

Test Plan for User Management Application

1. What is Being Tested

This test suite verifies the functionality of a User Management system including:

- User Authentication: Login with valid and invalid credentials.
- User CRUD operations:
 - Create a new user
 - Retrieve user list
 - Update an existing user
 - Delete a user
- UI Functionality: The React frontend correctly performs and reflects all CRUD operations.
- API Endpoints: RESTful endpoints respond correctly to valid and invalid requests.

2. Test Coverage Areas

- Login page input validation and success/failure scenarios.
- Creating a new user and verifying it appears in the UI list.
- Editing an existing user and asserting updated data is shown.
- Deleting a user and confirming removal from the list.
- UI alerts and notifications on success or error.
- API Tests (Backend Automation)
 - POST /login: Valid and invalid login attempts, response validation.
 - GET /users: Fetch user list and validate response format.
 - POST /users: Create user with valid and invalid payloads.
 - PUT /users/:id: Update user details, including non-existent user cases.
 - DELETE /users/:id: Delete existing and non-existing users, validate response codes.

3. Tools Used and Why

Tool	Purpose	Reason
React Testing Library / Jest	Unit and integration tests on React components	Popular, easy to integrate, supports JSX and hooks

Cypress

End-to-end UI functional
testing

Provides powerful, developer-
friendly UI automatio

4. How to Run the Tests

Prerequisites Node.js installed and Project dependencies installed via npm install

In the terminal enter the following commands:

```
cd client
```

```
npm start
```

While that is running, in a new terminal run the following command:

```
npx cypress run
```

Cypress should open at this point”

Click e2e testing

Click chrome

Click E2E Testing in Chrome

Click app.cy.js

5. Assumptions and Limitations

The public API or mock API may not persist data; hence the local state is maintained for UI testing.

Authentication is mocked/simplified; no real token validation or expiry.

Tests assume stable network and environment; flaky tests may occur if server is unavailable.

UI tests rely on element IDs which must remain consistent.