React Testing

🍆🌽🥒 “Eat your veggies” 🍆🌽🥒

Overview

- What is Testing
- Why we need to Test
- Types of Tests
    - Unit Testing
    - Component Testing
    - Snapshot Testing
    - Visual Regression Testing
    - Integration Testing
    - E2E Testing
- What is Jest
- What is @testing-library
- What is TDD
- TDD while component testing a basic UI + Pair Programming
- Q & A with special guest Jenson aka Jestlord

Before we begin

Clone our practice repo!

https://github.com/scrambledeggs/booky-tdd

Run `yarn install` afterwards to install packages

What is Testing

Testing is the process of executing a program with the aim of finding errors.

Why we need to Test

- Ensures that nothing breaks when we make changes to our codebase
- Solidifies any business rules or requirements that were initially discussed by our stakeholders
- Allows us to embrace more errors with an open heart
- Self-documenting and easier to read for newly onboarded developers
- Enables us to design our code better because of implied decoupling through mocks and stubs
- Makes us a better developer in the long run

Types of Testing

- Unit Testing
- Component Testing
- Snapshot Testing
- Visual Regression Testing
- Integration Testing
- E2E Testing

Unit Testing

```
// add.ts
const add = (x, y): number => {
  return x + y
}

export default add
```

```
// add.test.ts
import add from 'add'

it('should return sum of x and y', () => {
  const sum = add(1, 2)
  expect(sum).toBe(3)
})
```

Component Testing

```
// CustomButton.tsx
const CustomButton = ( => {
  return (
    <button>
      <p>Push Me</p>
    </button>
 )
}

export default CustomButton
```

```
// CustomButton.test.tsx
import { render, screen } from '@testing-library/react'
import CustomButton from './CustomButton'

it('should render Button', () => {
  render(<CustomButton />)
  const button = screen.getByRole(‘button’)
  expect(button).toBeDefined()
})

it('should fire a function when clicked’, () => {
  ...
  ...
})
```

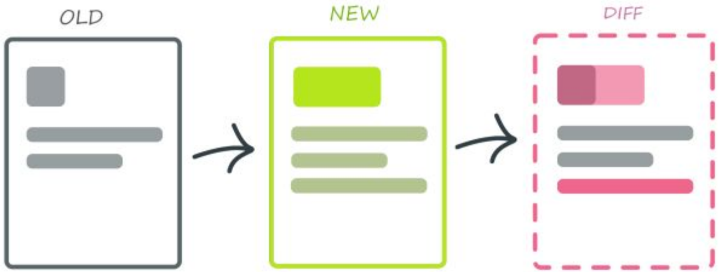
Snapshot Testing

```
// CustomButton.tsx
const CustomButton = ( => {
  return (
    <button>
      <p>Push Me</p>
    </button>
 )
}

export default CustomButton
```

```
// CustomButton.test.tsx
import { render, screen } from '@testing-library/react'
import CustomButton from './CustomButton'

it('should follow original design specs', () => {
  render(<CustomButton />)
  const button = screen.getByRole(‘button’)
  expect(button).toMatchSnapshot()
})
```

Visual Regression Testing

- Stores screenshots instead of DOM elements
- Useful for testing UI responsiveness (Mobile vs Web)
- Requires larger storage



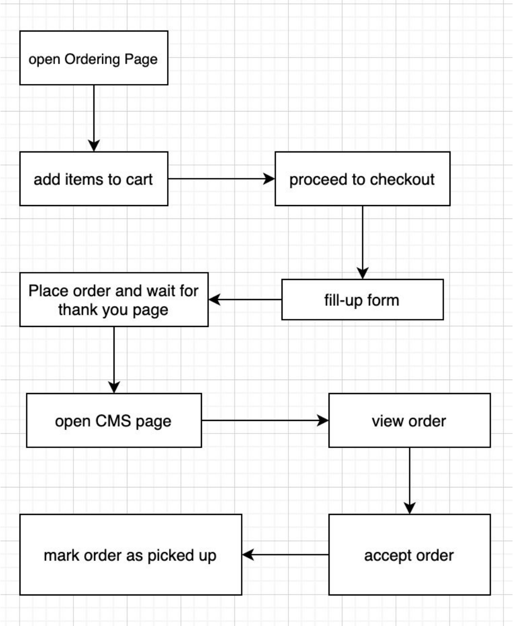
Integration Testing

```
// HomePage.tsx
import CustomButton from ‘./CustomButton’
const Home = ( => {
  return (
    <div>
      <p>Home</p>
      <input type=”text” />
      <CustomButton />
    </div>
 )
}

export default Home
```

```
// HomePage.test.tsx
import { render, screen } from '@testing-library/react'
import userEvent from ‘@testing-library/user-event
import HomePage from './HomePage

it('should update title when button is pressed', async () => {
  render(<HomePage />)
  const button = screen.getByRole(‘button’)

  userEvent.click(button)

  const title = await screen.findByText(‘New Title’)
  expect(title).toBeDefined()
})
```

End-to-End Testing

- Code-agnostic
- Can be run separately from codebase
- Playwright, Cypress, Selenium, Puppeteer
- Can be inconsistent sometimes



What is Jest

- Jest is a **test runner**
- Executes test files from `__tests__` or `*.test.js/ts/jsx/tsx`
- `describe(), test(), it(), expect(), beforeAll(), beforeEach`(,)etc

What is `@testing-library`

- family of packages that helps you test UI components in a user-centric way

Arrange

`@testing-library/react`

- ❑ Render react components
- ❑ Find specific DOM components after rendering (button, input, text, etc)

Assert

`@testing-library/jest-dom`

- ❑ Is text visible to user?
- ❑ Is checkbox checked?
- ❑ Is input field focused?

Act

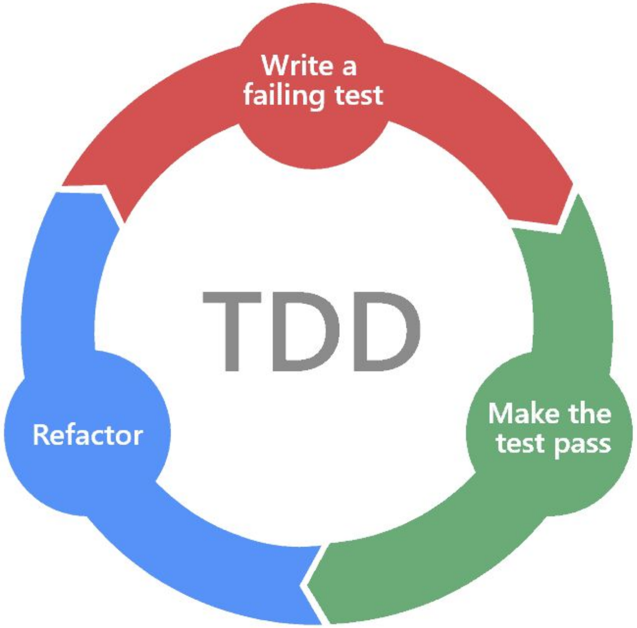
`@testing-library/user-event`

- ❑ Pressing a button
- ❑ Typing on input field
- ❑ Clearing an input field
- ❑ Hovering a UI, etc

What is TDD

Test-Driven Development

Red, Green, Refactor



Coding Time

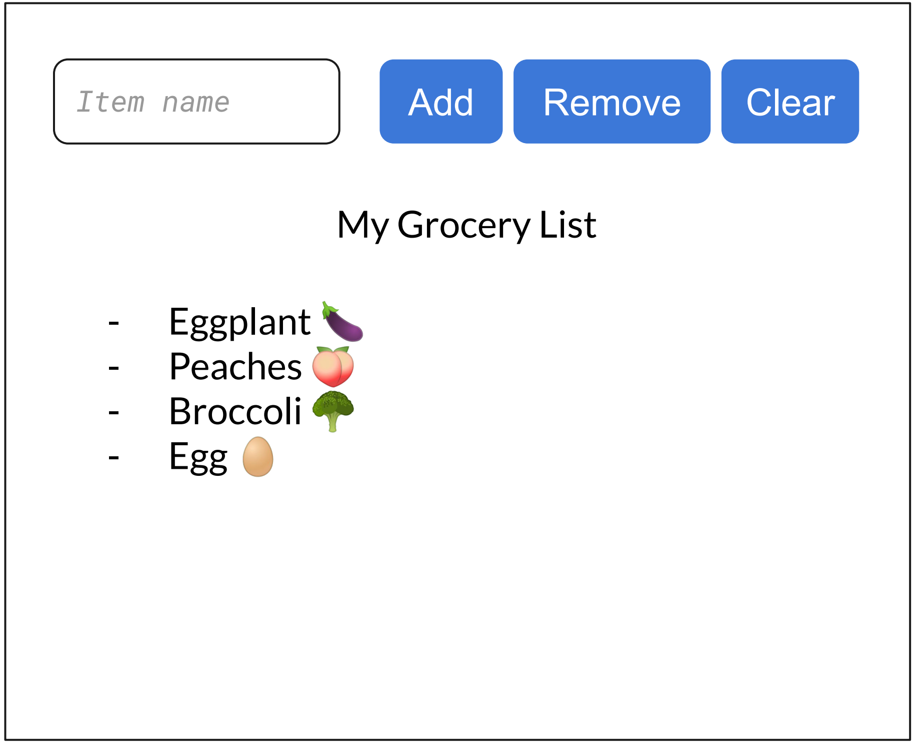
Let’s practice TDD 🤓

Make sure you have cloned our sample app and have executed `yarn install`

https://github.com/scrambledeggs/booky-tdd

What are we coding?

- Let’s create a Grocery List App



My Grocery List

- Eggplant 🍆
- Peaches 🍑
- Broccoli 🥦
- Egg 🥚

Q&A with Jenson

Don’t be shy

fin.

I hope you learned something new today!

Reference

What is testing
https://www.geeksforgeeks.org/types-software-testing/

Visual Regression testing
https://medium.com/loftbr/visual-regression-testing-eb74050f3366

What is @testing-library
https://testing-library.com/

What is Jest
https://jestjs.io/docs/getting-started