

React

Code Café React App

Dr. Atef Bader

Code Café React App

The screenshot shows a web browser window with the O'Reilly Library website. The page title is "Chapter 9. Router". The main text describes the "Code Café" app and the problem of routing. A red box highlights the text: "You will use the popular React Router library to create routes in your application. By the end of this chapter, clicking a `Thumbnail` item will route the user to a new URL and display the item's name (Figure 9.1, 'Item details page'). Later, you will add the functionality for customers to see details about the item and, finally, order an item from the details page." A red arrow points from this text to a blue callout box on the right.

Chapter 9. Router

The home page looks great. Visitors to Code Café find a stylish app and can easily see what items are available for purchase.

Unfortunately, clicking a thumbnail does not do anything. Customers have no way to buy or even find out more about the café's tasty treats.

In this chapter, you will take a step toward fixing that. It is time to introduce routing to your application.

In React, you use routing to keep the UI in sync as a user navigates through the application. Remember that React is declarative, so you define the UI for a URL, and React will render the elements needed to make it happen.

You will use the popular React Router library to create routes in your application. By the end of this chapter, clicking a `Thumbnail` item will route the user to a new URL and display the item's name (Figure 9.1, "Item details page"). Later, you will add the functionality for customers to see details about the item and, finally, order an item from the details page.

8. Interacting with a Server 9. Router 10. Conditional Rendering

6h 42m remaining

cart Highlight All Match Case Match Diacritics Whole Words 53 of 139 matches

- Use React Router library to create routes in your application.

- Clicking a Thumbnail item will route the user to a new URL and display the item's name

Code Café React App

- App.js has new routes:

- /details
- /:id
- /*
- /

```
09-router > src > JS App.js > App
1  import axios from 'axios';
2  import { useEffect, useState } from 'react';
3  import {
4    BrowserRouter as Router,
5    Routes,
6    Route,
7  } from 'react-router-dom';
8  import DetailItem from './components/DetailItem';
9  import Details from './components/Details';
10 import Header from './components/Header';
11 import Home from './components/Home';
12 import NotFound from './components/NotFound';
13
14 function App() {
15   const [items, setItems] = useState([]);
16
17   useEffect(() => {
18     axios.get('/api/items')
19       .then((result) => setItems(result.data))
20       .catch(console.error);
21   }, []);
22
23   return (
24     <Router>
25       <Header />
26       <Routes>
27         <Route path="/details" element={<Details items={items} />} />
28         <Route path=":id" element={<DetailItem />} />
29         <Route index element={<div>No Item Selected</div>} />
30       </Routes>
31       <Route path="/" element={<Home items={items} />} />
32       <Route path="*" element={<NotFound />} />
33     </Routes>
34   </Router>
35 );
36 }
37
38 export default App;
```

Code Café React App

- Startup your backend server on port 3030

```
> npm start  
 > node app.js  
  
Listening on http://localhost:3030  
Accessing /api/items  
Accessing /api/items  
Accessing /  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /main.69b8defce96165e7e728.hot-update.json  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items
```

Code Café React App

Select Windows PowerShell

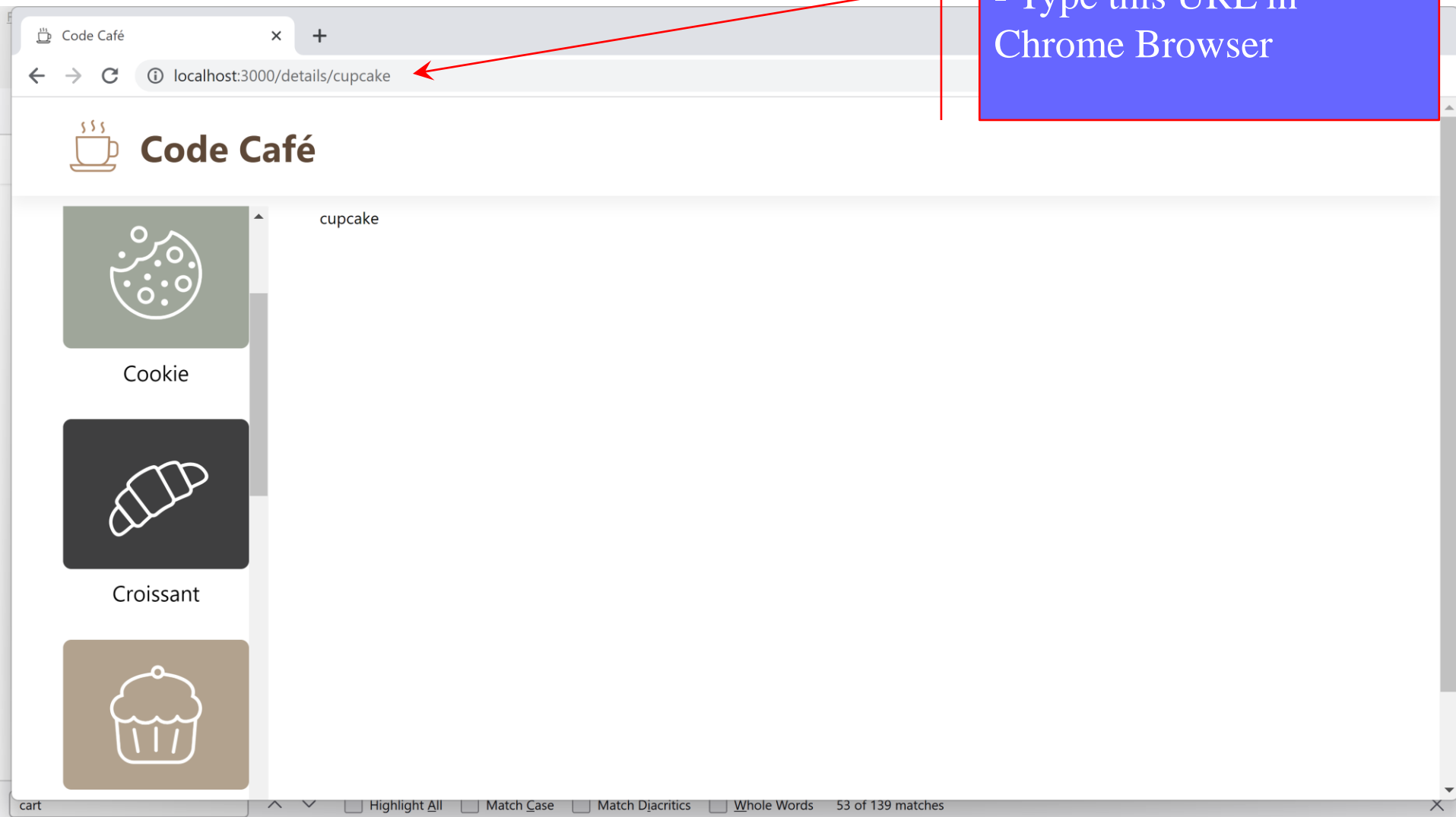
Note that the development build is not optimized.
To create a production build, use `npm run build`.

webpack compiled **successfully**

- Startup React server on
port 3000

Code Café React App

- Type this URL in
Chrome Browser



Code Café React App

File Edit View History Bookmarks Tools Help

O'Reilly Library: USABILITY ENGINEERING 11. useReducer | React Programming

https://learning.oreilly.com/library/view/react-programming-the/9780137901777/ch11.xhtml 67%

O'REILLY Topics Start Learning Search 50,000+ courses, events, titles, and more

Chapter 11. useReducer

In [Chapter 4, State](#), you used the `useState` hook to access a component's state. In this chapter, you will explore another React hook: `useReducer`, which lets you set, modify, and access a state variable within a component.

You will take advantage of `useReducer` to add a shopping cart to Code Café and manage its ongoing value.

If you have previously used Redux, a third-party library for managing state, some of the patterns you see in this chapter might be familiar. However, you do not need any prior knowledge of or experience with Redux to use the `useReducer` hook.

By the end of this chapter, users will be able to add items to a cart, and a new cart icon in the header will display the number of items in the cart ([Figure 11.1, "Code Café's header at the end of the chapter"](#)).

10. Conditional Rendering 11. useReducer 12. Editing the Cart

6h 42m remaining

cart ^ v ☐ Highlight All ☐ Match Case ☐ Match Diacritics ☐ Whole Words 53 of 139 matches

You will take advantage of `useReducer` to add a shopping cart to Code Café and manage its ongoing value.

Code Café React App

- App.js uses cartReducer to add items to the cart

```
12-editing-the-cart > src > JS App.js > App > useEffect() callback
1  import axios from 'axios';
2  import { useEffect, useReducer, useState } from 'react';
3  import {
4    BrowserRouter as Router,
5    Routes,
6    Route,
7  } from 'react-router-dom';
8  import DetailItem from './components/DetailItem';
9  import Cart from './components/Cart';
10 import Details from './components/Details';
11 import Header from './components/Header';
12 import Home from './components/Home';
13 import NotFound from './components/NotFound';
14 import { cartReducer, CartTypes, initialCartState } from './reducers/cartReducer';
15
16 function App() {
17   const [items, setItems] = useState([]);
18   const [cart, dispatch] = useReducer(cartReducer, initialCartState);
19   const addToCart = (itemId) => dispatch({ type: CartTypes.ADD, itemId });
20
21   useEffect(() => {
22     axios.get('/api/items')
23       .then((result) => setItems(result.data))
24       .catch(console.error);
25   }, []);
26
27   return (
28     <Router>
29       <Header cart={cart} />
30       {items.length === 0
31         ? <div>Loading...</div>
32         : (
33           <Routes>
34             <Route
35               path="/cart"
36               element=<Cart cart={cart} dispatch={dispatch} items={items} />
37             />
38             <Route path="/details" element=<Details items={items} />
39             <Route
40               path="/"
41             />
42           </Routes>
43         )
44       }
45     </Router>
46   );
47 }
```


Code Café React App

- Startup your backend server on port 3030

```
> npm start  
 > node app.js  
  
Listening on http://localhost:3030  
Accessing /api/items  
Accessing /api/items  
Accessing /  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /main.69b8defce96165e7e728.hot-update.json  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items
```

Code Café React App

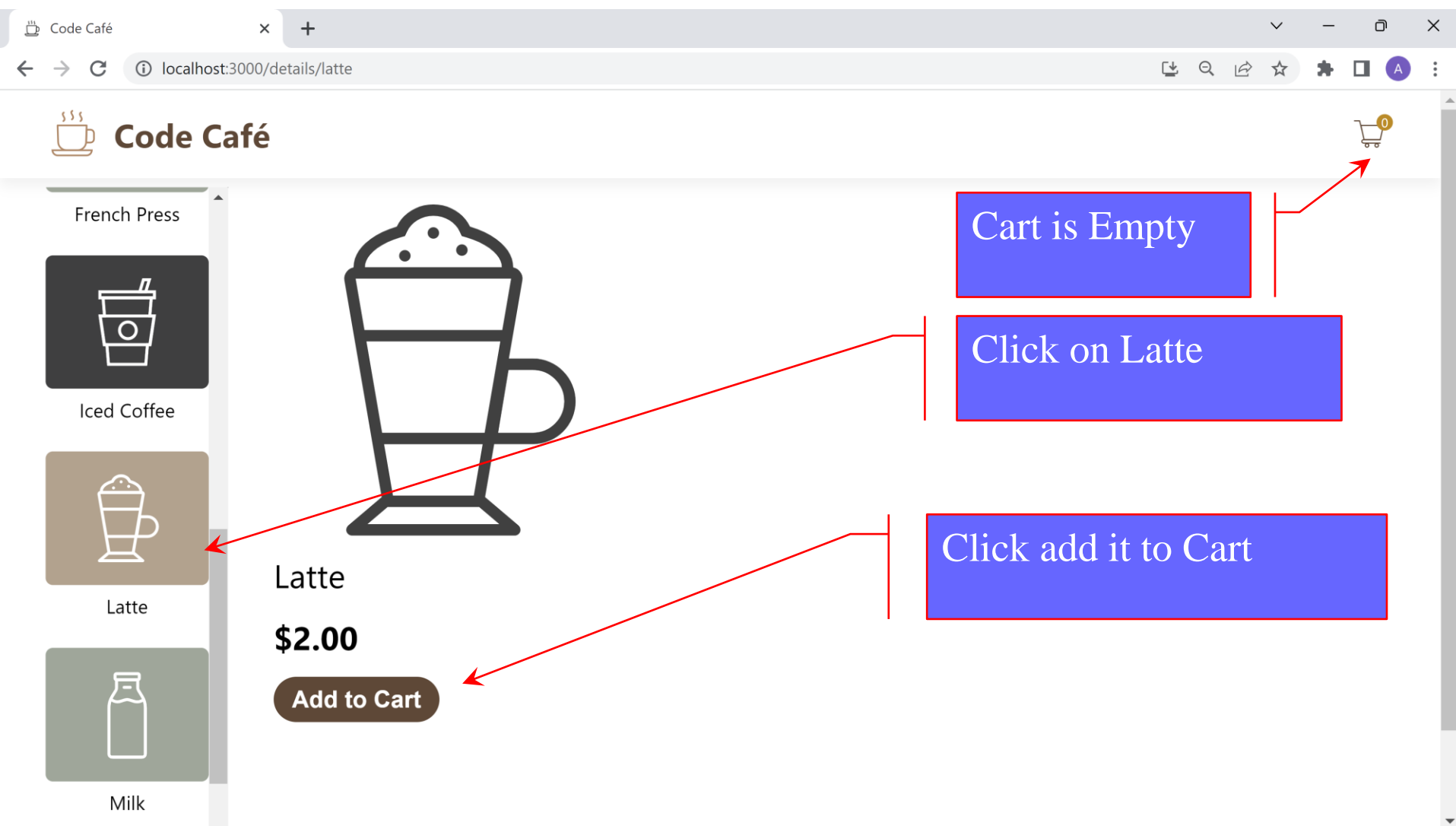
Select Windows PowerShell

Note that the development build is not optimized.
To create a production build, use `npm run build`.

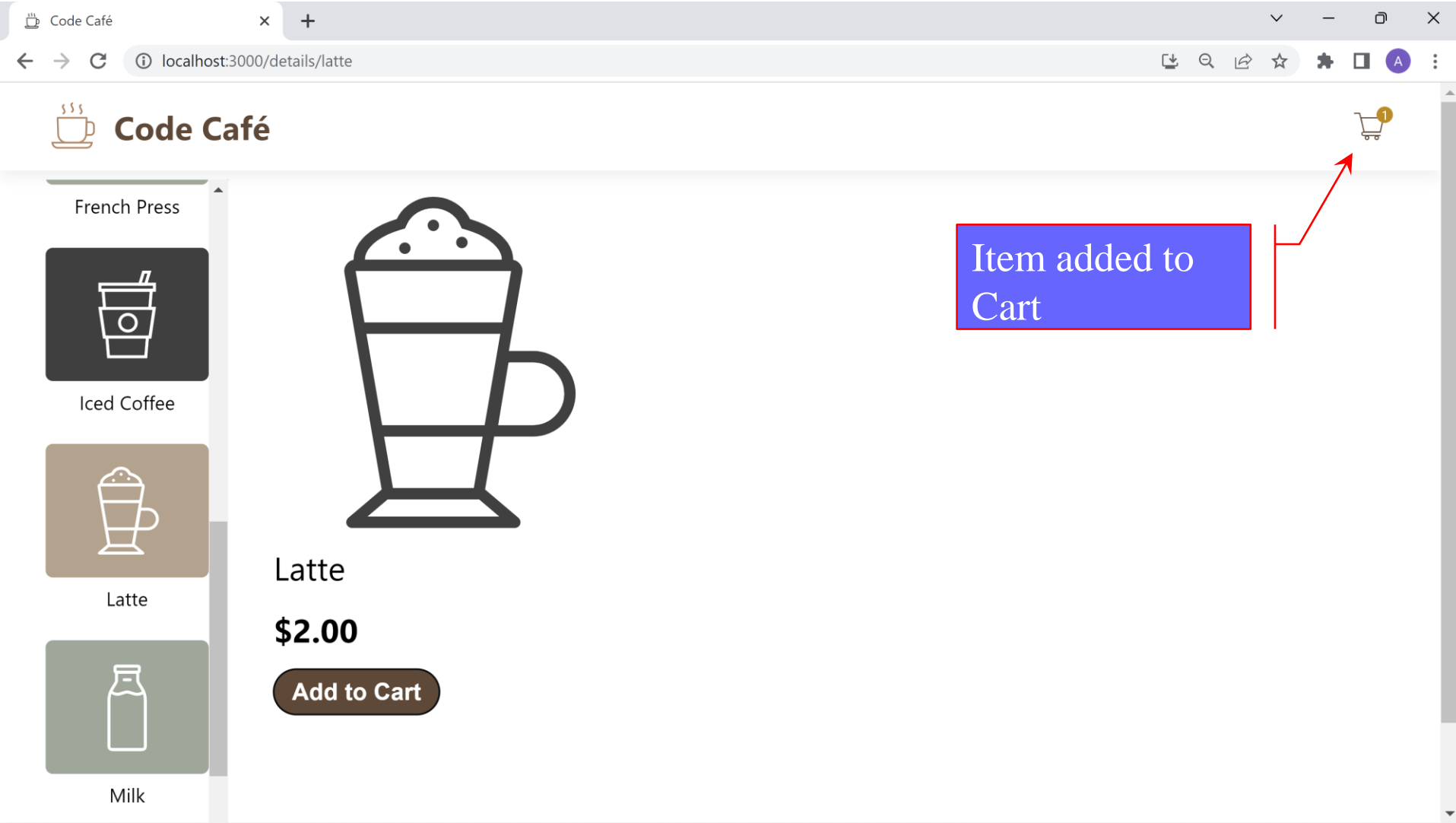
webpack compiled **successfully**

- Startup React server on
port 3000

Code Café React App



Code Café React App



Code Café React App

File Edit View History Bookmarks Tools Help

O'Reilly Library: USABILITY ENGINEERING

12. Editing the Cart | React Programming

https://learning.oreilly.com/library/view/react-programming-the/9780137901777/ch12.xht 67%

O'REILLY Topics Start Learning Search 50,000+ courses, events, titles, and more

Chapter 12. Editing the Cart

Developing Code Café has given you experience with building functional components. You have also added state to keep track of the user's `cart` and are using the `useReducer` hook to manage that state.

In this chapter, you will create a route for users to view and edit the cart. This chapter uses concepts you already know, giving you the opportunity to practice your skills as you expand your app.

Try to write each new piece of code without first looking at the snippets we provide. When you finish, compare your code with ours. You might do some things differently from how we do them – that is fine, as long as your app behaves as expected. But we do recommend that you edit your code to match ours before moving forward to make sure you do not run into compatibility issues later.

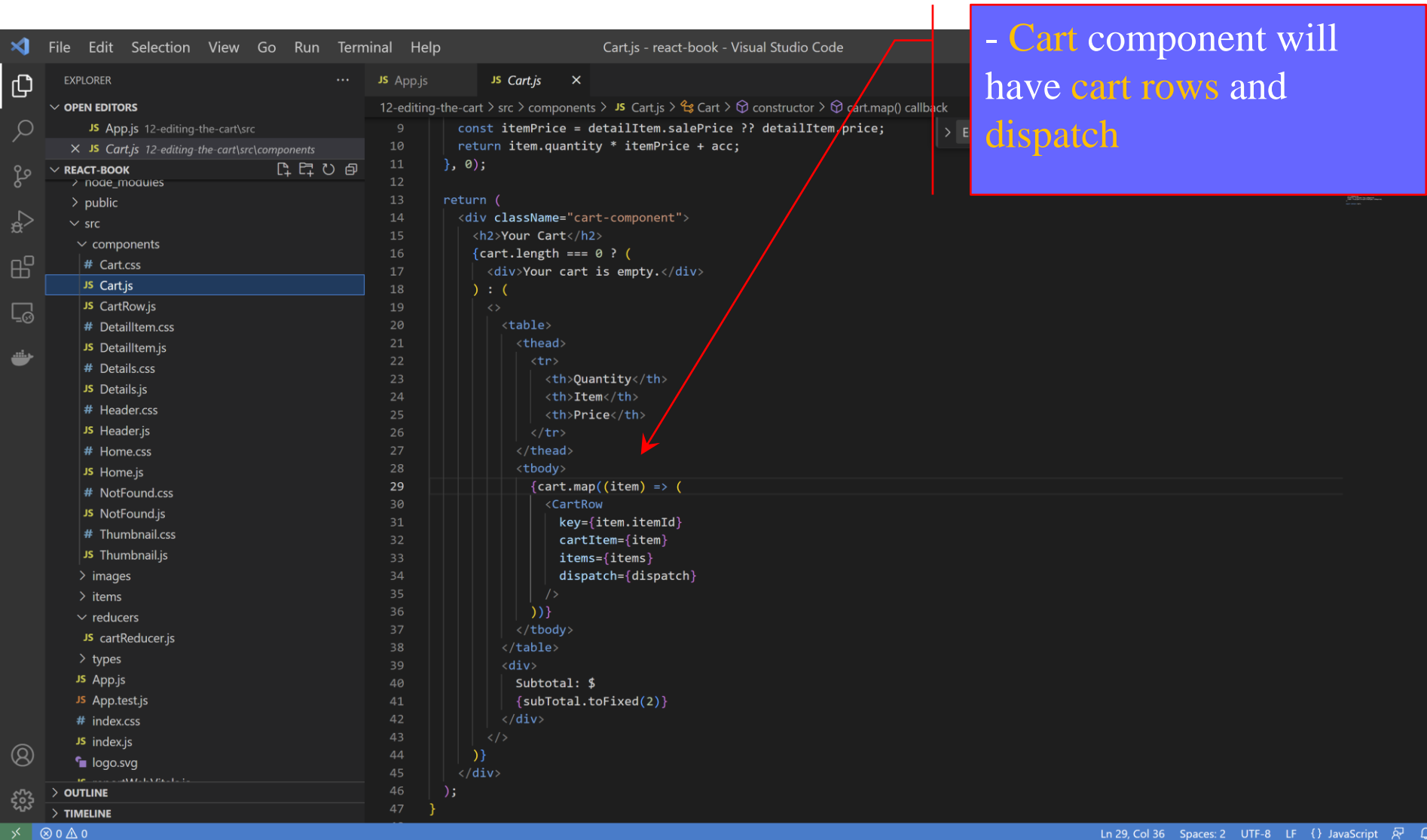
Figure 12.1, “Completed cart” shows what your cart page will look like at the end of the chapter:

11. useReducer 12. Editing the Cart 13. Forms

6h 42m remaining

cart ^ v Highlight All Match Case Match Diacritics Whole Words 53 of 139 matches

Code Café React App



The screenshot shows the Visual Studio Code editor with the `Cart.js` file open. The file is part of the `react-book` project, specifically in the `src/components` directory. The code defines a `Cart` component that renders a table of cart items. A red arrow points from a text box to the `CartRow` component in the table's `tbody`.

```
9      const itemPrice = detailItem.salePrice ?? detailItem.price;
10     return item.quantity * itemPrice + acc;
11   }, 0);
12
13   return (
14     <div className="cart-component">
15       <h2>Your Cart</h2>
16       {cart.length === 0 ? (
17         <div>Your cart is empty.</div>
18       ) : (
19         <>
20           <table>
21             <thead>
22               <tr>
23                 <th>Quantity</th>
24                 <th>Item</th>
25                 <th>Price</th>
26               </tr>
27             </thead>
28             <tbody>
29               {cart.map((item) => (
30                 <CartRow
31                   key={item.itemId}
32                   cartItem={item}
33                   items={items}
34                   dispatch={dispatch}
35                 />
36               ))}
37             </tbody>
38           </table>
39           <div>
40             Subtotal: $
41             {subTotal.toFixed(2)}
42           </div>
43         </>
44       )}
45     </div>
46   );
47 }
```

- Cart component will have cart rows and dispatch

Code Café React App

- Startup your backend server on port 3030

```
> npm start  
 > node app.js
```

Listening on http://localhost:3030 ←

```
Accessing /api/items  
Accessing /api/items  
Accessing /  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /main.69b8defce96165e7e728.hot-update.json  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items  
Accessing /api/items
```

Code Café React App

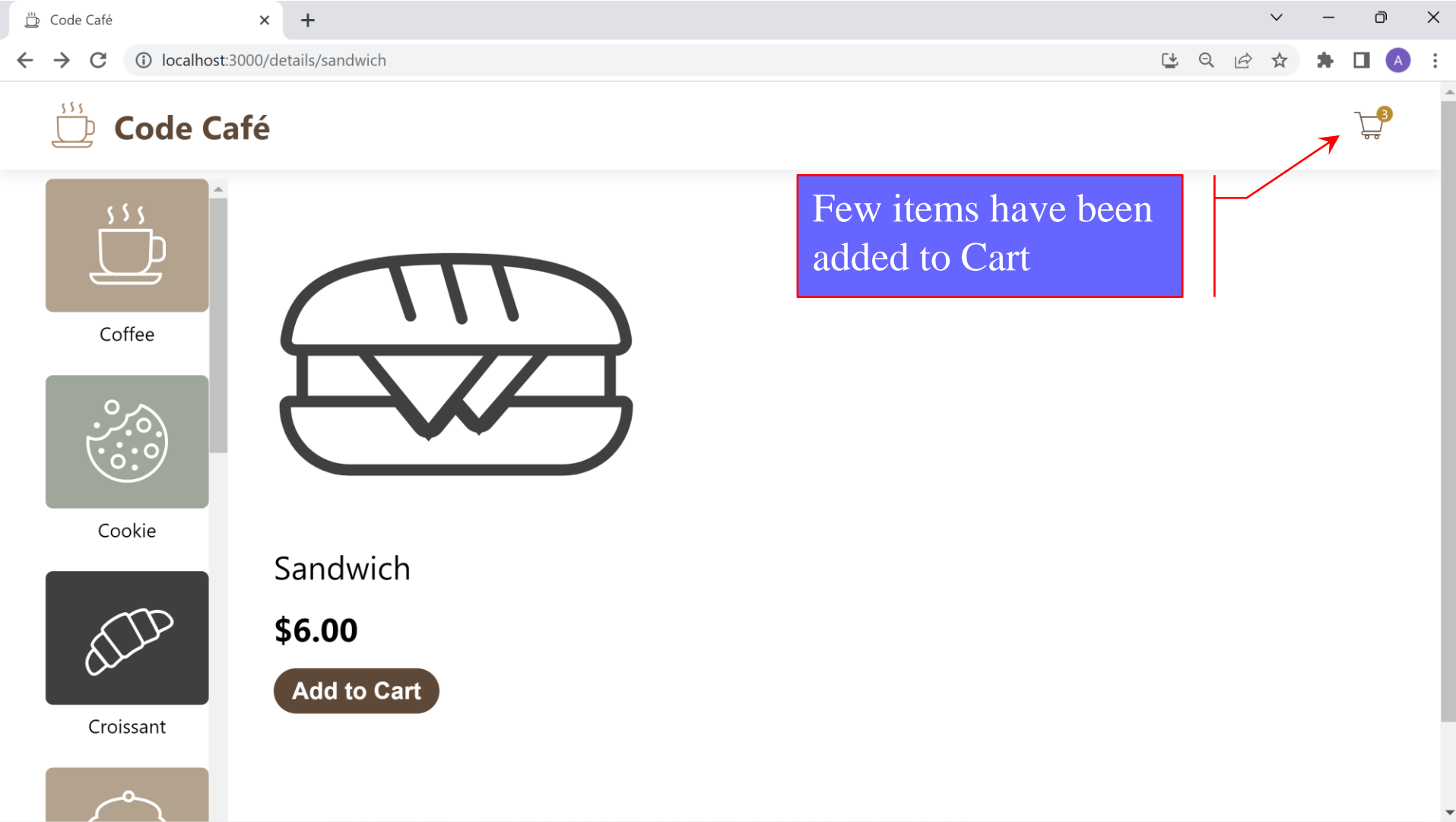
Select Windows PowerShell

Note that the development build is not optimized.
To create a production build, use `npm run build`.

webpack compiled **successfully**

- Startup React server on
port 3000

Code Café React App



Code Café React App

The screenshot shows a web browser window with the URL `localhost:3000/cart`. The page header features the "Code Café" logo and a shopping cart icon with a "3" badge. A blue box labeled "Click the Cart" has an arrow pointing to the cart icon. Below the header, the section "Your Cart" contains a table with three items: Milk, Latte, and Sandwich. Each item has a quantity of 1 and a price. To the right of each item is a yellow circle with a black "x". A blue box labeled "Content of Cart listed as a Tables of items" has an arrow pointing to the table. Another blue box labeled "Click x to remove item and update Cart" has an arrow pointing to one of the "x" icons. The subtotal is listed as \$8.50.

Code Café

Click the Cart

Your Cart

Quantity	Item	Price
1	Milk	\$0.50
1	Latte	\$2.00
1	Sandwich	\$6.00

Subtotal: \$8.50



Content of Cart listed as a Tables of items

Click x to remove item and update Cart

Code Café React App

The screenshot shows a web browser at `localhost:3000/cart` displaying the 'Code Café' shopping cart. The cart contains two items: Milk (\$0.50) and Sandwich (\$6.00). The subtotal is \$6.50. Three blue callout boxes with red arrows provide context:

- Cart got updated**: Points to the shopping cart icon in the top right corner, which has a yellow badge with the number '2'.
- Content of Cart list got updated**: Points to the table of items in the cart.
- Latte item go REMOVED from the list**: Points to the same table, indicating a previous item is no longer present.

Quantity	Item	Price
1	Milk	\$0.50 
1	Sandwich	\$6.00 

Subtotal: \$6.50

Code Café React App

File Edit View History Bookmarks Tools Help

O'Reilly Library: USABILITY ENGINEERING 13. Forms | React Programming: The Way of Thinking

https://learning.oreilly.com/library/view/react-programming-the-way-of-thinking/9780137901777/

O'REILLY Topics Start Learning Search 50,000+ courses, events, titles, and more

Chapter 13. Forms

Code Café's users can add tasty treats to the cart and update it to have the items they want. Next, they need to be able to submit their orders.

When the user places an order, you need to collect certain information, such as the customer's name. To do this, you will add a form to the cart page.

Forms allow users to input and submit information. If you are familiar with HTML forms, you will probably recognize the basic elements used to build forms in React. Because these elements are interactive and used to capture input, there are some caveats to working with them in React. This chapter will explore ways to implement form elements in your application, including managing state.

When the user is ready to place an order, they will need to enter their name and their ZIP code (to compute tax). They will also have the option to enter a phone number. To keep things simple, you will add the checkout form right on the cart page, rather than creating a new route (**Figure 13.1. "Completed checkout**

12. Editing the Cart 13. Forms 14. Local Storage and useRef

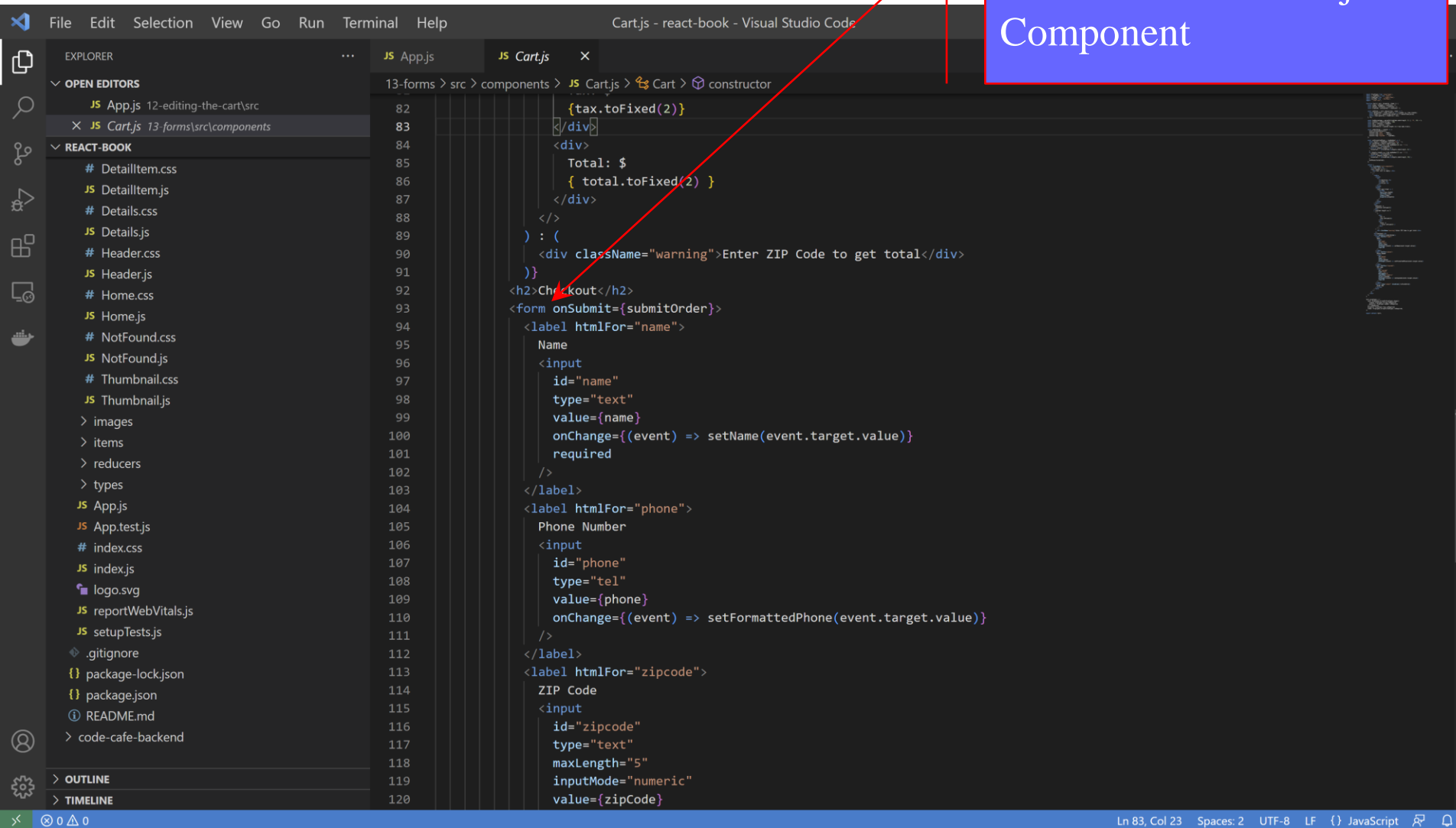
6h 42m remaining

cart ^ v ☐ Highlight All ☐ Match Case ☐ Match Diacritics ☐ Whole Words 53 of 139 matches

When the user places an order, you need to collect certain information, such as the customer's name. To do this, you will add a form to the cart page.

Code Café React App

- Add a form to Cart.js Component



```
82      {tax.toFixed(2)}
83    </div>
84    <div>
85      Total: $
86      { total.toFixed(2) }
87    </div>
88  </>
89  ) : (
90    <div className="warning">Enter ZIP Code to get total</div>
91  )}
92  <h2>Checkout</h2>
93  <form onSubmit={submitOrder}>
94    <label htmlFor="name">
95      Name
96      <input
97        id="name"
98        type="text"
99        value={name}
100        onChange={(event) => setName(event.target.value)}
101        required
102      />
103    </label>
104    <label htmlFor="phone">
105      Phone Number
106      <input
107        id="phone"
108        type="tel"
109        value={phone}
110        onChange={(event) => setFormattedPhone(event.target.value)}
111      />
112    </label>
113    <label htmlFor="zipcode">
114      ZIP Code
115      <input
116        id="zipcode"
117        type="text"
118        maxLength="5"
119        inputMode="numeric"
120        value={zipCode}
```

Code Café React App

- Startup your backend server on port 3030

[illegible]

Code Café React App

Select Windows PowerShell

Note that the development build is not optimized.
To create a production build, use `npm run build`.

webpack compiled **successfully**

- Startup React server on
port 3000

Code Café React App

The screenshot displays the Code Café React App interface. The browser address bar shows the URL `localhost:3000/cart`. The app header features the Code Café logo and a shopping cart icon with a notification badge showing the number 4. The main content area is divided into two sections: 'Your Cart' and 'Checkout'.

Your Cart

Quantity	Item	Price
1	Cookie	\$0.50
1	Latte	\$2.00
2	French Press	\$3.50

Subtotal: \$6.00
Tax: \$0.42
Total: \$6.42

Checkout

Name

Phone Number

ZIP Code

Order Now

Annotations with red arrows point to specific elements:

- Cart has item**: Points to the shopping cart icon in the top right corner.
- Tax/Total calculated**: Points to the 'Total: \$6.42' line item in the cart summary.
- User Info Form filled out**: Points to the 'Name' input field containing 'John Smith'.

References

- Rosson, M. B., John M. 2001. *Usability Engineering: Scenario-Based Development of Human-Computer Interaction* (1st Edition). San Francisco, CA: Morgan Kaufmann Publishers. [ISBN: 978-1558607125]
 - Tidwell, J., Brewer, C., Valencia, A. 2020. *Designing Interfaces* (3rd Edition). Sebastopol, CA: O'Reilly Media. [ISBN: 978-1492051961]
 - Klingman, L. 2023. *React Programming: The Big Nerd Ranch Guide*. Indianapolis, IN: Pearson Technology Group. [ISBN: 978-0137901692]
 - De Voil, N. 2016. *User Experience Foundations*. Swindon, UK: BCS Learning & Development. [ISBN: 978-1780173498]
 - Nielsen, J. 1994. *Usability Engineering*. Cambridge, MA: Morgan Kaufmann. [ISBN: 978-0125184069]
 - Ritter, M., Winterbottom, C. 2017. *UX for the Web*. Birmingham, UK: Packt Publishing. [ISBN: 978-1787128477]
 - Nielsen, J., Loranger, H. 2006. *Prioritizing Web Usability*. Berkeley, CA: New Riders. [ISBN: 978-0321350312]
 - Yablonski, J. 2020. *Laws of UX*. Sebastopol, CA: O'Reilly Media. [ISBN: 978-1492055310]
 - MacDonald, D. 2019. *Practical UI Patterns for Design Systems: Fast-Track Interaction Design for a Seamless User Experience*. New York, NY: Springer Science. [ISBN: 978-1484249376]
 - Timms, S. 2016. *Mastering JavaScript Design Patterns (2nd Edition)*. Birmingham, UK: Packt Publishing. [ISBN: 978-1785882166]
 - Larsen, J. 2016. *React Hooks in Action*. Shelter Island, NY: Manning Publications. [ISBN: 978-1617297632]
 - Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., Diakopoulos, N. 2016. *Designing the User Interface: Strategies for Effective Human-Computer Interaction (6th Edition)*. Essex, England: Pearson Education Publications. [ISBN: 978-1292153919]
 - Flanagan, D. 2020. *JavaScript: The Definitive Guide (7th Edition)*. Sebastopol, CA: O'Reilly Media. [ISBN: 978-1491952023]
 - Leventhal, Li., Julie Barnes, J. 2007. *Usability Engineering: Process, Products & Examples* (1st Edition). Essex, England: Pearson Education Publications. [ISBN: 978-0131570085]
-