UserListComponent

## Framework

Angular

# Goal

Test a component that loads data asynchronously using a mocked service. Handle and verify all three UI states: loading, success, and error.

# Instructions

1. Create a UserService with a method getUsers(): Observable<User[]>.
2. In your test, provide a stubbed version of the service using jest.fn() and of() or throwError() from RxJS.
3. Write a test to verify that “Loading…” appears before data is returned.
4. Use fixture.detectChanges() and fixture.whenStable() to wait for data loading to complete.
5. Test that a list of users appears after a successful fetch.
6. Write a separate test that simulates an error and verifies that an error message or fallback UI is shown.

# Tech & Tools

* Angular TestBed, ReactiveFormsModule, HttpClientTestingModule (if applicable)
* RxJS: of, throwError
* fixture.detectChanges(), fixture.whenStable(), nativeElement.querySelector

# What to Test

* “Loading…” message appears initially
* User list appears after data resolves
* Error message is shown when service throws
* No actual HTTP requests are made