

UserList Component

Framework

JavaScript

Goal

Create a JavaScript app in a TDD manner that uses Jest to test asynchronous behaviour by simulating data fetching and verifying UI updates during loading, success, and error states.

Instructions

1. Create a userService with fetchUsers() that returns a Promise
2. Use jest.mock() to control the returned data or simulate errors
3. Write a test that shows "Loading..." initially
4. Test that user data renders as a list after promise resolves
5. Test that error message appears if fetch fails
6. Use DOM manipulation to update the user list dynamically

Tech & Tools

- Jest mocks (jest.mock(), mockResolvedValue, mockRejectedValue)
- async/await for promise handling in tests
- DOM manipulation: innerHTML, appendChild
- Error handling with try/catch blocks

What to Test

- "Loading..." displays initially
- User list renders correctly after data loads (check for user names in DOM)
- Error message displays if fetch fails
- Loading indicator disappears after success or error
- No actual network requests are made