AP Lab 9

BSCS 7B

Muhammad Tayyab

217038

# Code:

## Apps.js:

import React, { Component } from 'react';

import Products from './components/Products';

import Filter from './components/Filter';

import Basket from './components/Basket';

import './App.css';

class App extends Component {

constructor() {

super();

this.state = { size: '', sort: '', cartItems: [], products: [], filteredProducts: [] };

}

componentWillMount() {

if (localStorage.getItem('cartItems')) {

this.setState({ cartItems: JSON.parse(localStorage.getItem('cartItems')) });

}

fetch('http://localhost:8000/products').then(res => res.json())

.catch(err => fetch('db.json').then(res => res.json()).then(data => data.products))

.then(data => {

this.setState({ products: data });

this.listProducts();

});

}

handleRemoveFromCart = (e, product) => {

this.setState(state => {

const cartItems = state.cartItems.filter(a => a.id !== product.id);

localStorage.setItem('cartItems', JSON.stringify(cartItems));

return { cartItems: cartItems };

})

}

handleAddToCart = (e, product) => {

this.setState(state => {

const cartItems = state.cartItems;

let productAlreadyInCart = false;

cartItems.forEach(cp => {

if (cp.id === product.id) {

cp.count += 1;

productAlreadyInCart = true;

}

});

if (!productAlreadyInCart) {

cartItems.push({ ...product, count: 1 });

}

localStorage.setItem('cartItems', JSON.stringify(cartItems));

return { cartItems: cartItems };

});

}

listProducts = () => {

this.setState(state => {

if (state.sort !== '') {

state.products.sort((a, b) =>

(state.sort === 'lowestprice'

? ((a.price > b.price) ? 1 : -1)

: ((a.price < b.price) ? 1 : -1)));

} else {

state.products.sort((a, b) => (a.id > b.id) ? 1 : -1);

}

if (state.size !== '') {

return { filteredProducts: state.products.filter(a => a.availableSizes.indexOf(state.size.toUpperCase()) >= 0) };

}

return { filteredProducts: state.products };

})

}

handleSortChange = (e) => {

this.setState({ sort: e.target.value });

this.listProducts();

}

handleSizeChange = (e) => {

this.setState({ size: e.target.value });

this.listProducts();

}

render() {

return (

<div className="container">

<h1>E-commerce Shopping Cart Application</h1>

<hr />

<div className="row">

<div className="col-md-9">

<Filter count={this.state.filteredProducts.length} handleSortChange={this.handleSortChange}

handleSizeChange={this.handleSizeChange} />

<hr />

<Products products={this.state.filteredProducts} handleAddToCart={this.handleAddToCart} />

</div>

<div className="col-md-3">

<Basket cartItems={this.state.cartItems} handleRemoveFromCart={this.handleRemoveFromCart} />

</div>

</div>

</div>

);

}

}

export default App;

## Index.js:

import React from 'react';

import ReactDOM from 'react-dom';

import './index.css';

import App from './App';

import \* as serviceWorker from './serviceWorker';

ReactDOM.render(<App />, document.getElementById('root'));

// If you want your app to work offline and load faster, you can change

// unregister() to register() below. Note this comes with some pitfalls.

// Learn more about service workers: https://bit.ly/CRA-PWA

serviceWorker.unregister();

## App.css:

.App {

text-align: center;

}

.App-logo {

animation: App-logo-spin infinite 20s linear;

height: 40vmin;

pointer-events: none;

}

.App-header {

background-color: #282c34;

min-height: 100vh;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

font-size: calc(10px + 2vmin);

color: white;

}

.App-link {

color: #61dafb;

}

@keyframes App-logo-spin {

from {

transform: rotate(0deg);

}

to {

transform: rotate(360deg);

}

}

## Products.js:

import React, { Component } from 'react';

import util from '../util'

export default class Products extends Component {

render() {

const productItems = this.props.products.map(product => (

<div className="col-md-4" key={product.id}>

<div className="thumbnail text-center">

<a href={`#${product.id}`}onClick={(e)=>this.props.handleAddToCart(e, product)}>

<img src={`products/${product.sku}\_2.jpg`} alt={product.title} />

<p>{product.title}</p>

</a>

<b>{util.formatCurrency(product.price)}</b>

<button className="btn btn-primary" onClick={(e)=>this.props.handleAddToCart(e, product)}>Add to cart</button>

</div>

</div>

));

return (

<div className="row">

{productItems}

</div>

)

}

}

## Filter.js:

import React, { Component } from 'react';

export default class Products extends Component {

render() {

return (

<div className="row">

<div className="col-md-4">

{`${this.props.count} products found.`}

</div>

<div className="col-md-4">

<label>Order by

<select className="form-control" value={this.props.sort} onChange={this.props.handleSortChange}>

<option value="">Select</option>

<option value="lowestprice">Lowest to highest</option>

<option value="highestprice">Highest to lowest</option>

</select>

</label>

</div>

<div className="col-md-4">

<label > Filter Size

<select className="form-control" value={this.props.size} onChange={this.props.handleSizeChange}>

<option value="">ALL</option>

<option value="x">XS</option>

<option value="s">S</option>

<option value="m">M</option>

<option value="l">L</option>

<option value="xl">XL</option>

<option value="xxl">XXL</option>

</select>

</label>

</div>

</div>

)

}

}

## Basket.js:

import React, { Component } from 'react';

import util from '../util'

export default class Basket extends Component {

render() {

const { cartItems } = this.props;

return (

<div className="alert alert-info">

{cartItems.length === 0

? "Basket is empty" :

<div>You have {cartItems.length} items in the basket. <hr /></div>

}

{cartItems.length > 0 &&

<div>

<ul style={{ marginLeft: -25 }}>

{cartItems.map(item => (

<li key={item.id}>

<b>{item.title}</b>

<button style={{ float: 'right' }} className="btn btn-danger btn-xs"

onClick={(e) => this.props.handleRemoveFromCart(e, item)}>X</button>

<br />

{item.count} X {util.formatCurrency(item.price)}

</li>))

}

</ul>

<b>Sum: {util.formatCurrency(cartItems.reduce((a, c) => (a + c.price \* c.count), 0))}

</b>

<button onClick={() => alert('Todo: Implement checkout page.')} className="btn btn-primary">checkout</button>

</div>

}

</div>

)

}

}

## Index.html:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<link rel="shortcut icon" href="%PUBLIC\_URL%/favicon.ico" />

<meta

name="viewport"

content="width=device-width, initial-scale=1, shrink-to-fit=no"

/>

<meta name="theme-color" content="#000000" />

<!--

manifest.json provides metadata used when your web app is installed on a

user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-manifest/

-->

<link rel="manifest" href="%PUBLIC\_URL%/manifest.json" />

<!--

Notice the use of %PUBLIC\_URL% in the tags above.

It will be replaced with the URL of the `public` folder during the build.

Only files inside the `public` folder can be referenced from the HTML.

Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC\_URL%/favicon.ico" will

work correctly both with client-side routing and a non-root public URL.

Learn how to configure a non-root public URL by running `npm run build`.

-->

<link rel="stylesheet" href="https://bootswatch.com/3/superhero/bootstrap.min.css">

<title>React App</title>

</head>

<body>

<noscript>You need to enable JavaScript to run this app.</noscript>

<div id="root"></div>

<!--

This HTML file is a template.

If you open it directly in the browser, you will see an empty page.

You can add webfonts, meta tags, or analytics to this file.

The build step will place the bundled scripts into the <body> tag.

To begin the development, run `npm start` or `yarn start`.

To create a production bundle, use `npm run build` or `yarn build`.

-->

</body>

</html>

# Output:





