**Lesson 06 Demo 02**

**Creating a React Redux Toolkit Rest API Cart Application**

**Objective:** To create a React application with Redux using the Thunk module to enable the management of a shopping cart from a JSON file

**Tools required:** Node.js and React.js

**Prerequisites:** None

Steps to be followed:

1. Create and set up the React project
2. Create the mobiles.json file
3. Create a features folder
4. Create a components folder
5. Add user-defined components in the App.js file
6. Configure the index.js file with store details
7. Test the application

**Step 1: Create and set up the React project**

* 1. Open a terminal window and run the following command to create a React application:

**npx create-react-app react-redux-toolkit-restapi-cart-app**



* 1. Open the created React application folder (**react-redux-toolkit-restapi-cart-app**) in VS Code by clicking on **File** in the top left corner and selecting **Open Folder**

**A screenshot of a computer

Description automatically generated**

* 1. Click on **Open** button

A screenshot of a computer

Description automatically generated

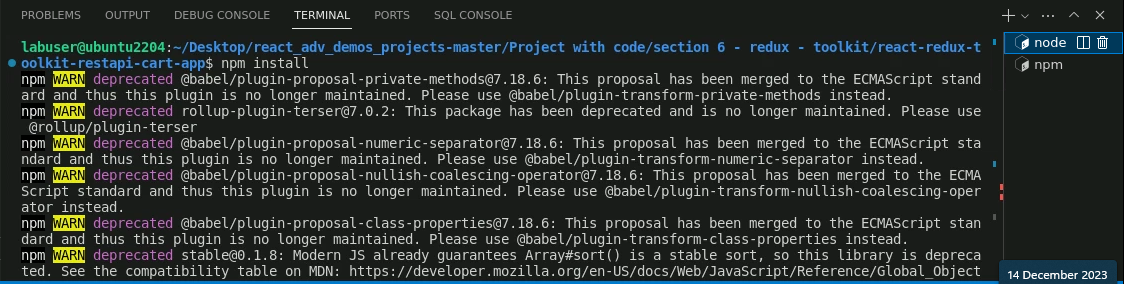
The folder structure appears as follows:

**A screenshot of a computer

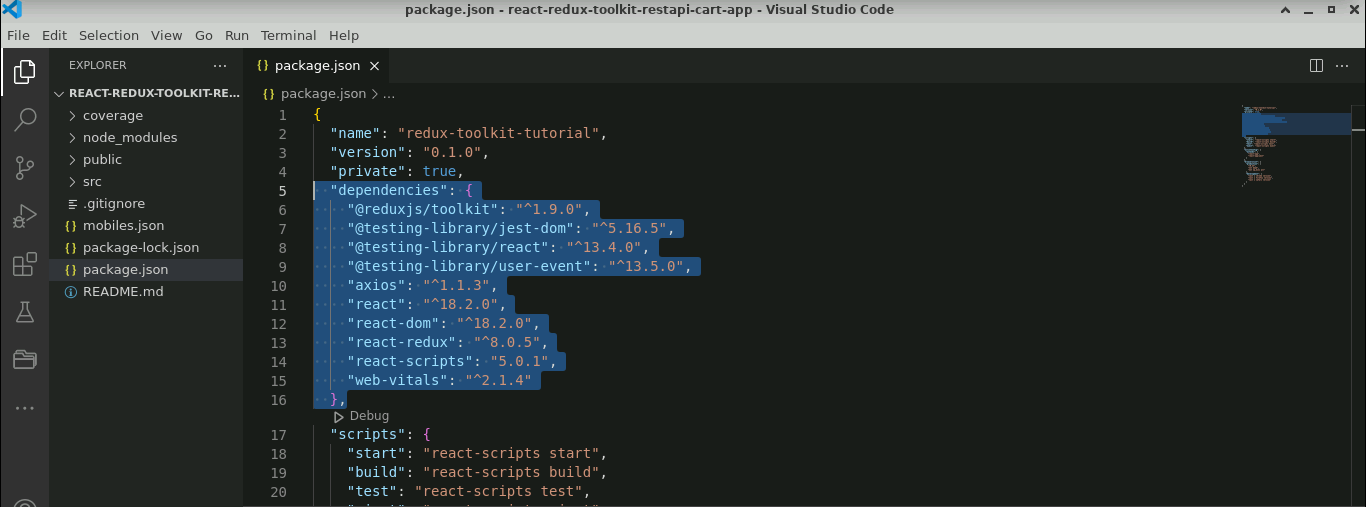
Description automatically generated**

* 1. Inside the project, open the **TERMINAL** and run the following command to install the required dependencies:

**npm install**

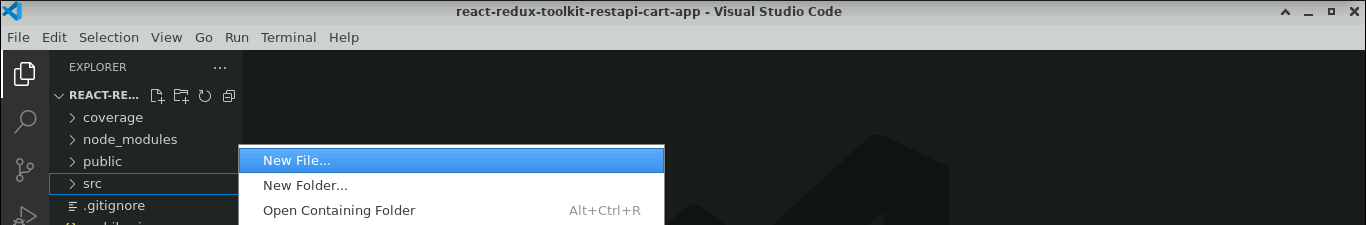
****

* 1. Open the **package.json** fileand view the external dependencies

****

**Step 2: Create the mobiles.json file**

1. Right-click on the **src** folder and select **New File** toenter the **mobiles.json**

****

**A computer screen shot of a black screen

Description automatically generated**

* 1. Inside the **mobiles.json** file, enter the following code:

**{**

**"mobiles":[**

**{"id":"rec1JZlfCIBOPdcT2","title":"Samsung Galaxy S8","price":"399.99","img":"https://www.course-api.com/images/cart/phone-1.png","amount":1}**

**,{"id":"recB6qcHPxb62YJ75","title":"google pixel","price":"499.99","img":"https://www.course-api.com/images/cart/phone-2.png","amount":1},**

**{"id":"recdRxBsE14Rr2VuJ","title":"Xiaomi Redmi Note 2","price":"699.99","img":"https://www.course-api.com/images/cart/phone-3.png","amount":1},**

**{"id":"recwTo160XST3PIoW","title":"Samsung Galaxy S7","price":"599.99 ","img":"https://www.course-api.com/images/cart/phone-4.png","amount":1}**

**]**

**}**

A screenshot of a computer

Description automatically generated

* 1. Install the **json-server** module by executing the command as shown below:

**npm install json-server -g**

**A screen shot of a computer

Description automatically generated**

* 1. Execute the following command to start the JSON server:

**json-server mobiles.json**

**A screenshot of a computer

Description automatically generated**

* 1. Run <http://localhost:3000/mobiles> (refer to the above screenshot) on the browser to get the Rest API

**A screenshot of a computer

Description automatically generated**

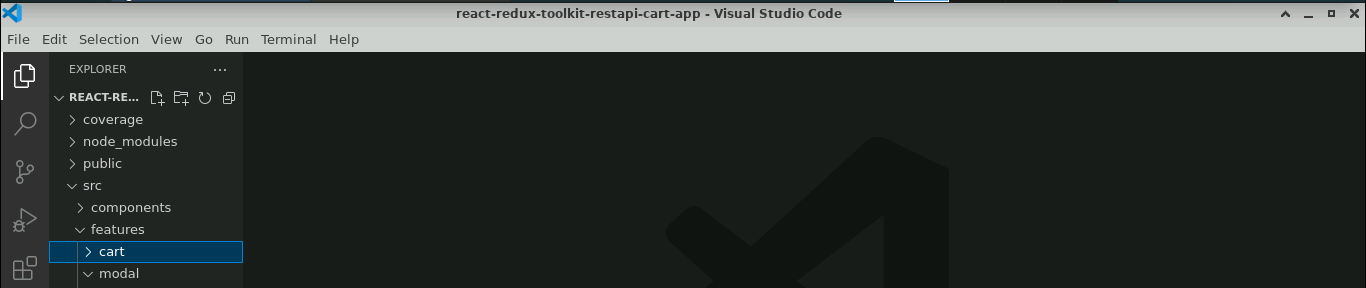
**Step 3: Create a features folder**

1. Right-click on the **src** folder and select **New File**

**A screenshot of a computer

Description automatically generated**

* 1. Create a folder named **features**, and create two folders named **cart** and **modal** within this folder

****

* 1. Inside the **cart** folder, create a file named **cartSlice.js** and enter the following code:

**import { createSlice, createAsyncThunk } from '@reduxjs/toolkit';**

**import axios from 'axios';**

**import { openModal } from '../modal/modalSlice';**

**const url = ' http://localhost:3000/mobiles';**

**const initialState = {**

**cartItems: [],**

**amount: 4,**

**total: 0,**

**isLoading: true,**

**};**

**export const getCartItems = createAsyncThunk(**

**'cart/getCartItems',**

**async (name, thunkAPI) => {**

**try {**

**// console.log(name);**

**// console.log(thunkAPI);**

**// console.log(thunkAPI.getState());**

**// thunkAPI.dispatch(openModal());**

**const resp = await axios(url);**

**return resp.data;**

**} catch (error) {**

**return thunkAPI.rejectWithValue('something went wrong');**

**}**

**}**

**);**

**const cartSlice = createSlice({**

**name: 'cart',**

**initialState,**

**reducers: {**

**clearCart: (state) => {**

**state.cartItems = [];**

**},**

**removeItem: (state, action) => {**

**const itemId = action.payload;**

**state.cartItems = state.cartItems.filter((item) => item.id !== itemId);**

**},**

**increase: (state, { payload }) => {**

**const cartItem = state.cartItems.find((item) => item.id === payload.id);**

**cartItem.amount = cartItem.amount + 1;**

**},**

**decrease: (state, { payload }) => {**

**const cartItem = state.cartItems.find((item) => item.id === payload.id);**

**cartItem.amount = cartItem.amount - 1;**

**},**

**calculateTotals: (state) => {**

**let amount = 0;**

**let total = 0;**

**state.cartItems.forEach((item) => {**

**amount += item.amount;**

**total += item.amount \* item.price;**

**});**

**state.amount = amount;**

**state.total = total;**

**},**

**},**

**extraReducers: (builder) => {**

**builder**

**.addCase(getCartItems.pending, (state) => {**

**state.isLoading = true;**

**})**

**.addCase(getCartItems.fulfilled, (state, action) => {**

**// console.log(action);**

**state.isLoading = false;**

**state.cartItems = action.payload;**

**})**

**.addCase(getCartItems.rejected, (state, action) => {**

**console.log(action);**

**state.isLoading = false;**

**});**

**},**

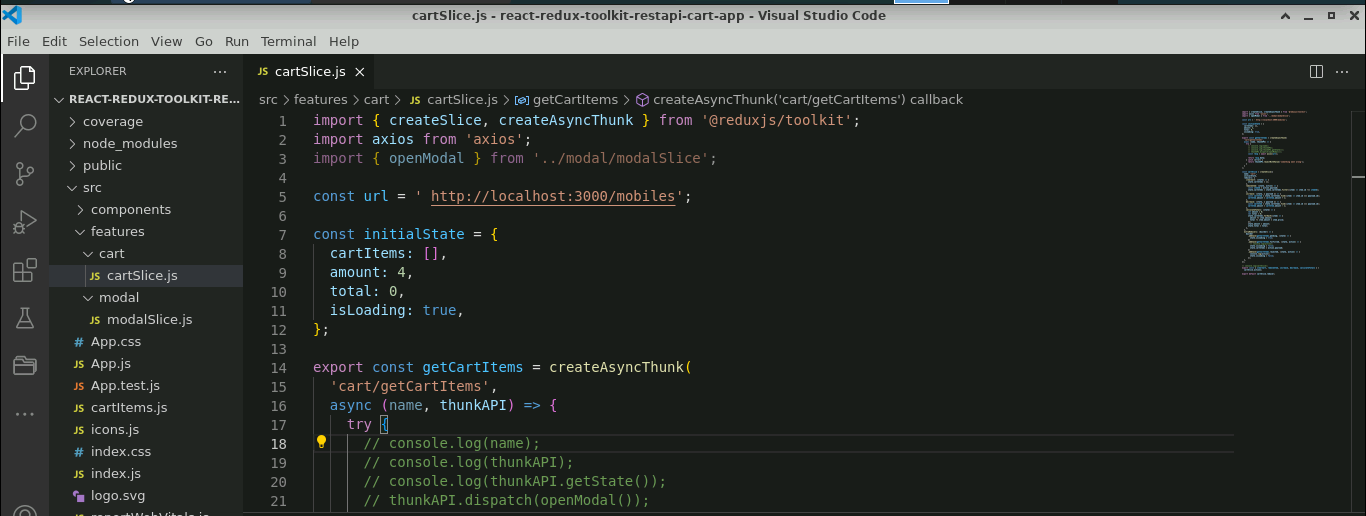
**});**

**// console.log(cartSlice);**

**export const { clearCart, removeItem, increase, decrease, calculateTotals } =**

**cartSlice.actions;**

**export default cartSlice.reducer;**

****

* 1. Inside the **modal** folder, create a file named **modalSlice.js** and enter the following code:

**import { createSlice } from '@reduxjs/toolkit';**

**const initialState = {**

**isOpen: false,**

**};**

**const modalSlice = createSlice({**

**name: 'modal',**

**initialState,**

**reducers: {**

**openModal: (state, action) => {**

**state.isOpen = true;**

**},**

**closeModal: (state, action) => {**

**state.isOpen = false;**

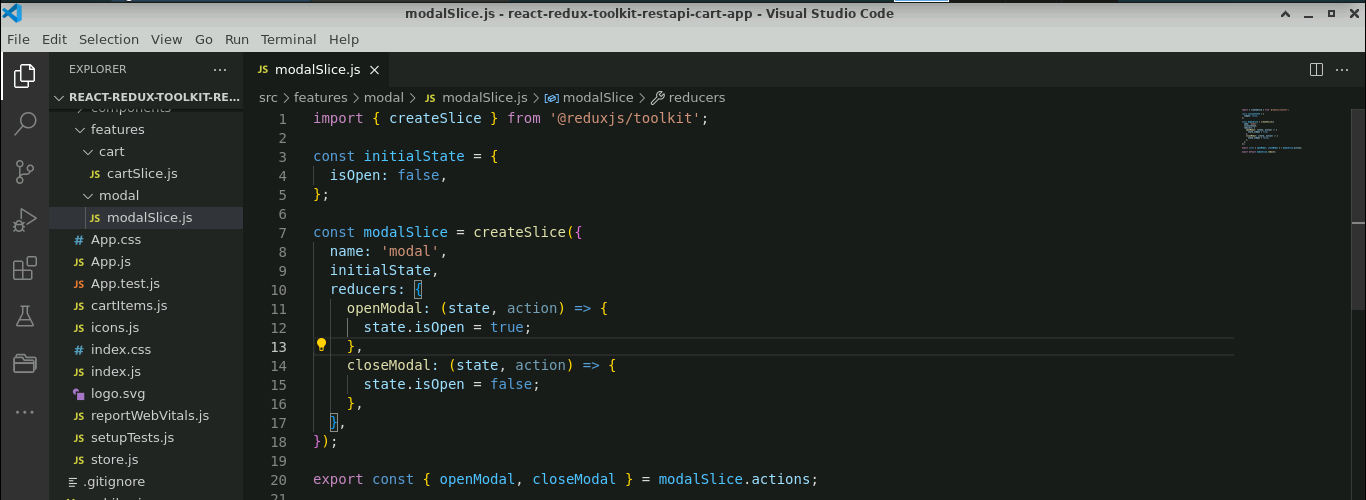
**},**

**},**

**});**

**export const { openModal, closeModal } = modalSlice.actions;**

**export default modalSlice.reducer;**

****

**Step 4: Create a components folder**

1. Create a folder named **components**, create a file named **CartContainer.js** with this folder, and enter the following code:

**import CartItem from './CartItem';**

**import { useDispatch, useSelector } from 'react-redux';**

**import { openModal } from '../features/modal/modalSlice';**

**const CartContainer = () => {**

**const dispatch = useDispatch();**

**const { cartItems, total, amount } = useSelector((store) => store.cart);**

**if (amount < 1) {**

**return (**

**<section className='cart'>**

**<header>**

**<h2>your bag</h2>**

**<h4 className='empty-cart'>is currently empty</h4>**

**</header>**

**</section>**

**);**

**}**

**return (**

**<section className='cart'>**

**<header>**

**<h2>your bag</h2>**

**</header>**

**<div>**

**{cartItems.map((item) => {**

**return <CartItem key={item.id} {...item} />;**

**})}**

**</div>**

**<footer>**

**<hr />**

**<div className='cart-total'>**

**<h4>**

**total <span>${total.toFixed(2)}</span>**

**</h4>**

**</div>**

**<button className='btn clear-btn' onClick={() => dispatch(openModal())}>**

**clear cart**

**</button>**

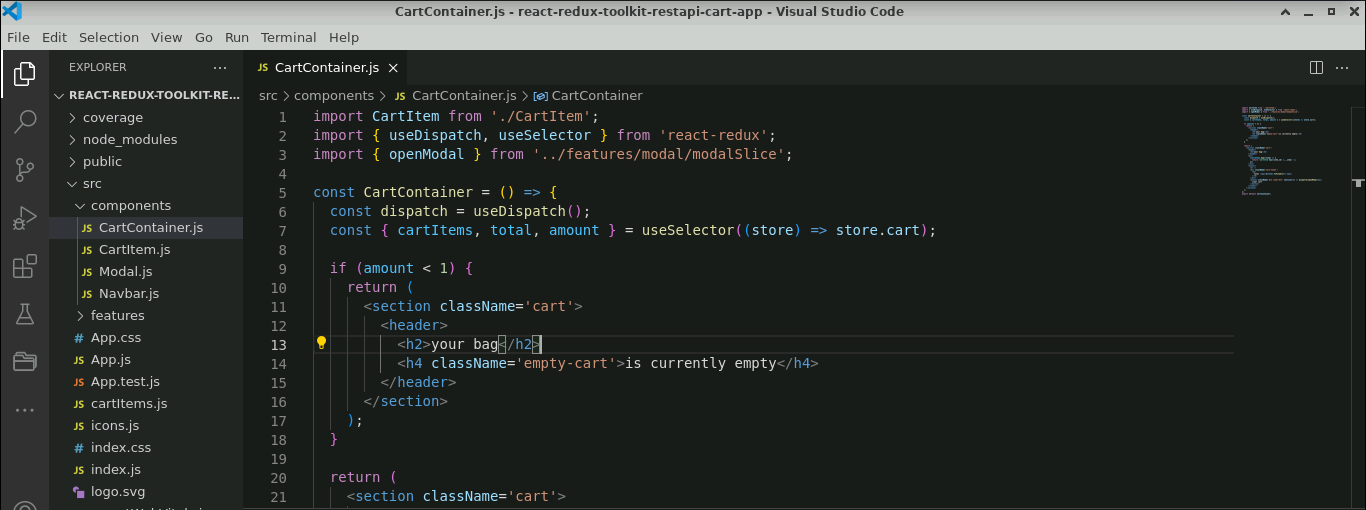
**</footer>**

**</section>**

**);**

**};**

**export default CartContainer;**

****

1. Create a file named **CartIteam.js** and enter the following code:

**import { ChevronDown, ChevronUp } from '../icons';**

**import { removeItem, increase, decrease } from '../features/cart/cartSlice';**

**import { useDispatch } from 'react-redux';**

**const CartItem = ({ id, img, title, price, amount }) => {**

**const dispatch = useDispatch();**

**return (**

**<article className='cart-item'>**

**<img src={img} alt={title} />**

**<div>**

**<h4>{title}</h4>**

**<h4 className='item-price'>${price}</h4>**

**<button**

**className='remove-btn'**

**onClick={() => {**

**dispatch(removeItem(id));**

**}}**

**>**

**remove**

**</button>**

**</div>**

**<div>**

**<button**

**className='amount-btn'**

**onClick={() => {**

**dispatch(increase({ id }));**

**}}**

**>**

**<ChevronUp />**

**</button>**

**<p className='amount'>{amount}</p>**

**<button**

**className='amount-btn'**

**onClick={() => {**

**if (amount === 1) {**

**dispatch(removeItem(id));**

**return;**

**}**

**dispatch(decrease({ id }));**

**}}**

**>**

**<ChevronDown />**

**</button>**

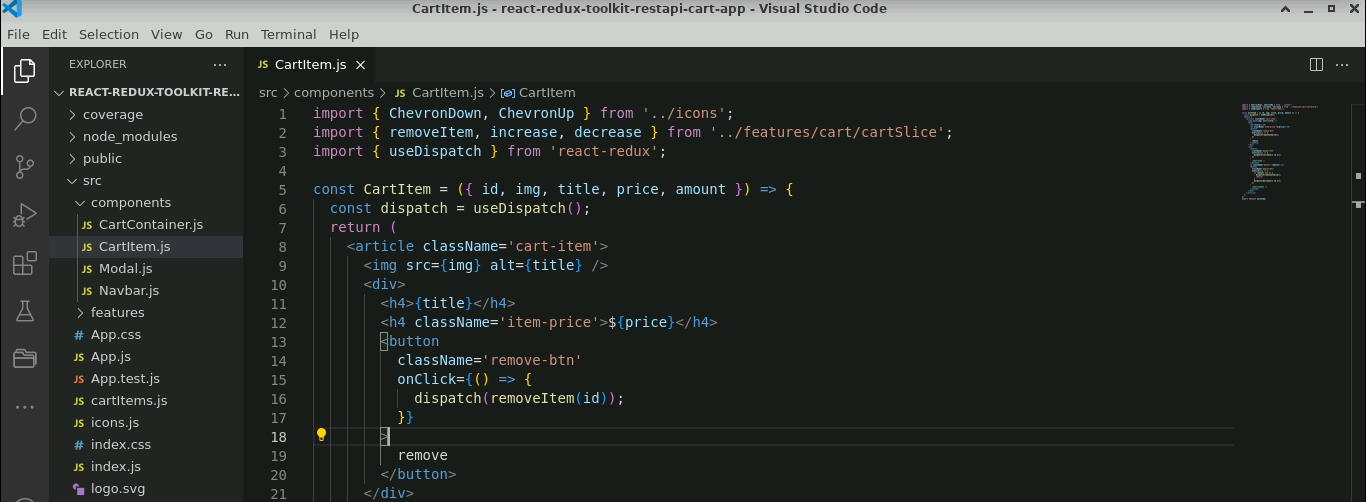
**</div>**

**</article>**

**);**

**};**

**export default CartItem;**

****

1. Create a file named **Modal.js** and enter the following code:

**import { useDispatch } from 'react-redux';**

**import { clearCart } from '../features/cart/cartSlice';**

**import { closeModal } from '../features/modal/modalSlice';**

**const Modal = () => {**

**const dispatch = useDispatch();**

**return (**

**<aside className='modal-container'>**

**<div className='modal'>**

**<h4>remove all items from your shopping cart?</h4>**

**<div className='btn-container'>**

**<button**

**type='button'**

**className='btn confirm-btn'**

**onClick={() => {**

**dispatch(clearCart());**

**dispatch(closeModal());**

**}}**

**>**

**confirm**

**</button>**

**<button**

**type='button'**

**className='btn clear-btn'**

**onClick={() => {**

**dispatch(closeModal());**

**}}**

**>**

**cancel**

**</button>**

**</div>**

**</div>**

**</aside>**

**);**

**};**

**export default Modal;**

**A screen shot of a computer program

Description automatically generated**

1. Create a file named **Navbar.js** and enter the following code:

**import { CartIcon } from '../icons';**

**import { useSelector } from 'react-redux';**

**const Navbar = () => {**

**const { amount } = useSelector((store) => store.cart);**

**return (**

**<>**

**<nav>**

**<div className='nav-center'>**

**<h3>redux toolkit</h3>**

**<div className='nav-container'>**

**<CartIcon />**

**<div className='amount-container'>**

**<p className='total-amount'>{amount}</p>**

**</div>**

**</div>**

**</div>**

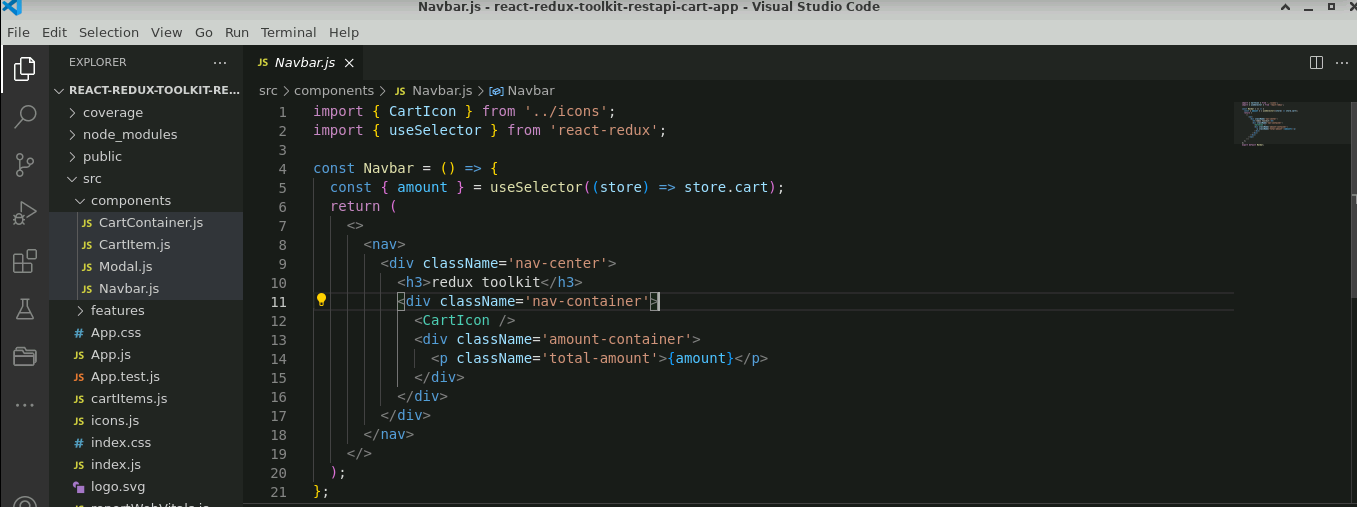
**</nav>**

**</>**

**);**

**};**

**export default Navbar;**

****

**Step 5: Add user-defined components in the App.js file**

* 1. In the **App.js** file, add **Navbar.js** and **Modal.js** components as shown below:

**import Navbar from './components/Navbar';**

**import CartContainer from './components/CartContainer';**

**import { useDispatch, useSelector } from 'react-redux';**

**import { calculateTotals, getCartItems } from './features/cart/cartSlice';**

**import { useEffect } from 'react';**

**import Modal from './components/Modal';**

**function App() {**

**const { cartItems, isLoading } = useSelector((store) => store.cart);**

**const { isOpen } = useSelector((store) => store.modal);**

**const dispatch = useDispatch();**

**useEffect(() => {**

**dispatch(calculateTotals());**

**}, [cartItems]);**

**useEffect(() => {**

**dispatch(getCartItems('random'));**

**}, []);**

**if (isLoading) {**

**return (**

**<div className='loading'>**

**<h1>Loading...</h1>**

**</div>**

**);**

**}**

**return (**

**<main>**

**{isOpen && <Modal />}**

**<Navbar />**

**<CartContainer />**

**</main>**

**);**

**}**

**export default App;**

**A screen shot of a computer

Description automatically generated**

**Step 6: Configure the index.js file with store details**

* 1. In the **index.js** file, create a **store** and configure the **reducer** details as shown below:

**import React from 'react';**

**import { createRoot } from 'react-dom/client';**

**import './index.css';**

**import App from './App';**

**import { store } from './store';**

**import { Provider } from 'react-redux';**

**const container = document.getElementById('root');**

**const root = createRoot(container);**

**root.render(**

**<React.StrictMode>**

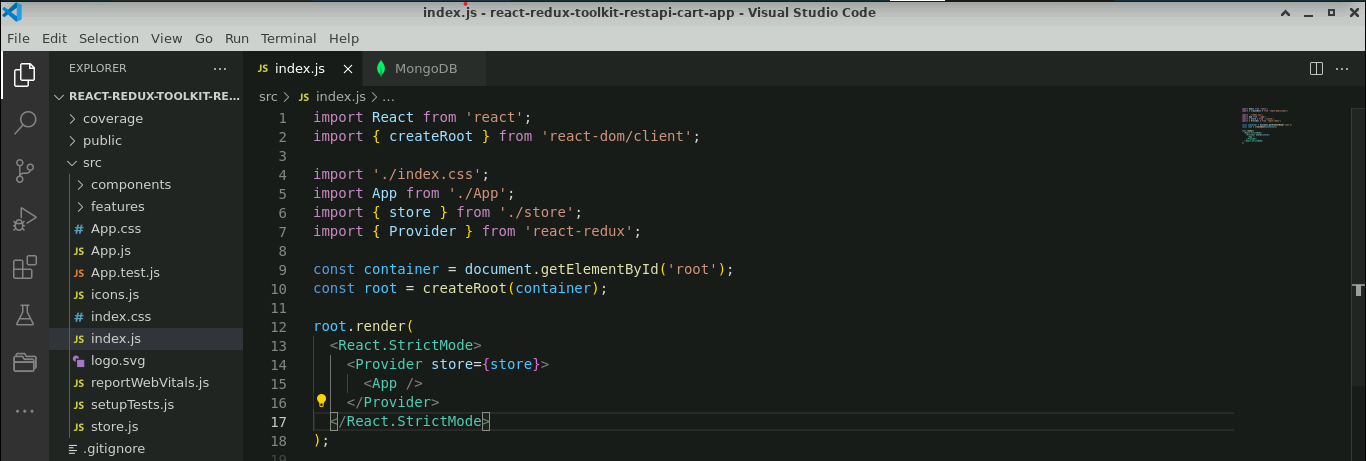
**<Provider store={store}>**

**<App />**

**</Provider>**

**</React.StrictMode>**

**);**

****

**Step 7: Test the application**

* 1. Open the terminal and run the below command to execute the application:

**npm start**

**A screen shot of a computer

Description automatically generated**

**A screen shot of a computer

Description automatically generated**

**Note:** It may ask to store the application in another port number.

So, **Would you like to run the app on another port instead? (y/n**) type y

The output appears as shown below:

A screenshot of a computer

Description automatically generated

* 1. To remove all the items from the cart, click on **CLEAR CART** from the above screenshot and click on **CONFIRM**

A screenshot of a computer

Description automatically generated

The output appears as shown below:

A screenshot of a computer

Description automatically generated

With this, you have successfully created a React application with Redux, using the Thunk module. It enables managing a shopping cart from JSON files and displaying the contents of the shopping cart within a web application.