Investigating the UI Testing of the open source project we've chosen

Here, the fundamental testing structure is as follows:

```
import React from 'react'
import { mount } from 'enzyme'
import Spacer from 'react-storefront/Spacer'

describe('Spacer', () => {
  let wrapper

  afterEach(() => {
    wrapper.unmount()
  })

  it('should render component', () => {
    wrapper = mount(<Spacer />)

    expect(wrapper.find(Spacer).exists()).toBe(true)
  })
})
```

The code mentioned above is for label testing. Importing the necessary items is the first thing you do. Then a class type with the keyword "describe" appears, in which you provide all the pertinent test cases. For instance, we are working with labels in the code above, hence only labels-related scenarios will be discussed here.

Following that, you specify a certain type of variable (in this example, "wrapper"), which is then used in all subsequent test cases.

After that, you always unmount in the "afterEach" section that comes right before our first test case. (A mounting note will be present for each test instance.)

Our test begins with the word "it," and you provide a plain-English description of that test in the parentheses that follow. Next, we describe and explain what a test should accomplish in the major portion of our test. We mount first, then construct a test (using testprop="testprop" from the example above). the statement (expect(wrapper.find (Typography). prop('testprop')). We anticipate this test to return a value of toBe('testprop')). We anticipate using the term "testprop" for this test.
.

Here is another example where we expect Breadcrumb to be empty when there is no element:

```
  it('should render empty span when no items provided', () => {
    wrapper = mount(<Breadcrumbs items={null} />)

    expect(wrapper.find(Container).children.length).toBe(1)
  })
```

In A similar manner, our exam begins with "it" and then gives a brief, plain-English overview of what to expect. After that, describe our test and then lastly state what we expect from the test.