UserFormComponent

## Framework

Angular

# Goal

Use a mock service to verify behaviour. Test that form submission correctly calls a method on an injected service using a Jest spy.

# Instructions

1. Create a UserService with a method saveUser(data: { name: string }).
2. In your test, create a mock for UserService using jest.fn() and useValue in TestBed.
3. Render the component with Reactive Forms, fill out the form using nativeElement.querySelector, and trigger submission.
4. Write a test to verify that the saveUser method was called with the correct object when the form is submitted.
5. Include a test to ensure it doesn’t call the service if the form is invalid.
6. Use fixture.detectChanges() and fixture.whenStable() as needed to reflect changes.

# Tech & Tools

* Angular ReactiveFormsModule, TestBed
* jest.fn() for service method mocking
* ComponentFixture, spyOn, FormBuilder, EventEmitter

# What to Test

* Valid form submission calls saveUser() with correct data
* Invalid form submission does not trigger service call
* Form values update correctly in the DOM
* Optional: success message appears after submit