

Test Cases

Functional Test Cases

Feature: LOGIN FUNCTIONALITY

1. Login with valid credentials
 - a. **Given** the login screen is displayed
 - b. **When** the user enters a valid username and password
 - c. **When** the user clicks the login button
 - d. **Then** the user is redirected to the dashboard page
2. Login with invalid credentials
 - a. **Given** the login screen is displayed
 - b. **When** the user leaves the username and password fields blank
 - c. **When** the user clicks the login button
 - d. **Then** the user is redirected to the 404 page
3. Login with invalid password
 - a. **Given** the login screen is displayed
 - b. **When** the user enters valid username and invalid password
 - c. **When** the user clicks the login button
 - d. **Then** the user is redirected to the 404 page
4. Login with invalid username
 - a. **Given** the login screen is displayed
 - b. **When** the user enters invalid username and valid password
 - c. **When** the user clicks the login button
 - d. **Then** the user is redirected to the 404 page
5. Login with empty password
 - a. **Given** the login screen is displayed

- b. **When** the user enters username and empty password
 - c. **When** the user clicks the login button
 - d. **Then** the user should see a popup with text **Please fill out this field**
- 6. Login with empty username
 - a. **Given** the login screen is displayed
 - b. **When** the user enters empty username and valid password
 - c. **When** the user clicks the login button
 - d. **Then** the user should see a popup with text **Please fill out this field**
- 7. Login with empty credentials
 - a. **Given** the login screen is displayed
 - b. **When** the user enters empty username and password
 - c. **When** the user clicks the login button
 - d. **Then** the user should see a popup with text **Please fill out this field**
- 8. Case sensitivity for username and password fields
 - a. **Given** the login screen is displayed
 - b. **When** the user enters random case username and password
 - c. **When** the user clicks the login button
 - d. **Then** the user is redirected to the 404 page
- 9. Password masking
 - a. **Given** the login screen is displayed
 - b. **When** the user enters random case username and password
 - c. **Then** validate the password field is masked
- 10. Username and Password field validation
 - a. **Given** the login screen is displayed

- b. **When** the user see the username and password
- c. **Then** do field validation for username and password field
- d. (Field validation includes long username and password, special chars etc)

Feature: Contact Form Functionality

1. Submit contact form with all details
 - a. **Given** the user is logged in
 - b. **When** the user navigates to Contact Page
 - c. **When** the Contact Page is displayed
 - d. **When** the user enters "<name>", "<email>" and "<message>" in the contact form
 - e. **When** the user clicks the send button
 - f. **Then** the contact form is submitted and the alert window is visible
2. Submit contact form with missing name
 - a. **Given** the user is logged in
 - b. **When** the user navigates to Contact Page
 - c. **When** the Contact Page is displayed
 - d. **When** the user enters missing "<name>", "<email>" and "<message>" in the contact form
 - e. **When** the user clicks the send button
 - f. **Then** the contact form is submitted and the alert window is visible
3. Submit contact form with missing email
 - a. **Given** the user is logged in
 - b. **When** the user navigates to Contact Page
 - c. **When** the Contact Page is displayed
 - d. **When** the user enters "<name>", missing "<email>" and "<message>" in the contact form

- e. **When** the user clicks the send button
 - f. **Then** the contact form is submitted and the alert window is visible
4. Submit contact form with missing name and email
- a. **Given** the user is logged in
 - b. **When** the user navigates to Contact Page
 - c. **When** the Contact Page is displayed
 - d. **When** the user enters missing "<name>", missing "<email>" and "<message>" in the contact form
 - e. **When** the user clicks the send button
 - f. **Then** the contact form is submitted and the alert window is visible
5. Submit contact form with missing message
- a. **Given** the user is logged in
 - b. **When** the user navigates to Contact Page
 - c. **When** the Contact Page is displayed
 - d. **When** the user enters "<name>", "<email>" and missing "<message>" in the contact form
 - e. **When** the user clicks the send button
 - f. **Then** the contact form is submitted and the alert window is visible
6. Submit contact form with all fields missing
- a. **Given** the user is logged in
 - b. **When** the user navigates to Contact Page
 - c. **When** the Contact Page is displayed
 - d. **When** the user enters all fields missing "<name>", "<email>" and "<message>" in the contact form
 - e. **When** the user clicks the send button
 - f. **Then** the contact form is submitted and the alert window is visible

7. Submit contact form with all fields as numeric values
 - a. **Given** the user is logged in
 - b. **When** the user navigates to Contact Page
 - c. **When** the Contact Page is displayed
 - d. **When** the user enters all fields as numeric values "<name>", "<email>" and "<message>" in the contact form
 - e. **When** the user clicks the send button
 - f. **Then** the contact form is submitted and the alert window is visible
8. Submit contact form with all fields as special characters
 - a. **Given** the user is logged in
 - b. **When** the user navigates to Contact Page
 - c. **When** the Contact Page is displayed
 - d. **When** the user enters all fields as special characters "<name>", "<email>" and "<message>" in the contact form
 - e. **When** the user clicks the send button
 - f. **Then** the contact form is submitted and the alert window is visible
9. Submit contact form with all fields as long strings
 - a. **Given** the user is logged in
 - b. **When** the user navigates to Contact Page
 - c. **When** the Contact Page is displayed
 - d. **When** the user enters all fields as long strings "<name>", "<email>" and "<message>" in the contact form
 - e. **When** the user clicks the send button
 - f. **Then** the contact form is submitted and the alert window is visible

Feature: Buy Tour Tickets Functionality

1. Buy Tickets flow with all fields
 - a. **Given** the user is logged in

- b. **When** the user navigates to Tour Page
 - c. **When** the Tour Page is displayed
 - d. **When** the user clicks the Buy Tickets button <position>
 - e. **Then** the user should see the Ticket Purchase Window
 - f. **When** the user enters "<quantity>" and "<contact>" in buy tickets form
 - g. **When** the user clicks the Pay button
 - h. **Then** the user should see the Ticket Results page
 - i. **Then** the user clicks the Close button
2. Buy Tickets flow with Quantity empty
- a. **Given** the user is logged in
 - b. **When** the user navigates to Tour Page
 - c. **When** the Tour Page is displayed
 - d. **When** the user clicks the Buy Tickets button <position>
 - e. **Then** the user should see the Ticket Purchase Window
 - f. **When** the user enters empty "<quantity>" and "<contact>" in buy tickets form
 - g. **When** the user clicks the Pay button
 - h. **Then** the user should see the Ticket Results page
 - i. **Then** the user clicks the Close button
3. Buy Tickets flow with contact details empty
- a. **Given** the user is logged in
 - b. **When** the user navigates to Tour Page
 - c. **When** the Tour Page is displayed
 - d. **When** the user clicks the Buy Tickets button <position>
 - e. **Then** the user should see the Ticket Purchase Window
 - f. **When** the user enters "<quantity>" and empty "<contact>" in buy tickets form
 - g. **When** the user clicks the Pay button
 - h. **Then** the user should see the Ticket Results page
 - i. **Then** the user clicks the Close button

4. Buy Tickets flow with all details empty
 - a. **Given** the user is logged in
 - b. **When** the user navigates to Tour Page
 - c. **When** the Tour Page is displayed
 - d. **When** the user clicks the Buy Tickets button <position>
 - e. **Then** the user should see the Ticket Purchase Window
 - f. **When** the user enters empty "<quantity>" and empty "<contact>" in buy tickets form
 - g. **When** the user clicks the Pay button
 - h. **Then** the user should see the Ticket Results page
 - i. **Then** the user clicks the Close button
5. Cancel Buy Tickets flow
 - a. **Given** the user is logged in
 - b. **When** the user navigates to Tour Page
 - c. **When** the Tour Page is displayed
 - d. **When** the user clicks the Buy Tickets button <position>
 - e. **Then** the user should see the Ticket Purchase Window
 - f. **When** the user enters "<quantity>" and "<contact>" in buy tickets form
 - g. **When** the user clicks the cancel button
 - h. **Then** the Ticket Purchase Window should close

Security Test Cases

1. Error Message Consistency
 - a. **Given** the login screen is displayed
 - b. **When** the user enters incorrect credentials
 - c. **Then** the error message does not reveal whether the username or password was incorrect
2. Session Hijacking

- a. **Given** the user is logged in successfully
- b. **When** the session token is reused in another browser
- c. **Then** the system invalidates the token and requires reauthentication

Usability Test Cases

1. Field Placeholder Text
 - a. **Given** the login screen is displayed
 - b. **When** the user views the username and password fields
 - c. **Then** the placeholder text provides guidance (e.g., "Username")
2. Responsive Design
 - a. **Given** the login screen is shown on various devices (desktop, tablet, mobile)
 - b. **When** the user interacts with the fields and button
 - c. **Then** the layout adjusts properly
3. Accessibility
 - a. **Given** the login screen is displayed
 - b. **When** a screen reader or tab navigation is used
 - c. **Then** all fields and buttons are accessible

Performance Test Cases

1. Page Load Time
 - a. **Given** the login screen is requested
 - b. **When** the page loads
 - c. **Then** it loads within 2 seconds
2. Concurrent Login Requests

- a. **Given** multiple users try to log in simultaneously
 - b. **When** 1000 login requests are sent concurrently
 - c. **Then** the system handles the load without crashing
3. System Behavior During High Load
- a. **Given** the system is under peak load
 - b. **When** a user tries to log in
 - c. **Then** the system responds within acceptable time limits

Edge Case Test Scenarios

1. Browser Back Button
- a. **Given** the user is logged in
 - b. **When** the user clicks the browser back button
 - c. **Then** the system does not display the login screen without logging out
2. Cross-Browser Compatibility
- a. **Given** the login screen is displayed
 - b. **When** the user accesses it using various browsers (Chrome, Firefox, Safari, Edge)
 - c. **Then** the login functionality works seamlessly