

# *Front End*

*Class 10*

*September 15, 2021*

# Agenda

- Studio walk through
- Kahoot
- Code walkthrough with slides
- Studio

# Time Management

Message from Jen Morgan  
Presenting info on time management.

Saturday 9/18 office hours, 3:30 - 4:30

<https://us02web.zoom.us/j/86248640518?pwd=aUYxNIM5Yk5zRDUyOGNvVIBnVURJUT09>

#resources channel will have the post presentation material

# useState() vs useReducer()

Generally, you'll know when you need useReducer() (→ when using useState() becomes cumbersome or you're getting a lot of bugs/ unintended behaviors)

## useState()

The main state management “tool”

Great for independent pieces of state/ data

Great if state updates are easy and limited to a few kinds of updates

## useReducer()

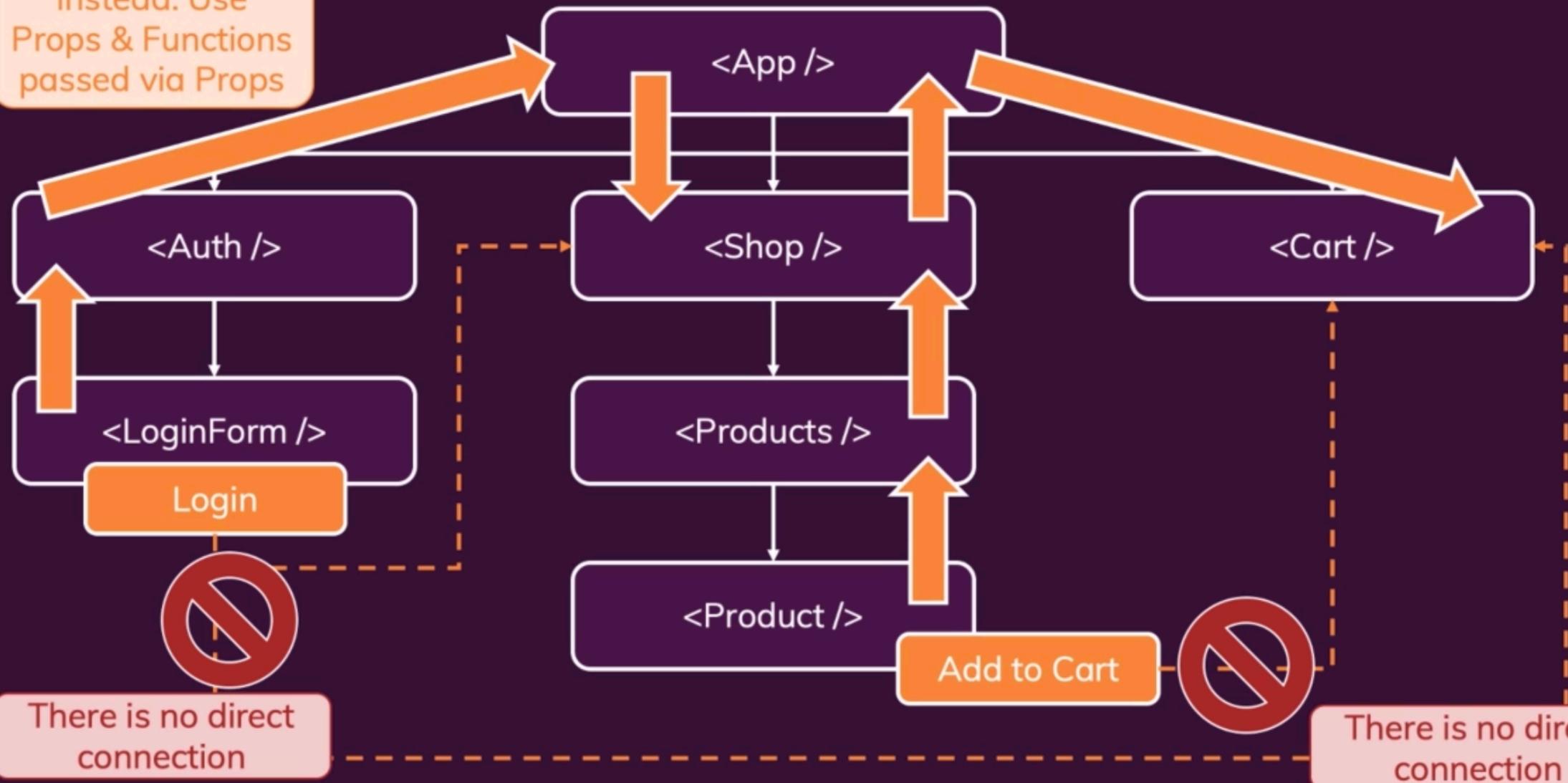
Great if you need “more power”

Should be considered if you have related pieces of state/ data

Can be helpful if you have more complex state updates

# Component Trees & Component Dependencies

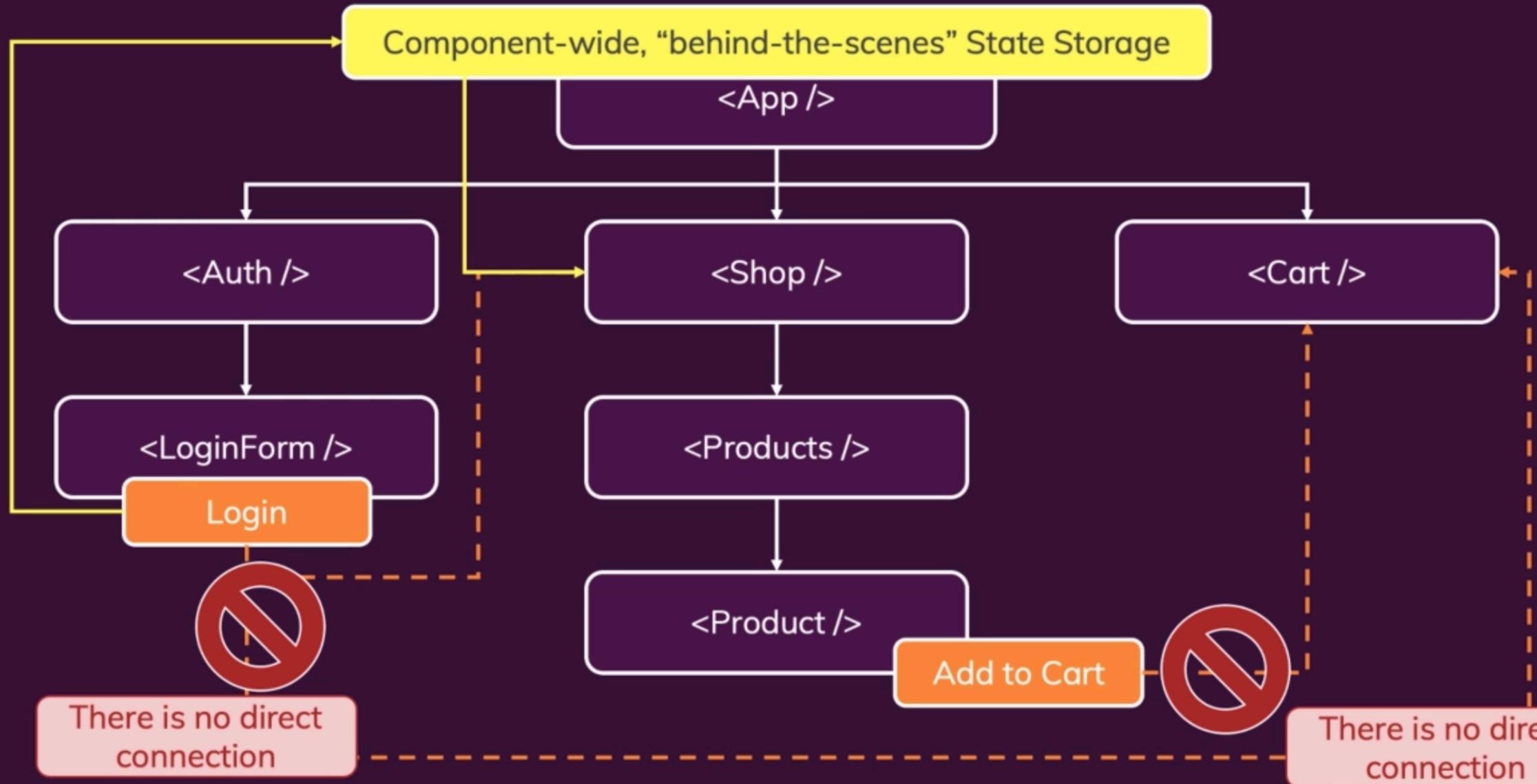
Instead: Use  
Props & Functions  
passed via Props



There is no direct  
connection

There is no direct  
connection

# Context to the Rescue!



```
import React from 'react';

const AuthContext = React.createContext({
  isLoggedIn: false,
});

export default AuthContext;
```

```
You, 4 minutes ago | Author (You)
import React, { useState, useEffect } from 'react';

import Login from './components/Login/Login';
import Home from './components/Home/Home';
import MainHeader from './components/MainHeader/MainHeader';
import AuthContext from './store/auth-context';

function App() {
  const [isLoggedIn, setIsLoggedIn] = useState(false);

  //run this code only once when the app starts up.
  useEffect(() => {
    const storedUserLoggedInInformation = localStorage.getItem('isLoggedIn');
    if (storedUserLoggedInInformation === '1') {
      setIsLoggedIn(true);
    }
  }, []);

  const loginHandler = (email, password) => {
    localStorage.setItem('isLoggedIn', '1');
    setIsLoggedIn(true);
  };

  const logoutHandler = () => {
    localStorage.removeItem('isLoggedIn');
    setIsLoggedIn(false);
  };

  return (
    <AuthContext.Provider>
      <MainHeader isAuthenticated={isLoggedIn} onLogout={logoutHandler} />
      <main>
        {!isLoggedIn && <Login onLogin={loginHandler} />}
        {isLoggedIn && <Home onLogout={logoutHandler} />}
      </main>
    </AuthContext.Provider>
  );
}

export default App;
```

You, 4 minutes ago • Uncommitted changes

```
return (
  <AuthContext.Provider
    value={{
      isLoggedIn: false,
    }}>
    You, seconds ago • Uncommitted changes
  >
    <MainHeader isAuthenticated={isLoggedIn} onLogout={logoutHandler} />
    <main>
      {!isLoggedIn && <Login onLogin={loginHandler} />}
      {isLoggedIn && <Home onLogout={logoutHandler} />}
    </main>
  </AuthContext.Provider>
);
}
```

```
import React from 'react';
import AuthContext from '../../../../../store/auth-context';

import classes from './Navigation.module.css';

const Navigation = (props) => {
  return (
    <AuthContext.Consumer>
      {(ctx) => {
        return (
          <nav className={classes.nav}>
            <ul>
              {ctx.isLoggedIn && (
                <li>
                  <a href="/">Users</a>
                </li>
              )}
              {ctx.isLoggedIn && (
                <li>
                  <a href="/">Admin</a>
                </li>
              )}
              {ctx.isLoggedIn && (
                You, 13 minutes ago • Uncommitted changes
                <li>
                  <button onClick={props.onLogout}>Logout</button>
                </li>
              )}
            </ul>
          </nav>
        );
      }}
    </AuthContext.Consumer>
  );
}
```

## A Typical Page

**Welcome back!**

```
return (
  <AuthContext.Provider
    value={[
      isLoggedIn: isLoggedIn,      You, seconds ago • Uncommitted changes
    ]}
  >
  <MainHeader isAuthenticated={isLoggedIn} onLogout={logoutHandler} />
  <main>
    {!isLoggedIn && <Logi (JSX attribute) onLogout: () => void
    {isLoggedIn && <Home onLogout={logoutHandler} />}
  </main>
</AuthContext.Provider>
);
}
```

```
import React from 'react';

import Navigation from './Navigation';
import classes from './MainHeader.module.css';

const MainHeader = (props) => {
  return (
    <header className={classes['main-header']}>
      <h1>A Typical Page</h1>
      <Navigation onLogout={props.onLogout} />
    </header>
  );
};

export default MainHeader;
```

```
return (
  <AuthContext.Provider
    value={{|
      isLoggedIn: isLoggedIn,
    }|}
  >
  <MainHeader onLogout={logoutHandler} /> You, a minute ago • Uncommitt
  <main>
    {!isLoggedIn && <Login onLogin={loginHandler}> />}
    {isLoggedIn && <Home onLogout={logoutHandler}> />}
  </main>
</AuthContext.Provider>
);
```

```
import React from 'react';

import Navigation from './Navigation';
import classes from './MainHeader.module.css';

const MainHeader = (props) => {
  return (
    <header className={classes['main-header']}>
      <h1>A Typical Page</h1>
      <Navigation onLogout={props.onLogout}> />
    </header>
  );
};

export default MainHeader;
```

## useContext hook

```
use, a minute ago · Reviewer (you)
import React, { useContext } from 'react';
import AuthContext from '../../store/auth-context';

import classes from './Navigation.module.css';

const Navigation = (props) => {
  const ctx = useContext(AuthContext);      You, seconds ago • Uncommitted change
  return (
    <nav className={classes.nav}>
      <ul>
        {ctx.isLoggedIn && (
          <li>
            <a href="/">Users</a>
          </li>
        )}
        {ctx.isLoggedIn && (
          <li>
            <a href="/">Admin</a>
          </li>
        )}
        {ctx.isLoggedIn && (
          <li>
            <button onClick={props.onLogout}>Logout</button>
          </li>
        )}
      </ul>
    </nav>
  );
}

export default Navigation;
```

Logout prop – handling logout.

Can Pass functions to other components. – Any component listening to the AuthContext can listen to the onLogout.

```
return (
  <AuthContext.Provider
    value={{
      isLoggedIn: isLoggedIn,
      onLogout: logoutHandler
    }}
  >
    <MainHeader onLogout={logoutHandler} />
    <main>
      {!isLoggedIn && <Login onLogin={loginHandler} />}
      {isLoggedIn && <Home onLogout={logoutHandler} />}
    </main>
  </AuthContext.Provider>
);
```

Logout prop – handling logout.  
Can Pass functions to other  
components. – Any component  
listening to the AuthContext can  
listen to the onLogout.

```
import React, { useContext } from 'react';
import AuthContext from '../../../../../store/auth-context';

import classes from './Navigation.module.css';

const Navigation = (props) => {
  const ctx = useContext(AuthContext);
  return (
    <nav className={classes.nav}>
      <ul>
        {ctx.isLoggedIn && (
          <li>
            <a href="/">Users</a>
          </li>
        )}
        {ctx.isLoggedIn && (
          <li>
            <a href="/">Admin</a>
          </li>
        )}
        {ctx.isLoggedIn && (
          <li>
            <button onClick={ctx.onLogout}>Logout</button>
          </li>
        )}
      </ul> You, 4 days ago • added
    </nav>
  );
};

export default Navigation;
```

Logout prop – handling logout.  
Can Pass functions to other  
components. – Any component  
listening to the AuthContext can  
listen to the onLogout.

E-Mail

abc@gmail.com

Password

.....

Login

## A Typical Page

Users Admin Logout

Welcome back!

## A Typical Page

E-Mail

Password

Login

Logout prop – handling logout.  
Can Pass functions to other  
components. – Any component  
listening to the AuthContext can  
listen to the onLogout.

Don't need props in navigation  
component anymore.

```
import React, { useContext } from 'react';
import AuthContext from '../store/auth-context';

import classes from './Navigation.module.css';

const Navigation = () => {
  const ctx = useContext(AuthContext);
  return (
    <nav className={classes.nav}>
      <ul>
        {ctx.isLoggedIn && (
          <li>
            <a href="/">Users</a>
          </li>
        )}
        {ctx.isLoggedIn && (
          <li>
            <a href="/">Admin</a>
          </li>
        )}
        {ctx.isLoggedIn && (
          <li>
            <button onClick={ctx.onLogout}>Logout</button>
          </li>
        )}
      </ul>
    </nav>
  );
}

export default Navigation;
```

Can pull more logic out of the App component into separate Context Management component.

```
import React, { useState } from 'react';

const AuthContext = React.createContext({
  isLoggedIn: false,
  onLogout: () => {},
  onLogin: (email, password) => {},
});

export const AuthContextProvider = (props) => {
  const [isLoggedIn, setIsLoggedIn] = useState(false);
  const logoutHandler = () => {
    setIsLoggedIn(false);
  };
  const loginHandler = () => {
    setIsLoggedIn(true);
  };
  <AuthContext.Provider
    value={{
      isLoggedIn: isLoggedIn,
      onLogout: logoutHandler,
      onLogin: loginHandler,
    }}
  >
    {props.children}
  </AuthContext.Provider>;
};

export default AuthContext;
```

## Move additional state logic out of App component

```
import React, { useState, useEffect } from 'react';

const AuthContext = React.createContext({
  isLoggedIn: false,
  onLogout: () => {},
  onLogin: (email, password) => {},
});

export const AuthContextProvider = (props) => [
  const [isLoggedIn, setIsLoggedIn] = useState(false);

  //run this code only once when the app starts up.
  useEffect(() => {
    const storedUserLoggedInInformation = localStorage.getItem('isLoggedIn');
    if (storedUserLoggedInInformation === '1') {
      setIsLoggedIn(true);
    }
  }, []);
  const logoutHandler = () => {
    localStorage.removeItem('isLoggedIn');
    setIsLoggedIn(false);
  };

  const loginHandler = () => {
    localStorage.setItem('isLoggedIn', '1');
    setIsLoggedIn(true);
  };
  <AuthContext.Provider
    value={{
      isLoggedIn: isLoggedIn,
      onLogout: logoutHandler,
      onLogin: loginHandler,
    }}
  >
    {props.children}
  </AuthContext.Provider>;
];

export default AuthContext;
```

## Can cleanup App component

```
You, seconds ago | 1 author (You)
import React from 'react';

import Login from './components/Login/Login';
import Home from './components/Home/Home';
import MainHeader from './components/MainHeader/MainHeader';      You, 4

function App() {
  return (
    <>
      <MainHeader onLogout={logoutHandler} />
      <main>
        {!isLoggedIn && <Login onLogin={loginHandler} />}
        {isLoggedIn && <Home onLogout={logoutHandler} />}
      </main>
    </>
  );
}

export default App;
```

Change index.js to wrap App component inside AuthContextProviders

```
You, seconds ago | Author (You)
import React from 'react';
import ReactDOM from 'react-dom';

import './index.css';
import App from './App';
import { AuthContextProvider } from './store/auth-context';

ReactDOM.render(
  <AuthContextProvider>
    <App />
  </AuthContextProvider>,      You, seconds ago • Uncommitted change
  document.getElementById('root')
);
```

Structure is leaner and more focused  
on App Context Management

```
import React from 'react';

import Login from './components/Login/Login';
import Home from './components/Home/Home';
import MainHeader from './components/MainHeader/MainHeader';
import { useContext } from 'react/cjs/react.development';
import AuthContext from './store/auth-context';

function App() {
  const ctx = useContext(AuthContext);
  return (
    <>
      <MainHeader /> You, seconds ago • Uncommitted changes
      <main>
        {!ctx.isLoggedIn && <Login />}
        {ctx.isLoggedIn && <Home />}
      </main>
    </>
  );
}

export default App;
```

Can add authCtx in different files  
instead of using state and passing  
props. Login.js

```
const authCtx = useContext(AuthContext);
```

```
const submitHandler = (event) => {
  event.preventDefault();
  authCtx.onLogin(emailState.value, pwdState.value);
};
```

**E-Mail**

abc@abc.com

**Password**

.....|

Login

## A Typical Page

Users

Admin

Logout

Welcome back!

# A Typical Page

The image shows a simplified representation of a login page. It features a white rectangular box with rounded corners and a thin gray border. Inside, there are two input fields: one labeled "E-Mail" and another labeled "Password", both represented by light red rectangles with red outlines. Below these fields is a gray rounded rectangle containing the word "Login".

E-Mail

Password

Login

# Concepts

Use props for configuration

Use Ctx for State Management across Components or across App.



## Context Limitations

React Context is **NOT optimized** for high frequency changes!

We'll explore a better tool for that, later

React Context also **shouldn't be used to replace ALL** component communications and props

Component should still be configurable via props and short "prop chains" might not need any replacement

# Rules of Hooks

Only call React Hooks in **React Functions**

React Component Functions

Custom Hooks  
(covered later!)

Only call React Hooks at the **Top Level**

Don't call them  
in nested  
functions

Don't call them  
in any block  
statements

+ extra, unofficial Rule for **useEffect()**: ALWAYS add everything you refer to inside of useEffect() as a dependency!

```
import React from 'react';
import classes from './Input.module.css';

const Input = (props) => {
  return (
    <div
      className={`${classes.control} ${props.isValid === false ? classes.invalid : ''}`}
    >
      <label htmlFor={props.id}>{props.label}</label>
      <input
        type={props.type}
        id={props.id}
        value={props.value}
        onChange={props.onChange}
        onBlur={props.onBlur}
      />
    </div>
  );
};

export default Input;
```

```
return (
  <Card className={classes.login}>
    <form onSubmit={submitHandler}>
      <Input
        id="email"
        type="email"
        label="E-mail"
        isValid={emailState.isValid}
        value={emailState.value}
        onChange={emailChangeHandler}
        onBlur={validateEmailHandler} You, a minute ag
      />
```

```
return (
  <Card className={classes.login}>
    <form onSubmit={submitHandler}>
      <Input
        id="email"
        type="email"
        label="E-mail"
        isValid={emailState.isValid}
        value={emailState.value}
        onChange={emailChangeHandler}
        onBlur={validateEmailHandler}
      />
      <Input
        id="password"
        type="password"
        label="Password"
        isValid={pwdState.isValid}
        value={pwdState.value}
        onChange={passwordChangeHandler}
        onBlur={validatePasswordHandler}
      />
      You, seconds ago • Uncommitted changes
    <div className={classes.actions}>
      <Button type="submit" className={classes.btn} disabled={!formIsValid}>
        Login
      </Button>
    </div>
  </form>
</Card>
);
};

export default Login;
```

**E-Mail**

abc@abc.com

**Password**

.....

**Login**

# Welcome back!

E-Mail

Password

Login

React with Components imperatively. Calling a function inside a component.

Remove button disable.

```
    dispatchEmail({ type: 'INPUT_BLUR' });

const validatePasswordHandler = () => {
  dispatchPassword({ type: 'INPUT_BLUR' });
};

const submitHandler = (event) => {
  event.preventDefault();
  if (formIsValid) {
    authCtx.onLogin(emailState.value, passwordState.value);
  } else if (!emailIsValid) {
    emailInputRef.current.activate();
  } else {
  }
};
```

## React with Components imperatively. Calling a function inside a component.

```
components > UI > Input > js input.js > input
import React, { useRef, useImperativeHandle } from 'react';
import classes from './Input.module.css';

const Input = React.forwardRef((props, ref) => {
  const inputRef = useRef();

  const activate = () => {
    inputRef.current.focus();
  };

  useImperativeHandle(ref, () => {
    return {
      focus: activate,
    };
  });

  return (
    <div
      className={`${classes.control} ${props.isValid === false ? classes.invalid : ''}`}
    >
      <label htmlFor={props.id}>{props.label}</label>
      <input
        type={props.type}
        id={props.id}
        value={props.value}
        onChange={props.onChange}
        onBlur={props.onBlur}
      />
    </div>
  );
};

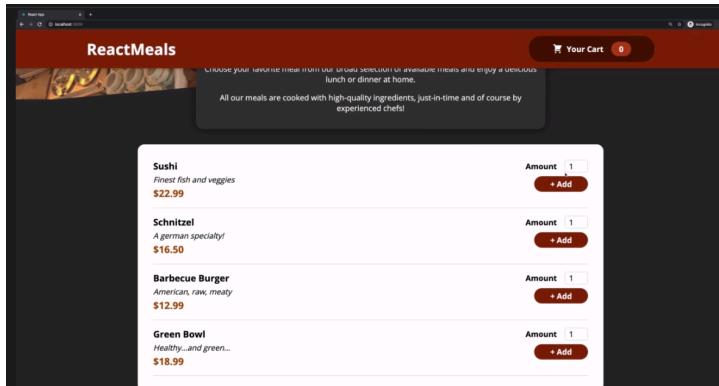
export default Input;
```

```
const emailInputRef = useRef();
const passwordInputRef = useRef();

return (
  <Card className={classes.login}>
    <form onSubmit={submitHandler}>
      <Input
        ref={emailInputRef}
        id="email"
        type="email"
        label="E-mail"
        isValid={emailState.isValid}
        value={emailState.value}
        onChange={emailChangeHandler}
        onBlur={validateEmailHandler}
      />
      <Input
        ref={passwordInputRef}
        id="password"
        type="password"
        label="Password"
        isValid={pwdState.isValid}
        value={pwdState.value}
        onChange={passwordChangeHandler}
        onBlur={validatePasswordHandler}
      />
      <div className={classes.actions}>
        <Button type="submit" className={classes.btn}>
          Login
        </Button>
      </div>
    </form>
  </Card>
);
};
```

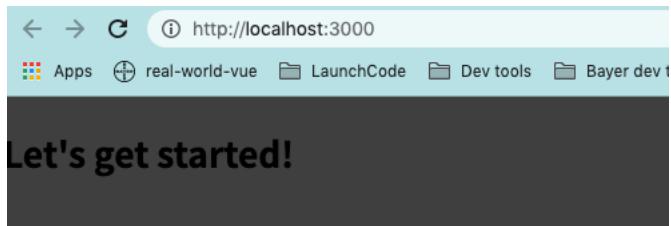
# Practice Project

- The food Order App.
- Practice concepts together. Add food to cart, view and modify cart, order.



# Practice Project

- Run 01-starting-project 5

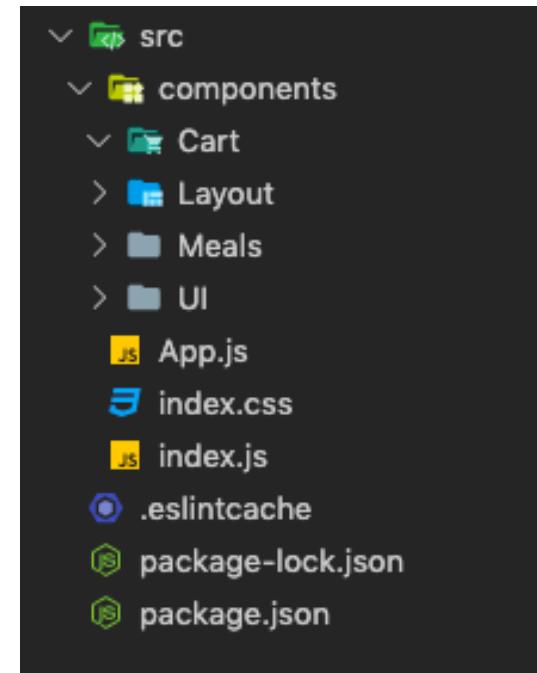


# Step 1: Identify Components

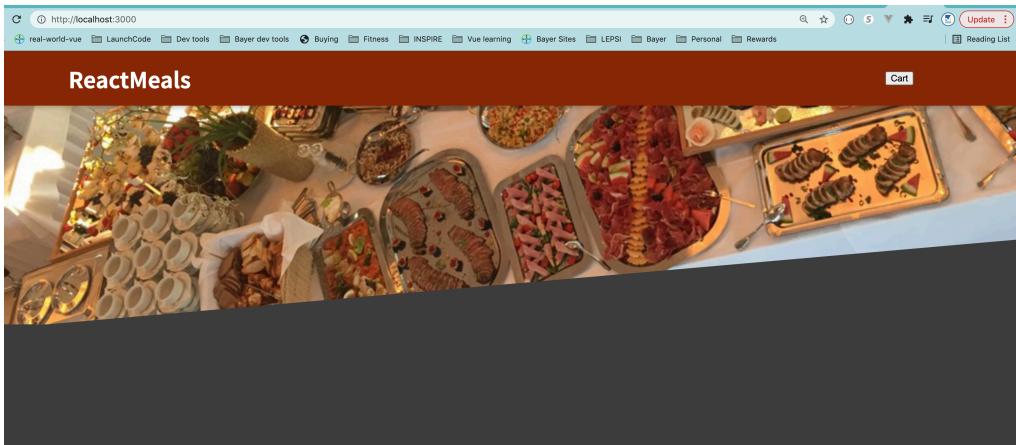
- Header
- Meals
- Cart

# Step 1: Organize Folder Structure

- UI – for ui elements. Input, Card, Modal Components. Not tied to specific feature but can be used in multiple places
- Layout – Header component
- Meals – List of Meal Components
- Cart – Cart related Components.



# Header Component



```
import Header from './components/Layout/Header';
function App() {
  return (
    <>
      <Header />
    </>
  );
}

export default App;
```

```
import classes from './Header.module.css';

import mealsImage from '../../assets/meals.jpg';

const Header = (props) => {
  return (
    <>
      <header className={classes.header}>
        <h1>ReactMeals</h1>
        <button>Cart</button>
      </header>
      <div className={classes['main-image']}>
        <img src={mealsImage} alt="A table full of delicious food" />
      </div>
    </>
  );
};

export default Header;
```

# Cart Button Component

```
import classes from './HeaderCartButton.module.css';
import CartIcon from '../Cart/CartIcon';
const HeaderCartButton = (props) => {
  return (
    <button className={classes.button}>
      <span className={classes.icon}>
        <CartIcon />
      </span>
      <span>Your Cart</span>
      <span className={classes.badge}>3</span>
    </button>
  );
};

export default HeaderCartButton;
```



# Meals Component

```
import MealsSummary from './MealsSummary';
import AvailableMeals from './AvailableMeals';
const Meals = () => {
  return (
    <div>
      <MealsSummary />
      <AvailableMeals />
    </div>
  );
}
export default Meals;
```

```
import classes from './MealsSummary.module.css';

const MealsSummary = () => {
  return (
    <section className={classes.summary}>
      <h2>Delicious Food, Delivered To You</h2>
      <p>
        Choose your favorite meal from our broad selection of available meals
        and enjoy a delicious lunch or dinner at home.
      </p>
      <p>
        All our meals are cooked with high-quality ingredients, just-in-time and
        of course by experienced chefs!
      </p>
    </section>
  );
}

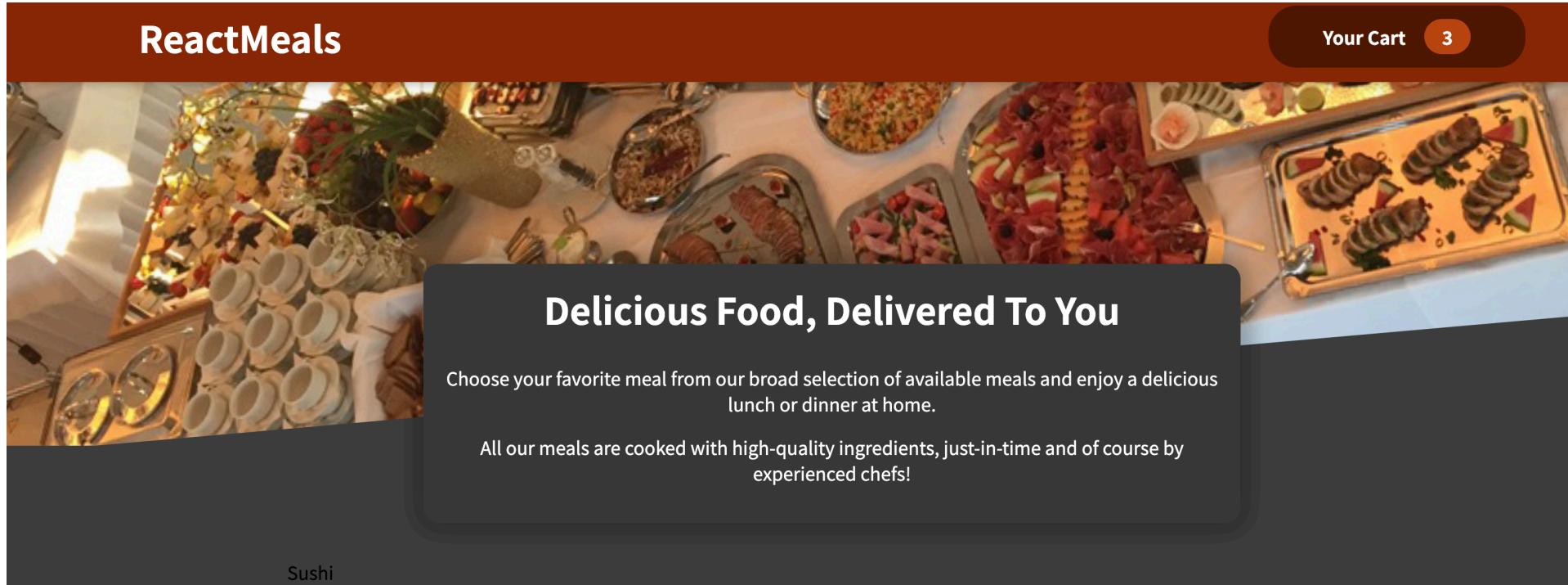
export default MealsSummary;
```

# Meals Component

```
import classes from './AvailableMeals.module.css';

const DUMMY_MEALS = [
  {
    id: 'm1',
    name: 'Sushi',
    description: 'Finest fish and veggies',
    price: 22.99,
  },
  {
    id: 'm2',
    name: 'Schnitzel',
    description: 'A german specialty!',
    price: 16.5,
  },
  {
    id: 'm3',
    name: 'Barbecue Burger',
    description: 'American, raw, meaty',
    price: 12.99,
  },
  {
    id: 'm4',
    name: 'Green Bowl',
    description: 'Healthy...and green...',
    price: 18.99,
  },
];
const AvailableMeals = () => {
  const mealsList = DUMMY_MEALS.map(meal => <li>{meal.name}</li>);
  return (
    <section className={classes.meals}>
      <ul>{mealsList}</ul>
    </section>
  );
};
export default AvailableMeals;
```

# Meals Component



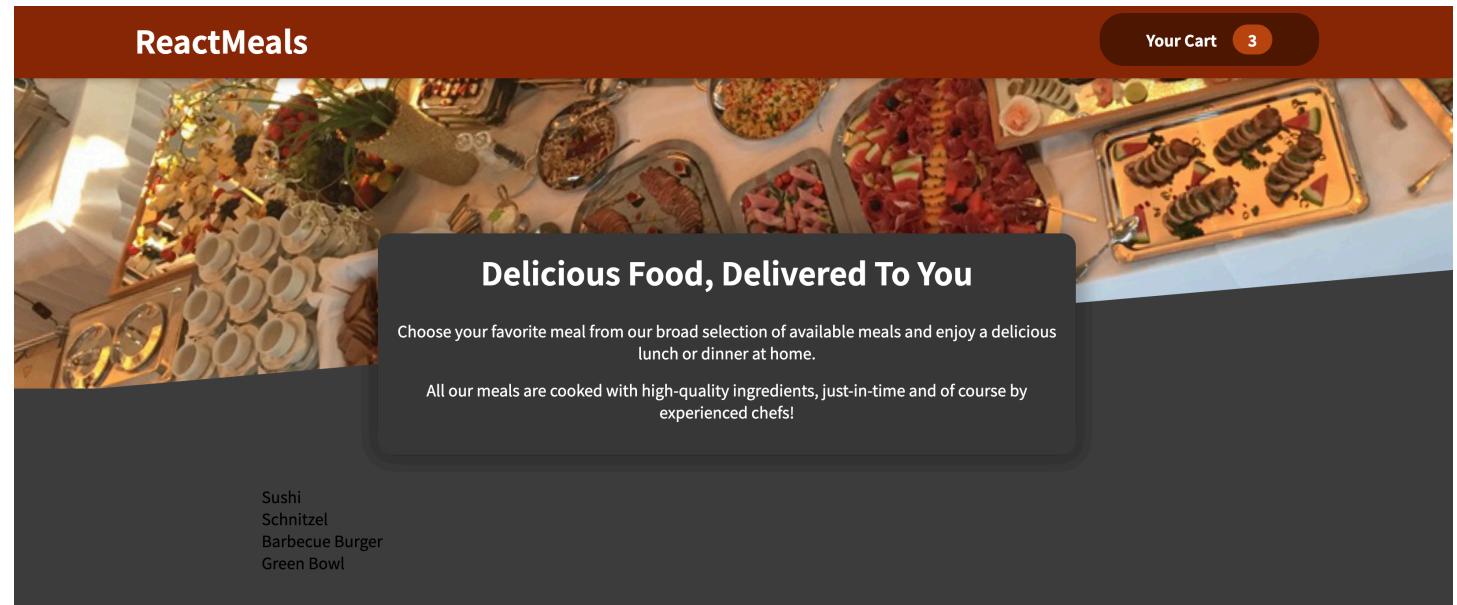
# Adding Meals Items and Displaying them

```
import classes from './Card.module.css';
const Card = (props) => {
  return <div className={classes.Card}>{props.children}</div>;
};

export default Card;
```

```
const AvailableMeals = () => {
  const mealsList = DUMMY_MEALS.map((meal) => <li>{meal.name}</li>);
  return (
    <section className={classes.meals}>
      <Card>
        <ul>{mealsList}</ul>
      </Card>
    </section>
  );
};

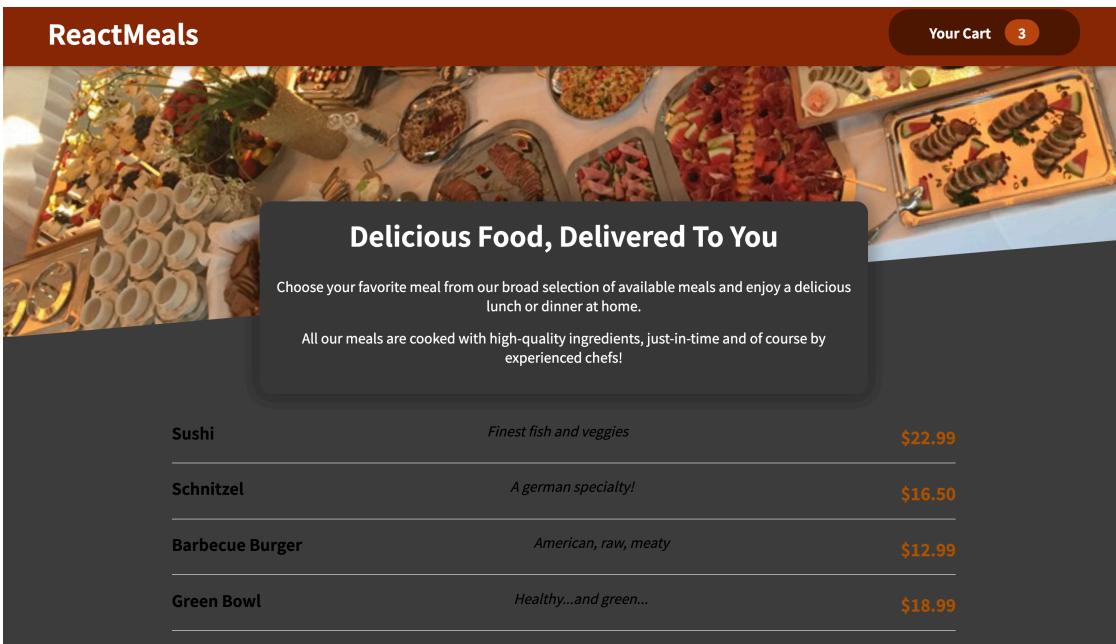
export default AvailableMeals;
```



# Adding Meals Items and Displaying them

```
import classes from './MealItem.module.css';
const MealItem = (props) => {
  const price = `$$ {props.price.toFixed(2)}`;
  return (
    <li>
      <div className={classes.meal}>
        <h3>{props.name}</h3>
        <div className={classes.description}>{props.description}</div>
        <div className={classes.price}>{price}</div>
      </div>
      <div></div>
    </li>
  );
}
export default MealItem;
```

```
const AvailableMeals = () => {
  const mealsList = DUMMY_MEALS.map(meal) => (
    <MealItem
      key={meal.id}
      name={meal.name}
      description={meal.description}
      price={meal.price}
    />
  );
  return (
    <section className={classes.meals}>
      <Card>
        <ul>{mealsList}</ul>
      </Card>
    </section>
  );
}
export default AvailableMeals;
```



# Adding form

```
import classes from './MealItemForm.module.css';
import Input from '../../../../../UI/Input';
const MealItemForm = (props) => {
  return (
    <form className={classes.form}>
      <Input
        label="Amount"
        input={{
          id: 'amount_' + props.id,
          type: 'number',
          min: '1',
          max: '5',
          step: '1',
          defaultValue: '1',
        }}
      />
      <button>+ Add</button>
    </form>
  );
};

export default MealItemForm;
```

```
import classes from './Input.module.css';
const Input = (props) => {
  return (
    <div className={classes.input}>
      <label htmlFor={props.input.id}>{props.label}</label>
      <input id={props.input.id} {...props.input} />
    </div>
  );
};

export default Input;
```

ReactMeals

Your Cart 3

**Delicious Food, Delivered To You**

Choose your favorite meal from our broad selection of available meals and enjoy a delicious lunch or dinner at home.

All our meals are cooked with high-quality ingredients, just-in-time and of course by experienced chefs!

Meal	Description	Price
Sushi	Finest fish and veggies	\$22.99
Schnitzel	A german specialty!	\$16.50
Barbecue Burger	American, raw, meaty	\$12.99

Amount 1    + Add

Amount 1    + Add

Amount 1    + Add

# Adding Modal

```
</head>
<body>
  <noscript>You need to enable JavaScript to run this app.</noscript>
  <div id="overlays"></div>
  <div id="root"></div>
```

```
import Modal from '../UI/Modal';
import classes from './Cart.module.css';
const Cart = (props) => {
  const cartItems = (
    <ul className={classes['cart-items']}>
      {[{ id: 'c1', name: 'Sushi', amount: 2, price: 12.99 }].map((item) => (
        <li>{item.name}</li>
      ))}
    </ul>
  );
  return (
    <Modal>
      {cartItems}
      <div className={classes.total}>
        <span>Total Amount</span>
        <span>35.62</span>
      </div>
      <div className={classes.action}>
        <button className={classes['button--alt']}>Close</button>
        <button className={classes.button}>Order</button>
      </div>
    </Modal>
  );
}
export default Cart;
```

```
import classes from './Modal.module.css';
import ReactDOM from 'react-dom';

const Backdrop = (props) => {
  return <div className={classes.backdrop} />;
};

const ModalOverlay = (props) => {
  return (
    <div className={classes.modal}>
      <div className={classes.content}>{props.children}</div>
    </div>
  );
};

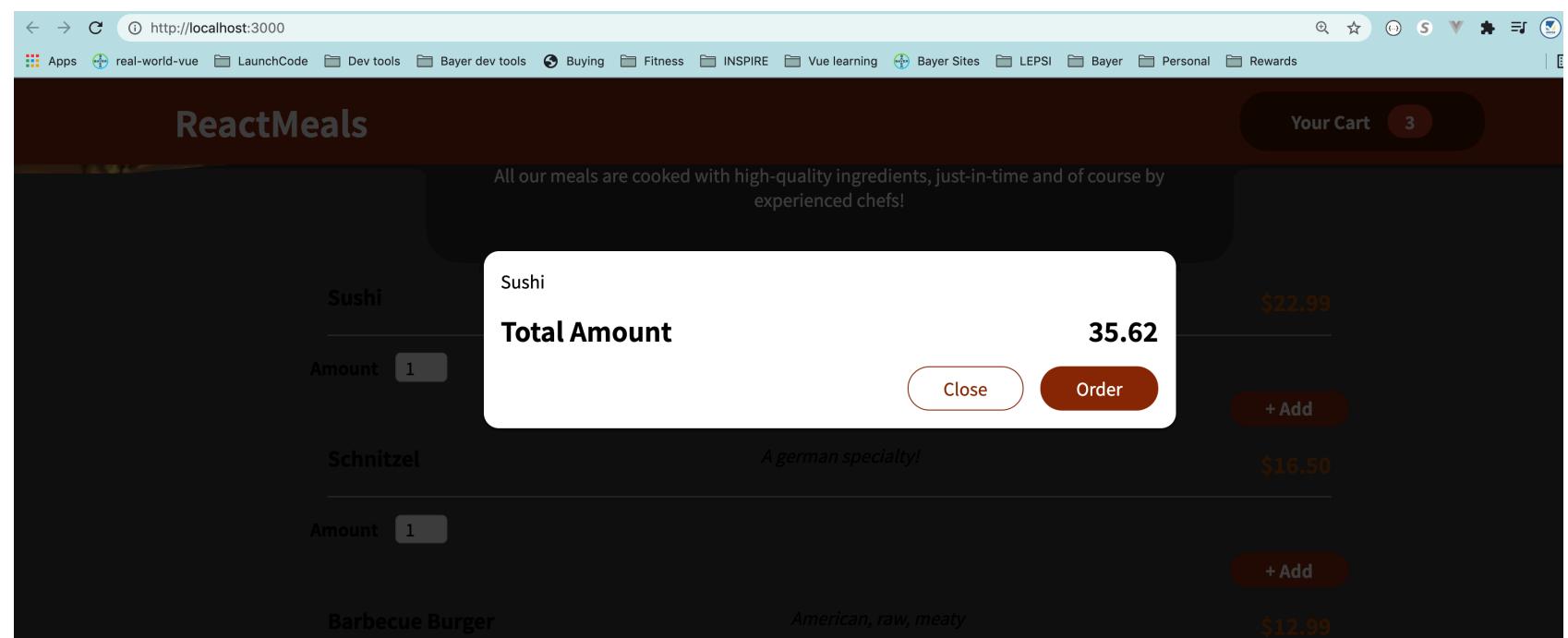
const portalElement = document.getElementById('overlays');
const Modal = (props) => {
  return (
    <>
      {ReactDOM.createPortal(<Backdrop />, portalElement)}
      {ReactDOM.createPortal(
        <ModalOverlay>{props.children}</ModalOverlay>,
        portalElement
      )}
    </>
  );
};

export default Modal;
```

# Adding Modal

```
import Cart from './components/Cart/Cart';
import Header from './components/Layout/Header';
import Meals from './components/Meals/Meals';
function App() {
  return (
    <Cart />
    <Header />
    <main>
      | <Meals />
    </main>
  );
}

export default App;
```



# Managing Cart and Modal State – prop chain

```
import classes from './HeaderCartButton.module.css';
import CartIcon from '../Cart/CartIcon';
const HeaderCartButton = (props) => {
  return (
    <button className={classes.button} onClick={props.onClick}>
      <span className={classes.icon}>
        <CartIcon />
      </span>
      <span>Your Cart</span>
      <span className={classes.badge}>3</span>
    </button>
  );
};

export default HeaderCartButton;
```

```
import classes from './Header.module.css';

import mealsImage from '../../../../../assets/meals.jpg';
import HeaderCartButton from './HeaderCartButton';

const Header = (props) => {
  return (
    <>
      <header className={classes.header}>
        <h1>ReactMeals</h1>
        <HeaderCartButton onClick={props.onShowCart} />
      </header>
      <div className={classes['main-image']}>
        <img src={mealsImage} alt="A table full of delicious food" />
      </div>
    </>
  );
};

export default Header;
```

```
import { useState } from 'react';
import Cart from './components/Cart/Cart';
import Header from './components/Layout/Header';
import Meals from './components/Meals/Meals';
function App() {
  const [cartIsShown, setCartIsShown] = useState(false);

  const showCartHandler = () => {
    setCartIsShown(true);
  };

  const hideCartHandler = () => {
    setCartIsShown(false);
  };
  return (
    <>
      {cartIsShown && <Cart />}
      <Header onShowCart={showCartHandler} />
      <main>
        <Meals />
      </main>
    </>
  );
}

export default App;
```

# Adding Cart to open- prop chain

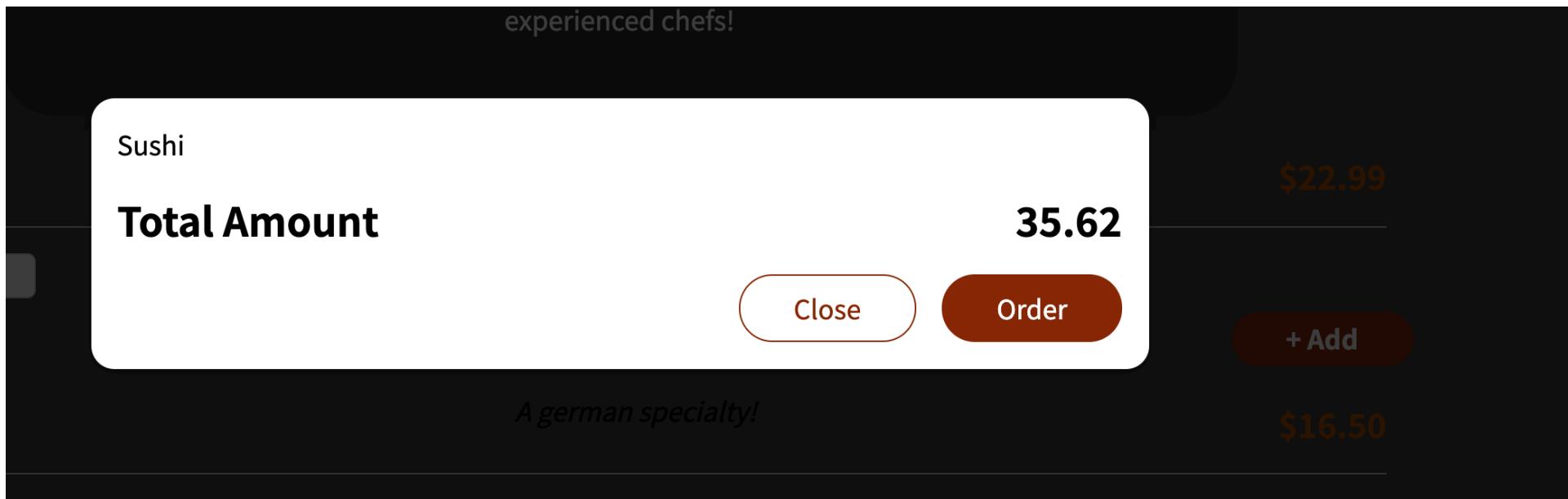
The screenshot shows a dark-themed user interface for a meal delivery service named "ReactMeals". At the top, there's a navigation bar with the "ReactMeals" logo on the left and a "Your Cart" button on the right, which has a small "3" badge indicating items in the cart.

The main content area displays a list of five meal options:

- Sushi**: Described as "Finest fish and veggies". Price: **\$22.99**.  
Quantity input: **1**. Add button: **+ Add**.
- Schnitzel**: Described as "A german specialty!". Price: **\$16.50**.  
Quantity input: **1**. Add button: **+ Add**.
- Barbecue Burger**: Described as "American, raw, meaty". Price: **\$12.99**.  
Quantity input: **1**. Add button: **+ Add**.
- Green Bowl**: Described as "Healthy...and green...". Price: **\$18.99**.  
Quantity input: **1**. Add button: **+ Add**.

Below the meal list, there's a general message: "All our meals are cooked with high-quality ingredients, just-in-time and of course by experienced chefs!"

# Adding Cart to open- prop chain



# Closing Cart – prop chain

```
import { useState } from 'react';
import Cart from './components/Cart/Cart';
import Header from './components/Layout/Header';
import Meals from './components/Meals/Meals';
function App() {
  const [cartIsShown, setCartIsShown] = useState(false);

  const showCartHandler = () => {
    setCartIsShown(true);
  };

  const hideCartHandler = () => {
    setCartIsShown(false);
  };
  return (
    <>
      {cartIsShown && <Cart onClose={hideCartHandler} />}
      <Header onShowCart={showCartHandler} />
      <main>
        <Meals />
      </main>
    </>
  );
}

export default App;
```

```
import Modal from '../UI/Modal';
import classes from './Cart.module.css';
const Cart = (props) => {
  const cartItems = (
    <ul className={classes['cart-items']}>
      {[{ id: 'c1', name: 'Sushi', amount: 2, price: 12.99 }].map((item) => (
        <li>{item.name}</li>
      ))}
    </ul>
  );
  return (
    <Modal onClose={props.onClose}>
      {cartItems}
      <div className={classes.total}>
        <span>Total Amount</span>
        <span>35.62</span>
      </div>
      <div className={classes.actions}>
        <button className={classes['button--alt']} onClick={props.onClose}>
          Close
        </button>
        <button className={classes.button}>Order</button>
      </div>
    </Modal>
  );
};
export default Cart;
```

```
component /> <!-- Modal --> <!-- Modal -->
import classes from './Modal.module.css';
import ReactDOM from 'react-dom';

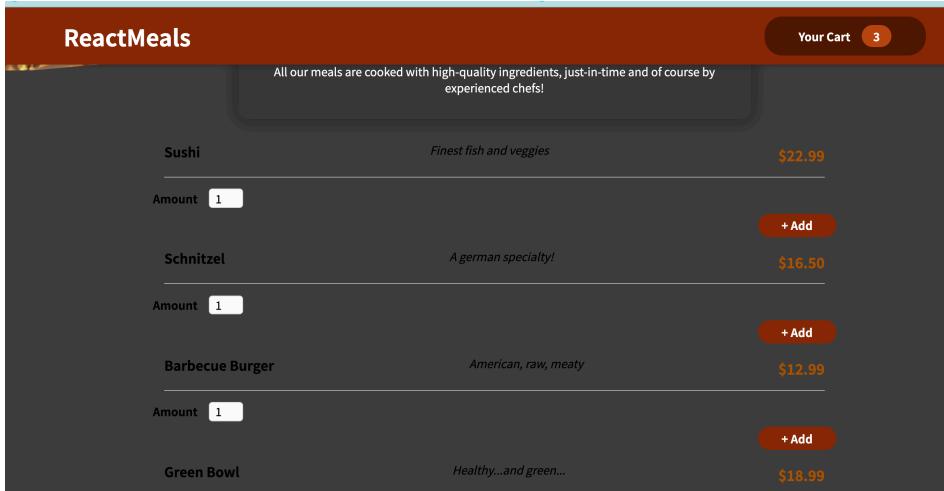
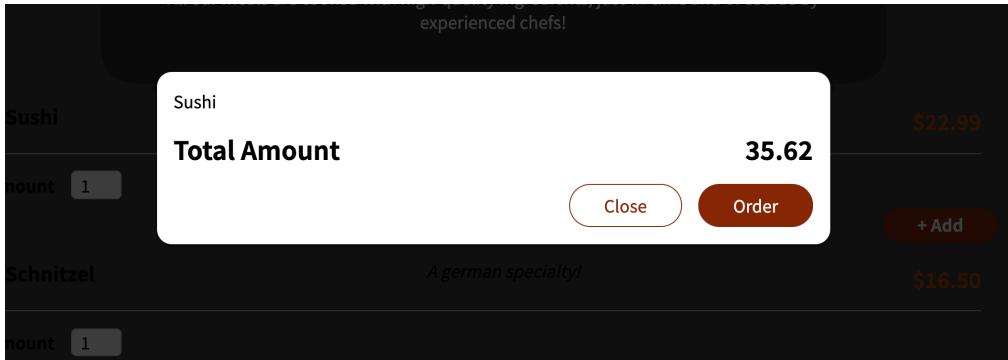
const Backdrop = (props) => {
  return <div className={classes.backdrop} onClick={props.onClick} />;
};

const ModalOverlay = (props) => {
  return (
    <div className={classes.modal}>
      <div className={classes.content}>{props.children}</div>
    </div>
  );
};

const portalElement = document.getElementById('overlays');
const Modal = (props) => {
  return (
    <>
      <ReactDOM.createPortal(<Backdrop onClick={props.onClose} />, portalElement)>
      <ModalOverlay>{props.children}</ModalOverlay>
      </ReactDOM.createPortal(>
    </>
  );
};

export default Modal;
```

# Closing Cart – prop chain



# Adding Cart Context

```
import CartContext from './cart-context';
const CartProvider = (props) => {
  const addItemToCartHandler = (item) => {};
  const removeItemFromCartHandler = (id) => {};
  const cartContent = {
    items: [],
    totalAmount: 0,
    addItem: addItemToCartHandler,
    removeItem: removeItemFromCartHandler,
  };
  return (
    <CartContext.Provider value={cartContent}>
      {props.children}
    </CartContext.Provider>
  );
};
export default CartProvider;
```

```
import React from 'react';
const CartContext = React.createContext({
  items: [],
  totalAmount: 0,
  addItem: (item) => {},
  removeItem: (id) => {},
});

export default CartContext;
```

```
App.js / App / hideCartHandler
import { useState } from 'react';
import Cart from './components/Cart/Cart';
import Header from './components/Layout/Header';
import Meals from './components/Meals/Meals';
import CartProvider from './store/CartProvider';
function App() {
  const [cartIsShown, setCartIsShown] = useState(false);

  const showCartHandler = () => {
    setCartIsShown(true);
  };

  const hideCartHandler = () => [
    setCartIsShown(false),
  ];
  return (
    <CartProvider>
      {cartIsShown && <Cart onClose={hideCartHandler} />}
      <Header onShowCart={showCartHandler} />
      <main>
        <Meals />
      </main>
    </CartProvider>
  );
}

export default App;
```

# Adding Cart Context

```
import { useContext } from 'react';
import CartContext from '../store/cart-context';
import classes from './HeaderCartButton.module.css';
import CartIcon from '../Cart/CartIcon';
const HeaderCartButton = (props) => {
  const cartCtx = useContext(CartContext);
  const numberofCartItems = cartCtx.items.reduce((curNumber, item) => {
    return curNumber + item.amount;
  }, 0);
  return (
    <button className={classes.button} onClick={props.onClick}>
      <span className={classes.icon}>
        | <CartIcon />
      </span>
      <span>Your Cart</span>
      <span className={classes.badge}>{numberofCartItems}</span>
    </button>
  );
};

export default HeaderCartButton;
```

# Adding a Cart Reducer

```
r store > CartProvider.js > cartReducer
import { useReducer } from 'react';
import CartContext from './cart-context';

const defaultCartState = {
  items: [],
  totalAmount: 0,
};

const cartReducer = (state, action) => {
  if(action.type === 'ADD') {
    const updatedItems = state.items.concat(action.item);
    const updatedTotalAmount = state.totalAmount + action.item.price * action.item.amount;
    return {
      items: updatedItems,
      totalAmount: updatedTotalAmount
    }
  }
  else if(action.type === 'REMOVE') {
  }
  return defaultCartState;
};
const CartProvider = (props) => {
  const [cartState, dispatchCartAction] = useReducer(
    cartReducer,
    defaultCartState
  );

  const addItemToCartHandler = (item) => {
    dispatchCartAction({ type: 'ADD', item: item });
  };

  const removeItemFromCartHandler = (id) => {
    dispatchCartAction({ type: 'REMOVE', id: id });
  };
};
```

# Working with Refs and Forward Refs

```
import classes from './MealItemForm.module.css';
import Input from '../../../../../UI/Input';
import { useRef, useState } from 'react';
const MealItemForm = (props) => {
  const [amountIsValid, setAmountIsValid] = useState(true);
  const amountInputRef = useRef();
  const submitHandler = (event) => {
    event.preventDefault();
    const enteredAmount = amountInputRef.current.value;
    const enteredAmountNumber = +enteredAmount;
    if (
      enteredAmount.trim().length === 0 ||
      enteredAmountNumber < 1 ||
      enteredAmountNumber > 5
    ) {
      setAmountIsValid(false);
      return;
    }
    props.onAddToCart(enteredAmountNumber);
  };

  return (
    <form className={classes.form} onSubmit={submitHandler}>
      <Input
        ref={amountInputRef}
        label="Amount"
        input={{
          id: 'amount_' + props.id,
          type: 'number',
          min: '1',
          max: '5',
          step: '1',
          defaultValue: '1',
        }}
      />
      <button>+ Add</button>
      {!amountIsValid && <p>Please enter a valid amount (1-5)</p>}
    </form>
  );
};

export default MealItemForm;
```

```
import React from 'react';
import classes from './Input.module.css';
const Input = React.forwardRef((props, ref) => {
  return (
    <div className={classes.input}>
      <label htmlFor={props.input.id}>{props.label}</label>
      <input ref={ref} id={props.input.id} {...props.input} />
    </div>
  );
});
export default Input;
```

```
components > Meals > MealItem > mealitem.js > mealitem
import { useContext } from 'react';
import classes from './MealItem.module.css';
import MealItemForm from './MealItemForm';
import CartContext from '../../../../../store/cart-context';
const MealItem = (props) => {
  const cartCtx = useContext(CartContext);
  const price = `${props.price.toFixed(2)}`;
  const addToCartHandler = (amount) => {
    cartCtx.addItem({
      id: props.id,
      name: props.name,
      amount: amount,
      price: props.price,
    });
  };
  return (
    <li>
      <div className={classes.meal}>
        <h3>{props.name}</h3>
        <div className={classes.description}>{props.description}</div>
        <div className={classes.price}>{price}</div>
      </div>
      <div>
        <MealItemForm onAddToCart={addToCartHandler} />
      </div>
    </li>
  );
};
export default MealItem;
```

# Working with Refs and Forward Refs

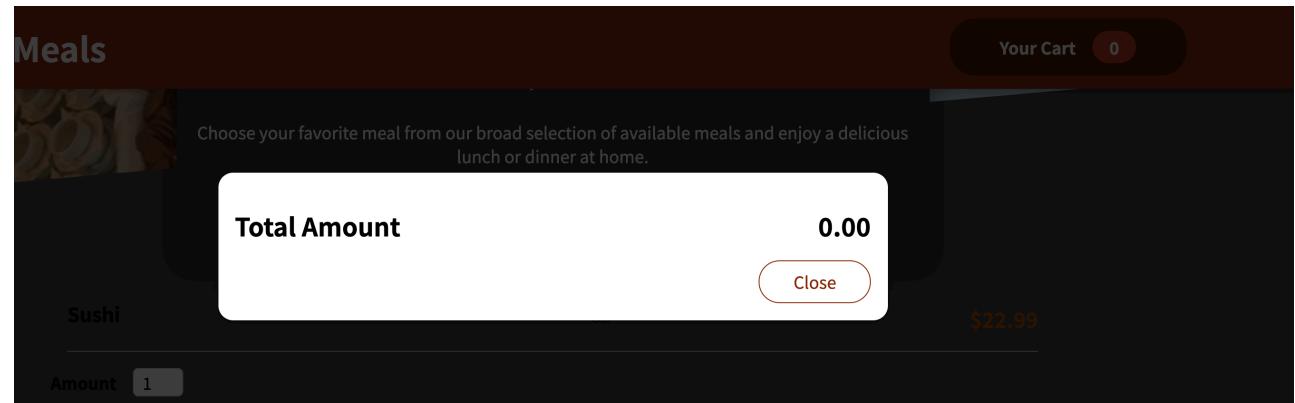
**ReactMeals**

Your Cart 3

Sushi	<i>Finest fish and veggies</i>	\$22.99
Amount	1	+ Add
Schnitzel	<i>A german specialty!</i>	\$16.50
Amount	1	+ Add
Barbecue Burger	<i>American, raw, meaty</i>	\$12.99
Amount	1	+ Add
Green Bowl	<i>Healthy...and green...</i>	\$18.99

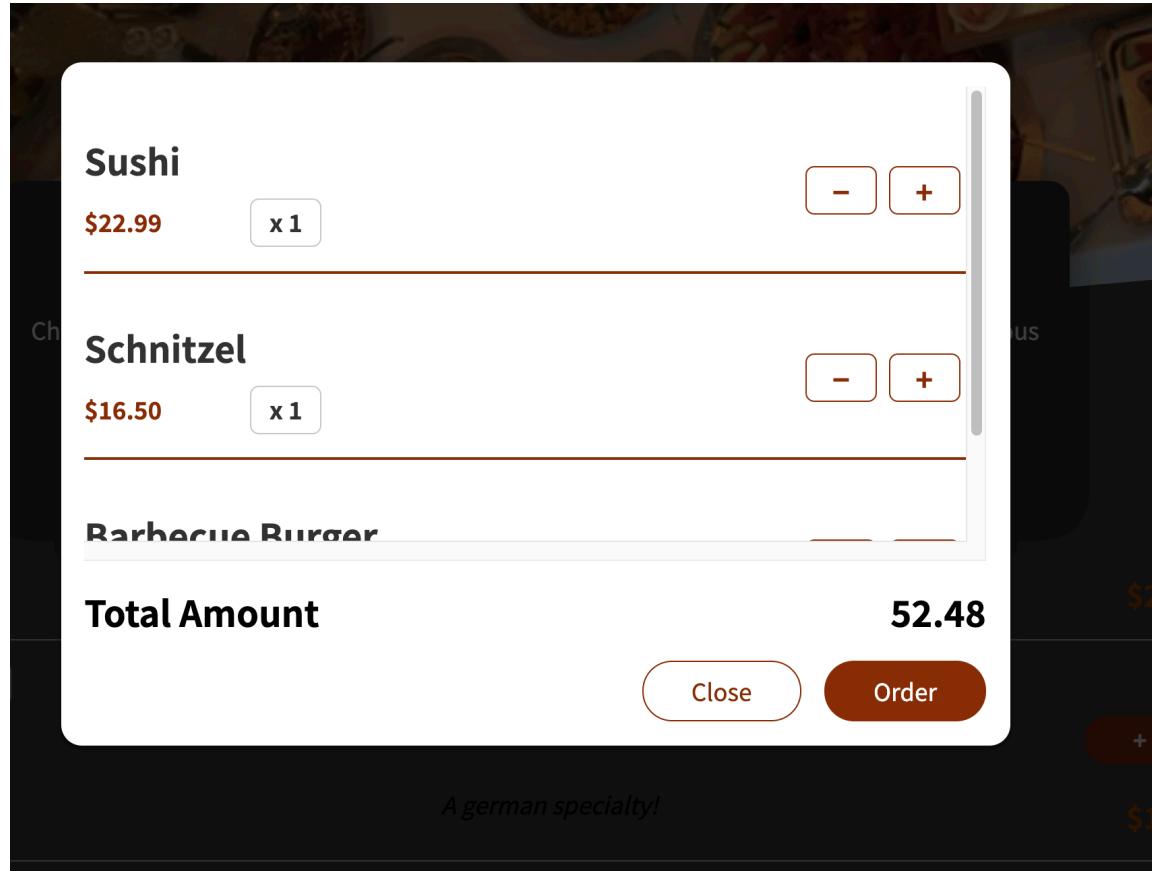
# Outputting Cart items

```
import Modal from '../UI/Modal';
import classes from './Cart.module.css';
import CartContext from '../../store/cart-context';
import { useContext } from 'react/cjs/react.development';
const Cart = (props) => {
  const cartCtx = useContext(CartContext);
  const totalAmount = `${cartCtx.totalAmount.toFixed(2)}`;
  const hasItems = cartCtx.items.length > 0;
  const cartItems = (
    <ul className={classes['cart-items']}>
      {cartCtx.items.map((item) => (
        <li>{item.name}</li>
      ))}
    </ul>
  );
  return (
    <Modal onClose={props.onClose}>
      {cartItems}
      <div className={classes.total}>
        <span>Total Amount</span>
        <span>{totalAmount}</span>
      </div>
      <div className={classes.actions}>
        <button className={classes['button--alt']} onClick={props.onClose}>
          Close
        </button>
        {hasItems && <button className={classes.button}>Order</button>}
      </div>
    </Modal>
  );
}
export default Cart;
```



# Outputting Cart items

```
import Modal from '../UI/Modal';
import classes from './Cart.module.css';
import CartContext from '../../../../../store/cart-context';
import CartItem from './CartItem';
import { useContext } from 'react/cjs/react.development';
const Cart = (props) => {
  const cartCtx = useContext(CartContext);
  const totalAmount = `${cartCtx.totalAmount.toFixed(2)}`;
  const hasItems = cartCtx.items.length > 0;
  const cartItemRemoveHandler = (id) => {};
  const cartItemAddHandler = (id) => {};
  const cartItems = (
    <ul className={classes['cart-items']}>
      {cartCtx.items.map((item) => (
        <CartItem
          key={item.id}
          name={item.name}
          amount={item.amount}
          price={item.price}
          onRemove={cartItemRemoveHandler.bind(null, item.id)}
          onAdd={cartItemAddHandler.bind(null, item)}
        />
      ))}
    </ul>
  );
  return (
    <Modal onClose={props.onClose}>
      {cartItems}
      <div className={classes.total}>
        <span>Total Amount</span>
        <span>{totalAmount}</span>
      </div>
      <div className={classes.actions}>
        <button className={classes['button--alt']} onClick={props.onClose}>
          | Close
        </button>
        {hasItems && <button className={classes.button}>Order</button>}
      </div>
    </Modal>
  );
}
export default Cart;
```



# Aggregating Cart items

# Studio