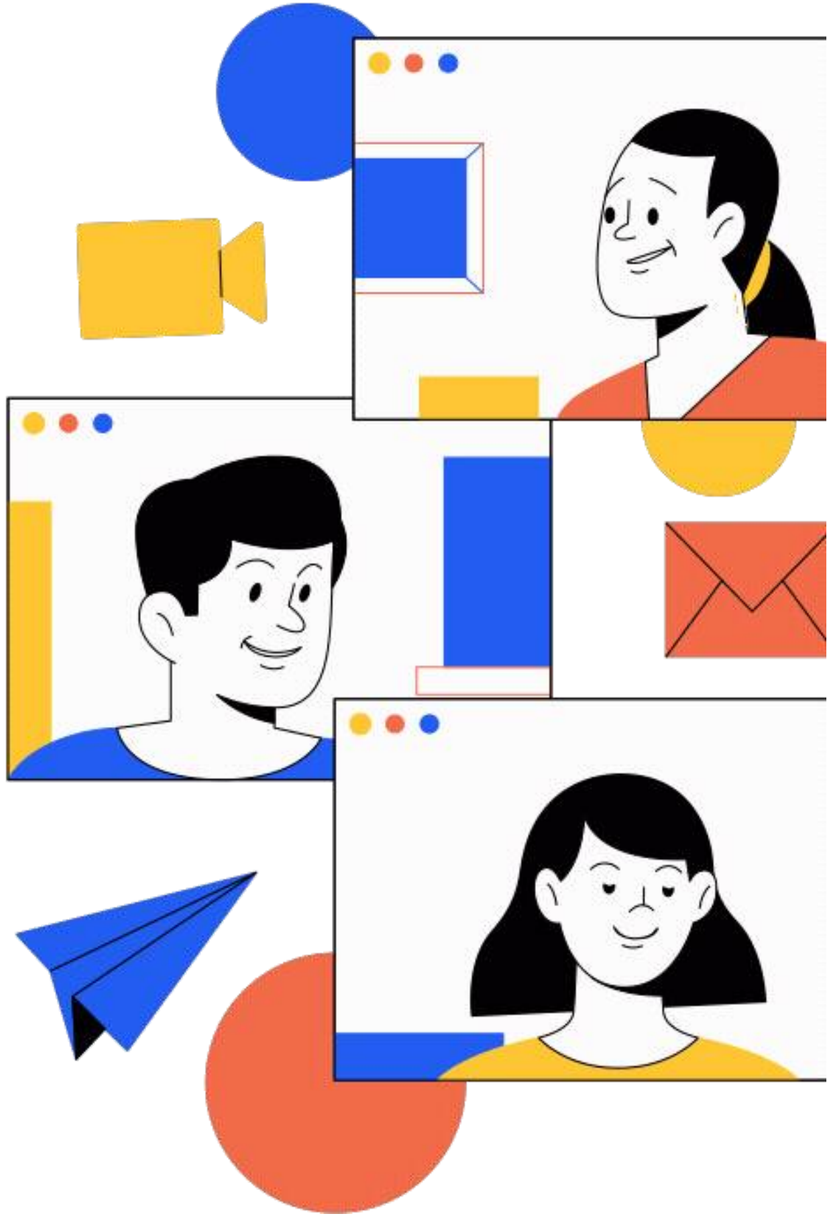
UDACITY

Week 8 - Testing with Jest

Test your code like hackers



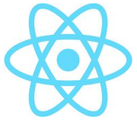
Ahmed Fouad Lotfy

React Session Lead

Agenda

What we'll cover in this session

- What is Jest?
- How it works?
- Jest Matchers
- Testing in React
- Live Demo



Why Testing (Jest)?

- **Why Testing is Crucial?** It helps catch bugs early and improve code quality. Ensures the application works as expected through different use cases. Provides confidence when refactoring or adding new features.

- **What is Jest?** it’s a JavaScript testing framework developed by Facebook. It is widely used for testing React applications due to its simplicity and rich features.

How it Works?

- **Installing Jest:** Use ***npm install --save-dev jest*** to add Jest to your project.

- **Running Jest:** Tests can be run with the command: ***npm test*** or ***jest***.

```
$ npm install --save-dev jest
npm WARN deprecated istanbul-lib-hook@1.2.1: 1.2.0 should have been a major vers
ion bump

npm WARN react-http-request@1.0.4 requires a peer of react@^16.0.0 but none is i
nstalled. You must install peer dependencies yourself.
```

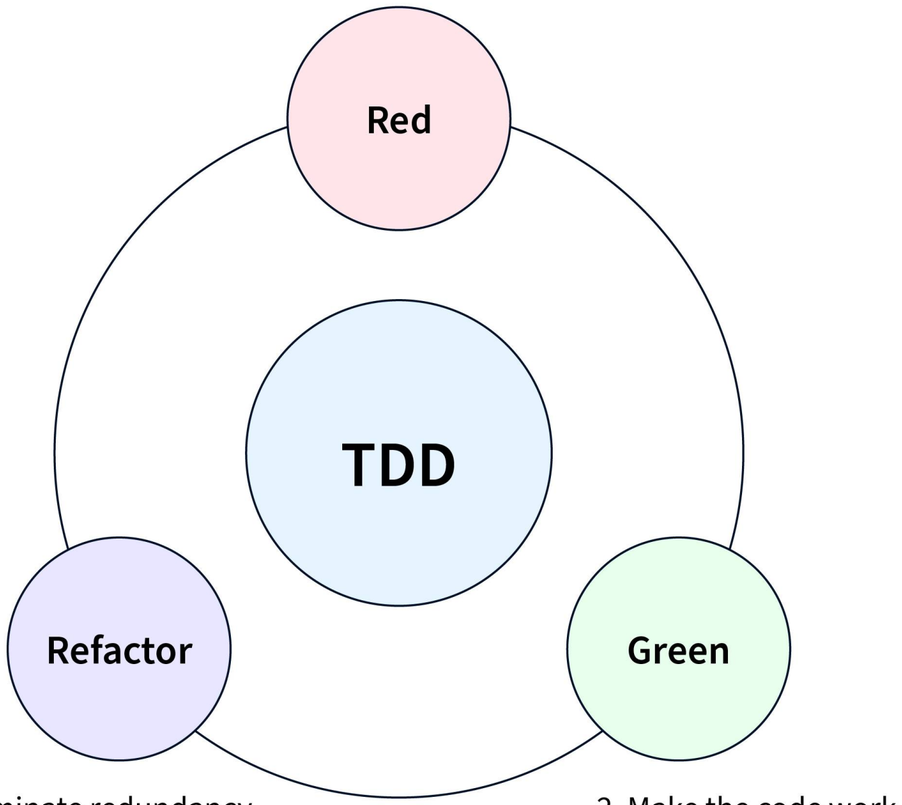
```
PS npm test

> learning-jest@1.0.0 test
> jest

 PASS  ./index.test.js
  FizzBuzz
    √ [3] should result in "fizz" (2 ms)
    √ [5] should result in "buzz" (1 ms)
    √ [15] should result in "fizzbuzz" (1 ms)
    √ [1,2,3] should result in "1, 2, fizz" (1 ms)

Test Suites: 1 passed, 1 total
Tests:       4 passed, 4 total
Snapshots:   0 total
Time:        0.586 s, estimated 1 s
Ran all test suites.
```

Jest

How it works?

The mantra of Test-Driven Development (TDD) is “red, green, refactor.”

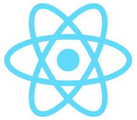
1. Write a test that fails



3. Eliminate redundancy

2. Make the code work

Jest

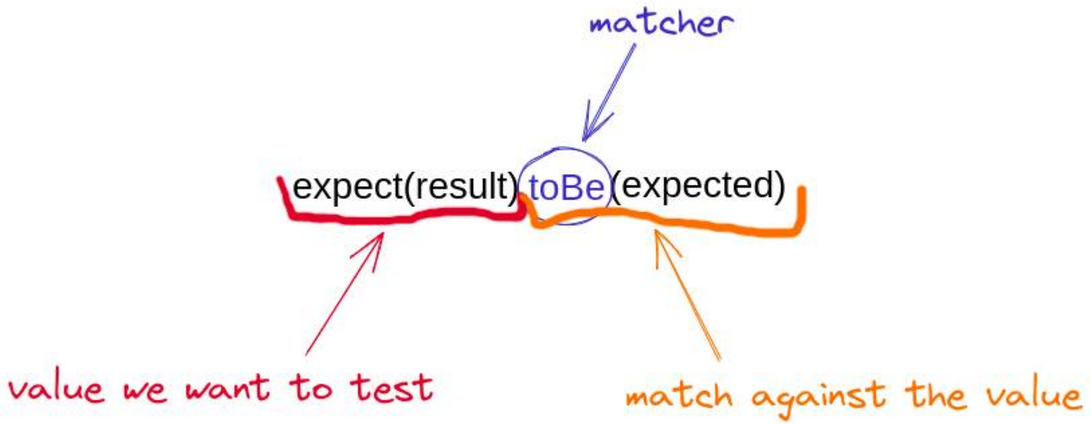


What are Jest Matchers?

	No Matches	1 Match	Multiple Matches
Single Element			
getBy	Throws Error	Returns Element	Throws Error
queryBy	Returns NULL	Returns Element	Throws Error
Multiple Elements			
getAllBy	Throws Error	Returns Array	Returns Array
queryAllBy	Returns Empty Array	Returns Array	Returns Array

What are Jest Matchers?

- **Definition:** Jest matchers help you assert specific conditions in your tests. Common matchers include .toBe(), .toEqual(), .toHaveLength(), etc.



Jest

Synchronous vs Asynchronous Functions?

- **How to Test Synchronous Functions:** Testing a simple function that adds two numbers.

```
test('adds 1 + 2 to equal 3', () => {
  expect(1 + 2).toBe(3);
});
```

- **How to Test Asynchronous Functions:** Using async/await to test functions that return a promise.

```
test('fetches data successfully', async () => {
  const data = await fetchData();
  expect(data).toBeDefined();
});
```

Testing in React

Jest snapshot testing

- A testing utility to render React components and interact with them for testing.
- Helps simulate user behavior with a focus on how the component is used, not just how it looks.
- **How to Render a Component for Testing?** Using React Testing Library’s render method

```
import { render } from '@testing-library/react';
render(<MyComponent />);
```

Testing in React

Jest snapshot testing

- **How to create and test Snapshots?** Using React Testing Library’s render method

- **Snapshots** store the rendered output of a component and compare it with the next render to catch unintended changes.

```
test('matches the snapshot', () => {
  const { asFragment } = render(<MyComponent />);
  expect(asFragment()).toMatchSnapshot();
});
```

DOM Interaction in React/Redux Tests

- **Selecting and Verifying DOM Elements:** Use getByText, getByRole, or getByTestId to target elements in the DOM.

```
const button = screen.getByText('Submit');
expect(button).toBeInTheDocument();
```

- **Firing Events on DOM Elements:** Simulate user actions like clicks or typing using fireEvent.

```
fireEvent.click(button);
```

- **Testing React Components with Redux:** Mock Redux state or use Provider from react-redux to wrap components and test their interaction with the Redux store.

```
import { Provider } from 'react-redux';
import { store } from './store';

render(
  <Provider store={store}>
    <MyComponent />
  </Provider>
);
```

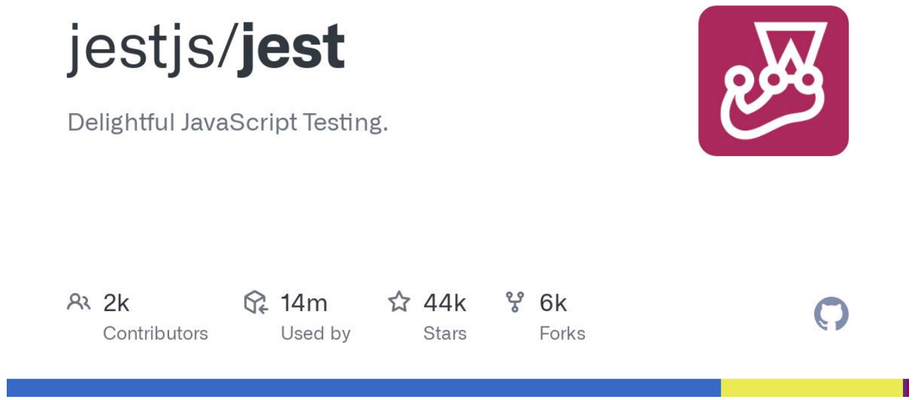
Key Takeaways

- **Why is Testing Important?**
  - *Prevents regression, ensures reliability, and promotes clean code.*

- **Jest for Unit and Integration Testing:**
  - **Unit Tests:** *Test small pieces of code like functions.*
  - **Integration Tests:** *Test components and their interactions.*

- **React Testing Library for DOM Interactions:**
  - *Useful for simulating real-world user actions and verifying the result.*

- **Snapshot Testing:**
  - *Ensure the UI does not change unexpectedly.*

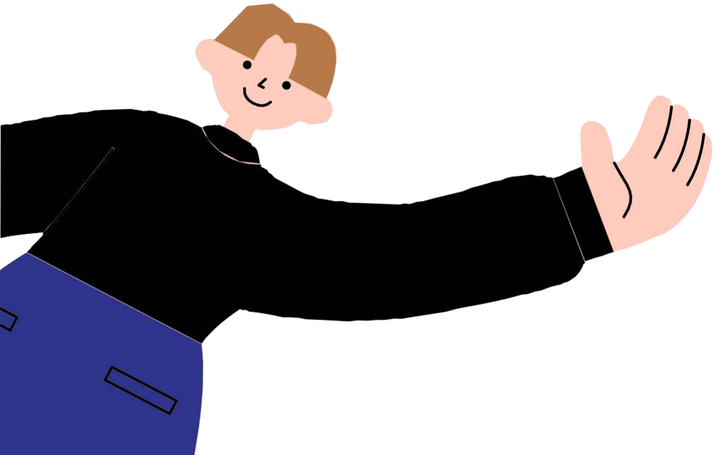


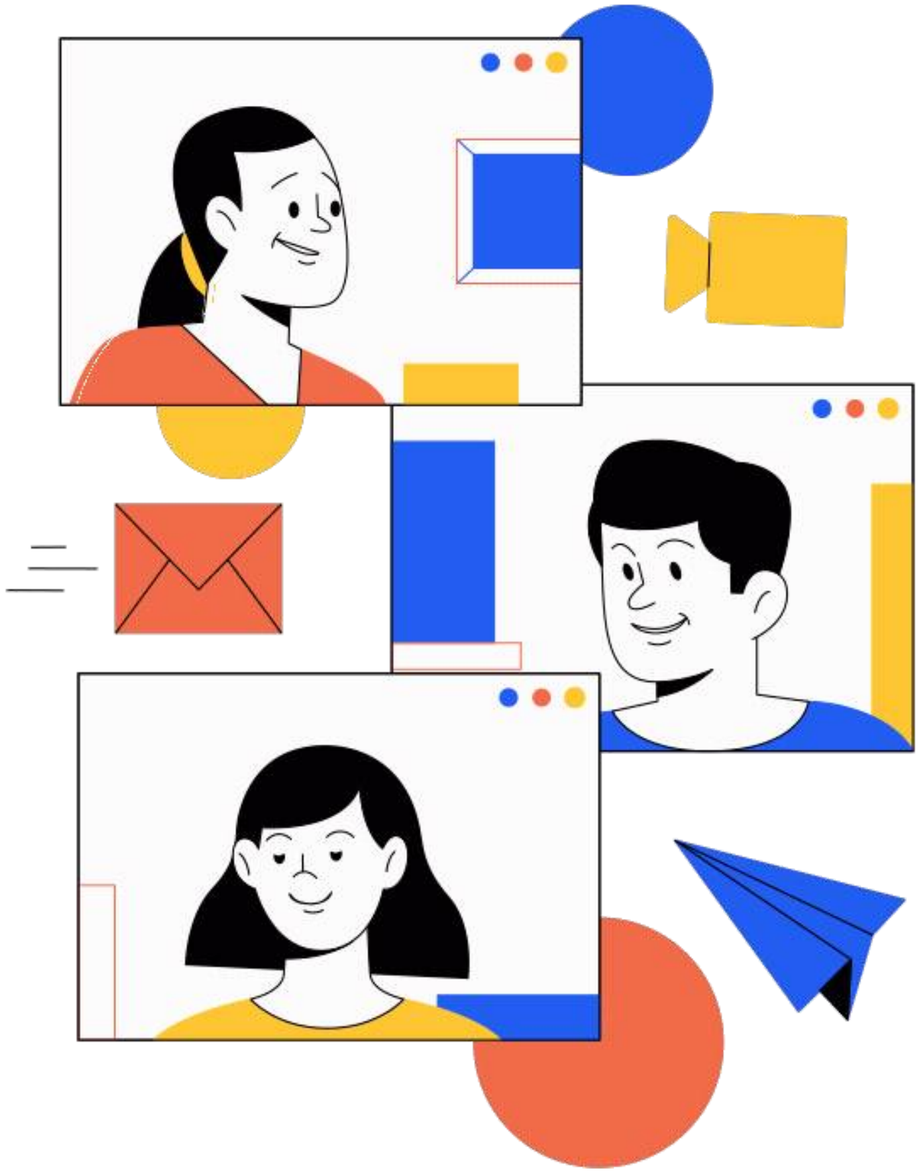
Its Demo Time

Next Session:

Hands on

Any Questions?





Thank you for attending!

Feel free to email at **a.lotfy@fci-cu.edu.eg** or reach me at circle anytime for any questions or clarifications!

Follow me on Github **@ahmeddxfouad**

code and slides are found at this **github repo**