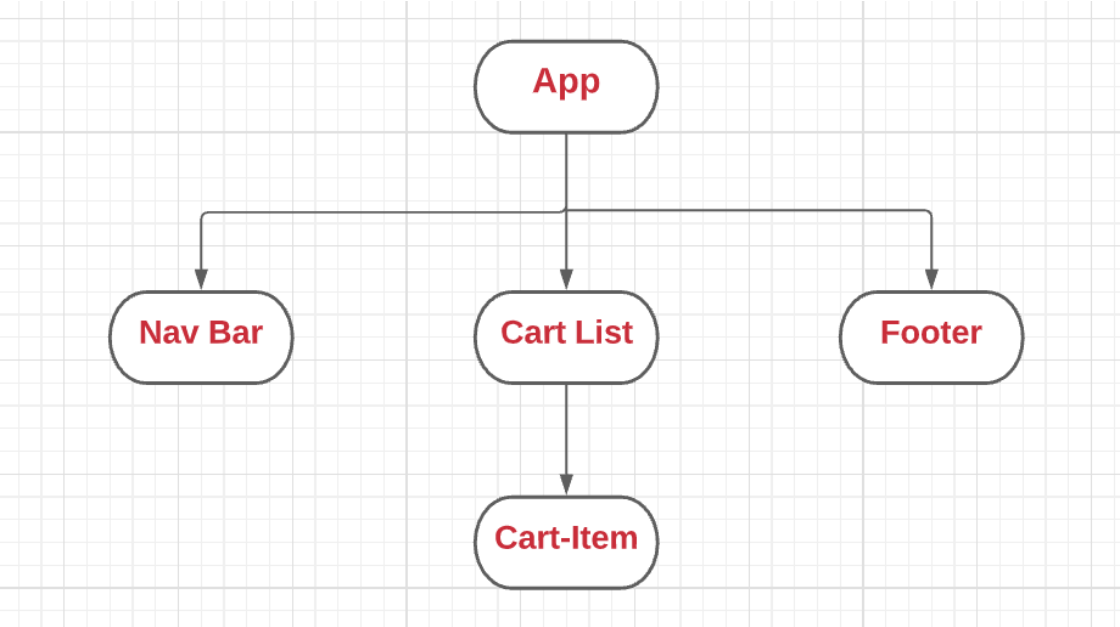
Mini Project: Starting The Project

Intro To Cart

In this project, we will create a cart app like amazon where we will have the following components:



CartItem:First Component

Let's understand the code for the first component in detail. Here is the code snippet for your reference:

```
import React from 'react';

class CartItem extends React.Component {
  render () {
    return (
      <div className="cart-item">
        <div className="left-block">
          <img/>
        </div>
        <div className="right-block">
          <div>Phone</div>
          <div>Rs 999</div>
          <div>Qty: 1</div>
          <div className="cart-item-actions">
            {/* Buttons */}
          </div>
        </div>
      </div>
    );
  }
}
export default CartItem;
```

Note: Here, we created a class component

Render() - Render function is a part of the component lifecycle that converts a react component to HTML code to be viewed by the browser. It creates a view that is to be rendered on the browser.

React.Component- React.Component is a library that provides different functions to the component class.

Export -. In React, we use the keyword export to export a particular module or a named parameter, or a combination.

Styling Of CartItem

After writing the basic code, we styled the cart item, but regular CSS can't be used to style JSX file, so we created javascript functions to apply styles on CartItem. Here is the code snippet for your reference:

```
class CartItem extends React.Component {
  render () {
    return (
      <div className="cart-item">
        <div className="left-block">
          <img style={styles.image} />
        </div>
        <div className="right-block">
          <div style={ { fontSize: 25 } }>Phone</div>
          <div style={ { color: '#777' } }>Rs 999</div>
          <div style={ { color: '#777' } }>Qty: 1</div>
          <div className="cart-item-actions">
            {/* Buttons */}
          </div>
        </div>
      </div>
    );
  }
}

const styles = {
  image: {
    height: 110,
    width: 110,
    borderRadius: 4,
    background: '#ccc'
  }
}
```

Here we created constant named styles in which we mentioned styles for our image and used this javascript variable in image div. As we know {} are used to render javascript code in JSX file so we used style={styles.image}.

Adding State to CartItem

What is State?

A state is a way to store your local data for that particular component. A state is a plain javascript object. To define the state of a component, we use a constructor.

Here is a code snippet to get more details about the state:

```
class CartItem extends React.Component {
  constructor(){
      super();
      this.state={
          price:999,
          title: 'Mobile-Phone',
          qty:1,
          img:''
      }
  }
  render () {
    const {price,title,qty}=this.state;
    return (
      <div className="cart-item">
        <div className="left-block">
          <img style={styles.image} />
        </div>
        <div className="right-block">
          <div style={ { fontSize: 25 } }>{title}</div>
          <div style={ { color: '#777' } }>Rs {price}</div>
          <div style={ { color: '#777' } }>Qty: {qty}</div>
```

As mentioned, the constructor is used to store state of a component, so we saved data of cart-item using the former.

super() - Super is a constructor of parent class that is CartItem which is under React.Component Library. We call super class in state constructor because we are inheriting state constructor in our parent constructor.

Object Destructuring - Destructuring is a JavaScript expression that allows us to extract data from arrays, objects, and maps and set them into new, distinct variables. Destructuring will enable us to extract multiple properties, or items, from an array at a time.

We extracted the price title and quantity values from the state object and used them in the div component.

Increasing Quantity

Our target was to create a button with an onClick event to increase or decrease the cart item's quantity. So here we are demonstrating the increaseQuantity function and some technical terms related to it. Here is the code snippet for your reference:

```
class CartItem extends React.Component {
  constructor(){
      super();
      this.state={
          price:999,
          title: 'Mobile-Phone',
          qty:1,
          img:''

      }
      this.increaseQuantity=this.increaseQuantity.bind(this);

  }
increaseQuantity() {
    console.log('this',this.state)
}
```

```
<img
    alt=" increase"
    className="action-icons"
    onClick={this.increaseQuantity}
    src="https://as2.ftcdn.net/jpg/02/22/71/79/500_F_222717975_8TfDJLKSAjUmukqhJFcfrhGaNP9xaeP
```

You might be wondering about the bind function that is used in the constructor. An explanation to this is:
When we click on the increase button, the click event works and calls **increaseQuantity** function, which says to console this.state. But when our internal code searches for the value of the keyword **'this'**, it cannot find that as it is lost. So it shows an error. To solve this problem, the bind() function comes in.

Bind() - The **bind()** method creates a new function that, when called, has **this** keyword set to the provided value, with a given sequence of arguments preceding any provided when the new function is called.
Bind() function remembers the value of “this” keyword and renders the data from the object.

Summarising It

Let’s summarise what we have learnt in this module:

- Learned about creating cart-item.
- Styled the cart item and learned about the state object.
- Learned about some technical terms like object destructuring, bind() and render().

Some References:

- More information about bind() function : https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_objects/Function/bind
- More information about object destructuring: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Destructuring_assignment