driver.find\_element(By.ID, "currentpassword").click()

self.driver.find\_element(By.ID, "currentpassword").send\_keys("kaif")

self.driver.find\_element(By.NAME, "submit").click()

self.driver.find\_element(By.ID, "user-profile").click()

self.driver.find\_element(By.CSS\_SELECTOR, ".navbar-nav").click()

self.driver.find\_element(By.LINK\_TEXT, "Logout").click()

test = TestDefaultSuite()

test.setup\_method()

test.test\_aleem()

test.teardown\_method()

**Selenium Web driver script**

import time

from selenium import webdriver

from selenium.webdriver.common.by import By

driver = webdriver.Chrome()

driver.get("http://localhost/boards/boards/rems/")

time.sleep(1)

driver.set\_window\_size(1296, 688)

time.sleep(1)

driver.find\_element(By.LINK\_TEXT, "Login").click()

driver.find\_element(By.ID, "email").click()

driver.find\_element(By.ID, "email").send\_keys("kaif@gmail.com")

driver.find\_element(By.ID, "password").click()

driver.find\_element(By.ID, "password").send\_keys("kaif")

time.sleep(1)

driver.find\_element(By.CSS\_SELECTOR, ".btn:nth-child(4)").click()

time.sleep(1)

driver.find\_element(By.CSS\_SELECTOR, ".owl-item:nth-child(6) img").click()

driver.find\_element(By.CSS\_SELECTOR, ".form-group:nth-child(1) > #fullname").click()

driver.find\_element(By.CSS\_SELECTOR, ".form-group:nth-child(1) > #fullname").click()

driver.find\_element(By.CSS\_SELECTOR, ".widget--content #email").click()

driver.find\_element(By.ID, "message").click()

driver.find\_element(By.ID, "message").send\_keys("kaif, please show this hoarding personally")

time.sleep(1)

driver.find\_element(By.NAME, "submit").click()

time.sleep(3)

driver.find\_element(By.LINK\_TEXT, "My Account").click()

time.sleep(1)

driver.find\_element(By.CSS\_SELECTOR, ".dropdown-menu > li:nth-child(2) > a").click()

time.sleep(1)

driver.find\_element(By.ID, "currentpassword").click()

driver.find\_element(By.ID, "currentpassword").send\_keys("kaif")

driver.find\_element(By.ID, "newpassword").click()

driver.find\_element(By.ID, "newpassword").send\_keys("kaif")

driver.find\_element(By.ID, "confirmpassword").click()

driver.find\_element(By.ID, "confirmpassword").send\_keys("kaif")

time.sleep(1)

driver.find\_element(By.NAME, "submit").click()

time.sleep(1)

driver.find\_element(By.ID, "user-profile").click()

time.sleep(1)

driver.find\_element(By.CSS\_SELECTOR, ".navbar-nav").click()

time.sleep(1)

driver.find\_element(By.LINK\_TEXT, "Logout").click()

driver.quit()