**UI Automation Test Plan / Strategy Document**

**Project: Login & User Management Web App**

Automation Type: UI Functional Testing

Framework: Cucumber + JavaScript (Node.js) + Selenium

**Tooling:**

- Test Runner: Cucumber.js

- Assertions: Chai

- Browser Automation: Playwright / Selenium

- CI: GitHub Actions / Postman CLI (optional)

**Objective**

Automate end-to-end user interactions for the login and user management features of the application to ensure core flows work consistently across releases.

**Scope**

* Login functionality (valid and invalid cases)
* Signup / new user creation
* View user list
* Edit user details
* Delete user
* UI element presence
* Success/failure messages

**Test Design Strategy**

**Test Framework:**

* BDD-style step definitions using Gherkin
* Feature files for each module (login.feature, userManagement.feature)
* JavaScript step implementations in step\_definitions folder

**Structure:**

bdd-test/

├── features/

│ ├── login.feature

│ ├── userManagement.feature

│ └── step\_definitions/

│ ├── loginSteps.js

│ ├── userSteps.js

│ └── hooks.js

**Setup:**

* Before hooks initialize browser and page context
* After hooks close browser sessions
* Each scenario starts from a clean app state

Sample Feature: Login

Feature: User Login

Scenario: Successful login with valid credentials

Given I am on the login page

When I enter valid username and password

And I click the login button

Then I should see the welcome page

Scenario: Failed login with wrong credentials

Given I am on the login page

When I enter invalid username and password

Then I should see "Invalid username or password"

**Test Data Strategy**

* Hardcoded data used for login/signup (e.g. admin/admin123)
* Future plan: move data to JSON file or external fixture for reusability

Test Execution Strategy

Local Execution:

Via npm run test or directly using cucumber-js

CI Execution:

Run tests in GitHub Actions / any CI pipeline

Generate HTML or JSON reports

Reporting

Console summary (pass/fail)

Optional: Use plugins like cucumber-html-reporter for HTML reports

CI results visible on GitHub Actions dashboard

Risks & Mitigations

**Risk Mitigation**

DOM structure changes Use stable selectors like data-testid

Test flakiness due to timing issues Use await and proper waits

Manual test data management Automate data setup and cleanup

**Screenshots:**













