**Lesson 06 Demo 02**

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

**Objective:** To demonstrate the react with redux using thunk module retrieve mobile details from json file. Increment as well as decrement quantity and view the total cost.

**Tools required:** Node JS and React JS

**Prerequisites:** HTML, CSS, JavaScript ES5/ES6, Basic React Concept

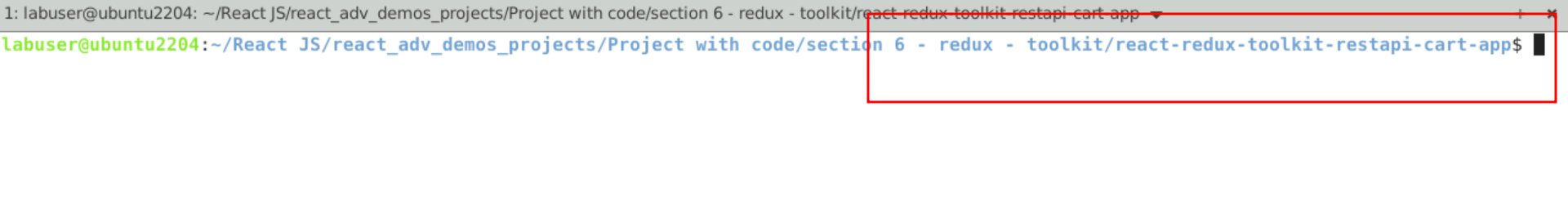
**Note** : All react js project already created with version 18.x with Sample App.js file

**Steps to be followed:**

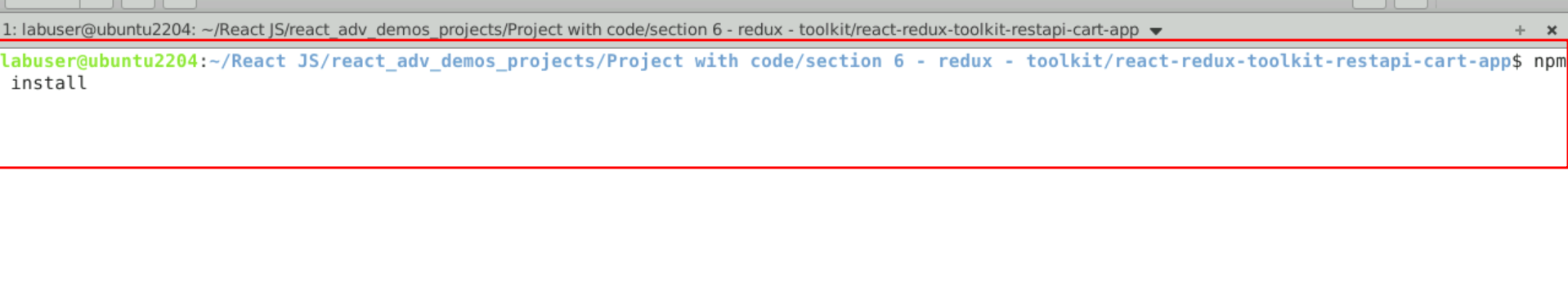
1. Set up for react js project
2. Create mobiles.json file which contains few static mobiles details in the form of json file.
3. Create features folder which contains two folder ie cart and modal.
4. Now create components folder which contains which contains more than one component.
5. Use all user defined component in App.js file
6. In index.js file configure store details.
7. Test the application.

**Step 1: Set up for react js project**

1. Open a terminal window inside a React JS pre-created project ie **react-redux-toolkit-restapi-cart-app**

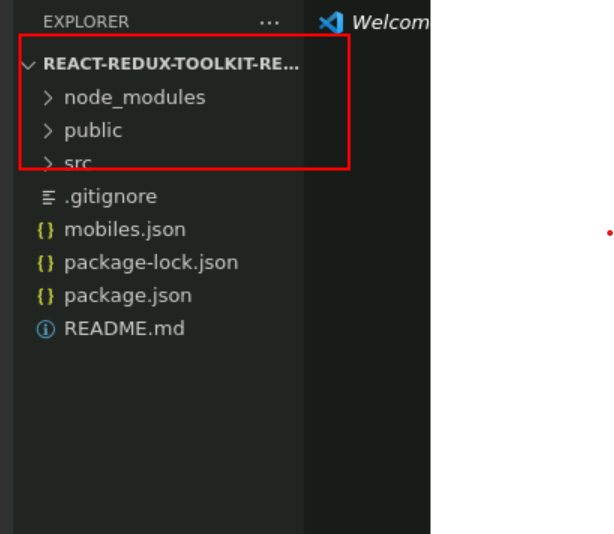


1. Now you need to run the command as **npm install.** This command helps us to installed all required dependencies mention in package.json file in local machine in the form of node\_module folder.

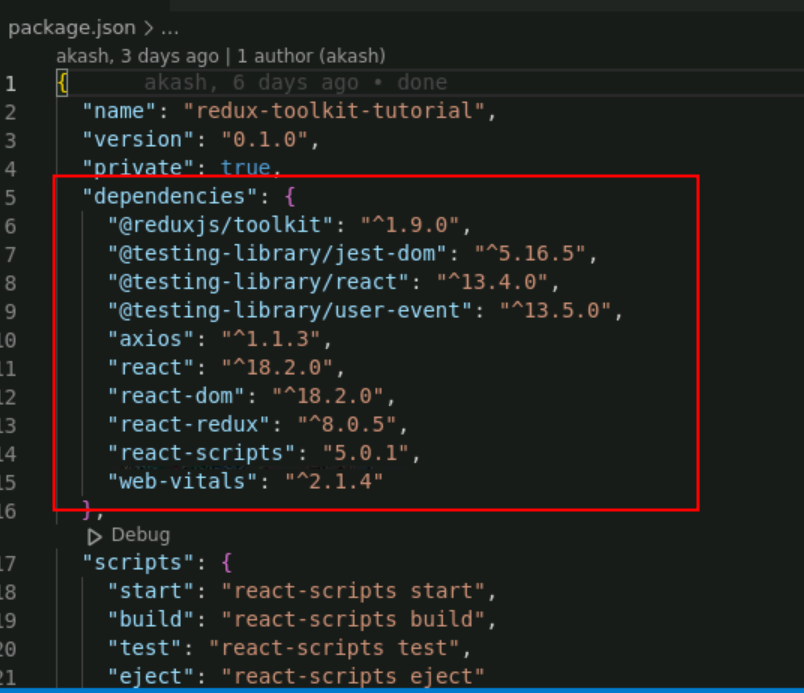


1. Now open **react-redux-toolkit-restapi-cart-app** folder in VS Code Editor

Note: short cut to open write **code .**

****

1.4 now open package.json file and view external dependencies.

****

**Step 2 : Create mobiles.json file which contains few static mobiles details in the form of json file.**

2.1 create the mobiles.json file which contains static json data.

**mobiles.json**

{

"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},

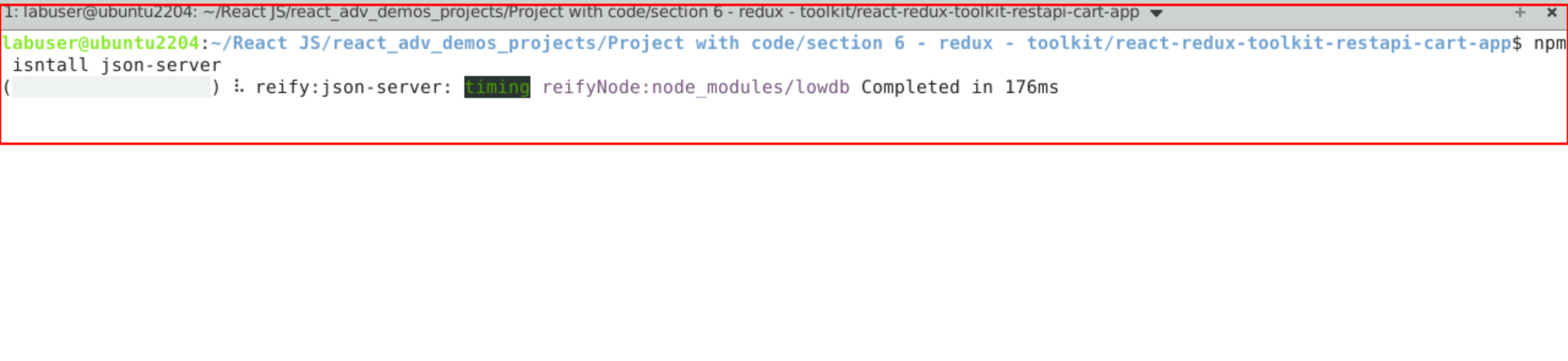
{"id":"recwTo160XST3PIoW","title":"Samsung Galaxy S7","price":"599.99 ","img":"https://www.course-api.com/images/cart/phone-4.png","amount":1}

]

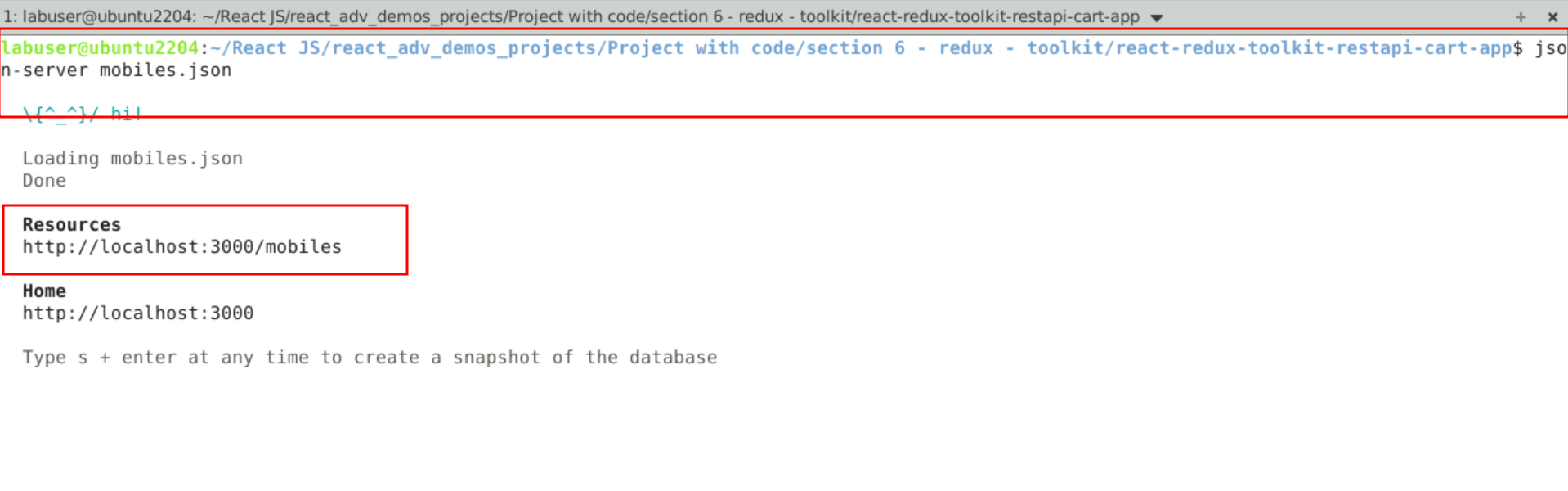
}

2.2 We need to run this file using the node module as **json-server**. So first install json-server module using command as **npm install json-server -g**

Open the folder inside that location **mobiles.json** file present.



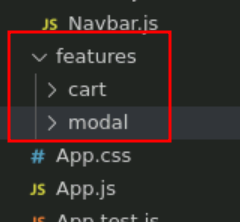
Now you can run the command as json-server mobiles.json file which provide rest api.



So here <http://localhost:3000/mobiles> URL provide REST API. You can add and retrieve operation through react JS.

**Step 3:** Create features folder which contains two folder ie cart and modal.

3.1 create features folder with sub folder as cart and modal inside src folder



3.2 inside features folder contains modal folder with modelSlice.js file. This file take the help of redux toolkit to createSlice which contains name, intialState and reducer to open and close the modal.

**modalSlice.js**

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;

3.3 Inside features folder contains cart folder with cartSlice.js file. This file take the help of redux toolkit to createSlice which contains name, intialState and reducer with clearCart, removeCart, increase and decrement qty, calcuateTotal with createAsyncThunk with axios modules.

**cartSlice.js**

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;

**Step 4 : Now create components folder which contains which contains more than one components.**

4.1 Create Modal.js fille which is take the help of useDispatch hook to call reducer toolkit clearCar and closeModel functions.

**Model.js**

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;

4.2 Create CartContainer.js fille which is take the help of useDispatch hook to open the Modal and useSelector hook to access the mobile details.

**CartContainer.js**

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;

4.3 Create Cartitem.js fille which is responsible to display all cart details retrieve from cartContainer.js component.

**CartItem.js**

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;

4.4 Create Navbar.js component responsible to do the total amount calculation.

**Navbar.js**

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 : Use all user defined component in App.js file**

5.1 In App.js file add Navbar.js and Modal.js components. Take the help of useSelector, useDispatch, useEffect hook to develop the code.

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;

**Step 6 : In index.js file configure store details.**

6.1 In index.js file we need to create store and configure reducer details

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 />

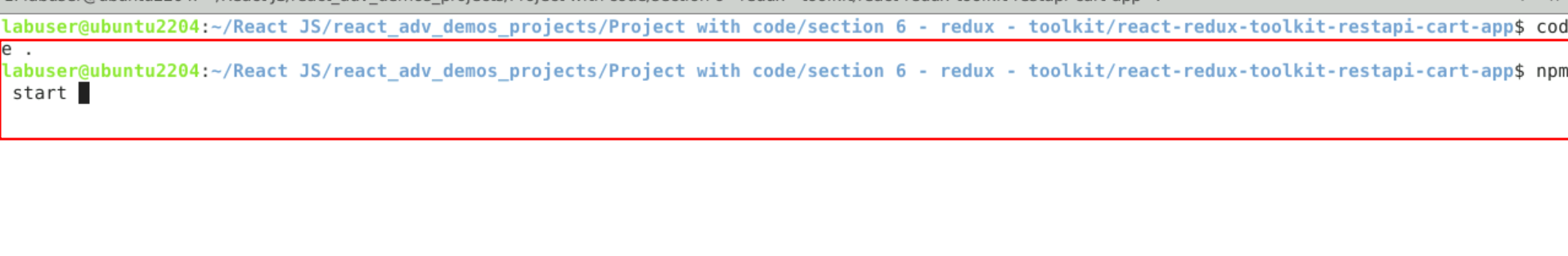
</Provider>

</React.StrictMode>

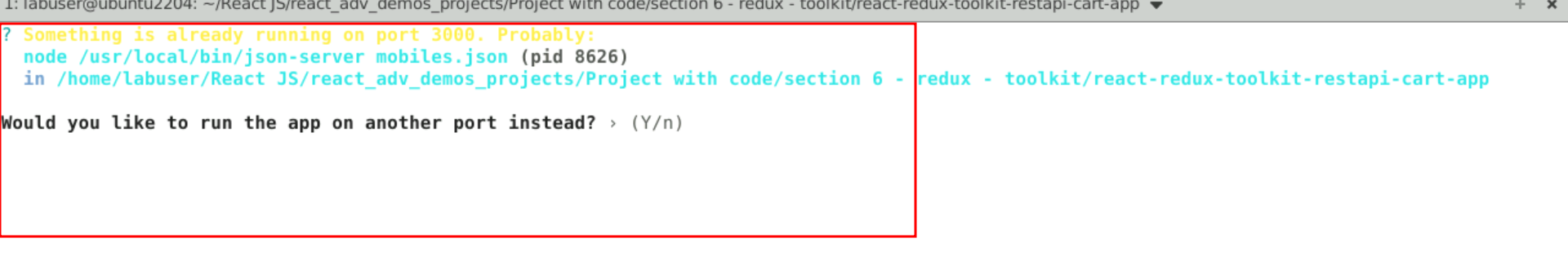
);

**Step 7 Now we run the application using npm start**

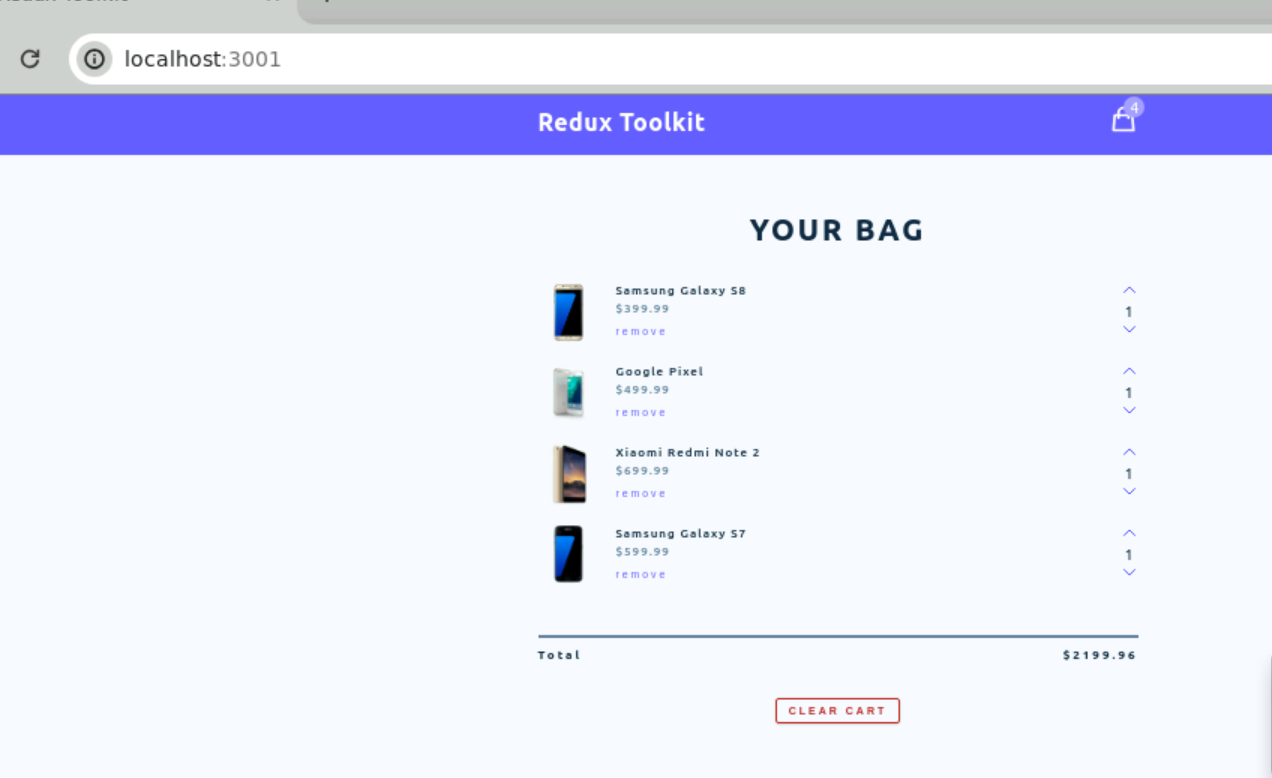
**7.1** Now run the application usiing command as **npm start**



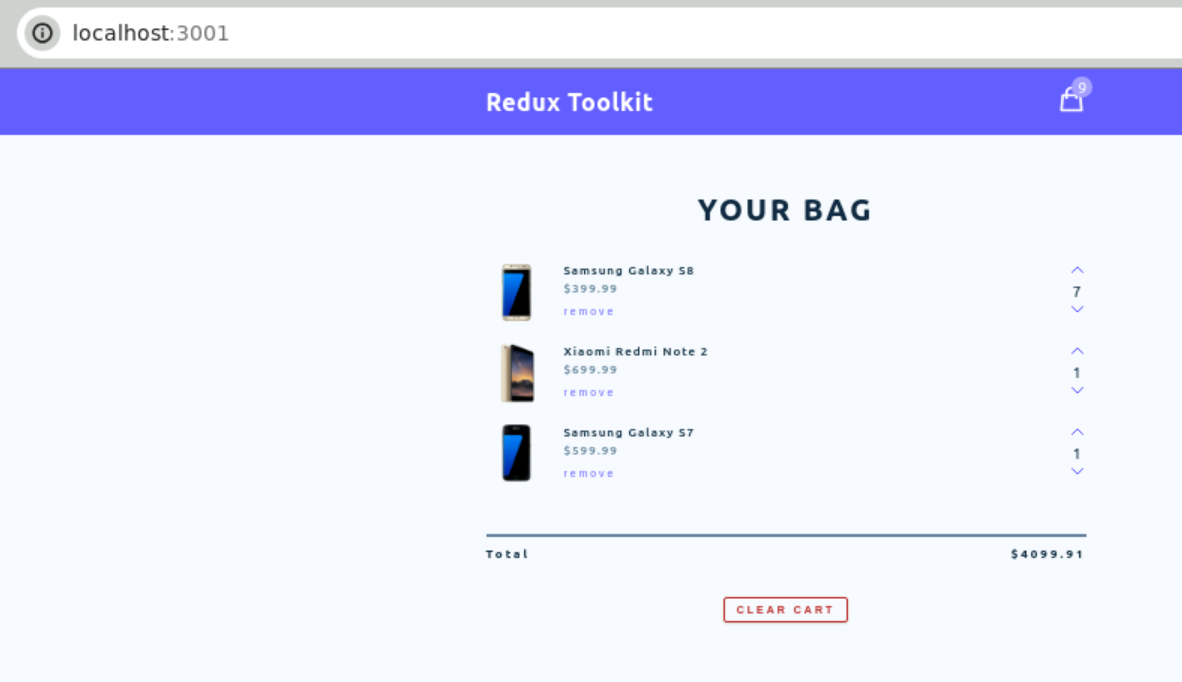
**7.2** It may ask you to store this application in another port number. Because by default json-server also run on default port number 3000 as well as react js also run on default port number 3000. So already json-server running employee.json file on port number 3000. So give yes it will run on 3001 port number.



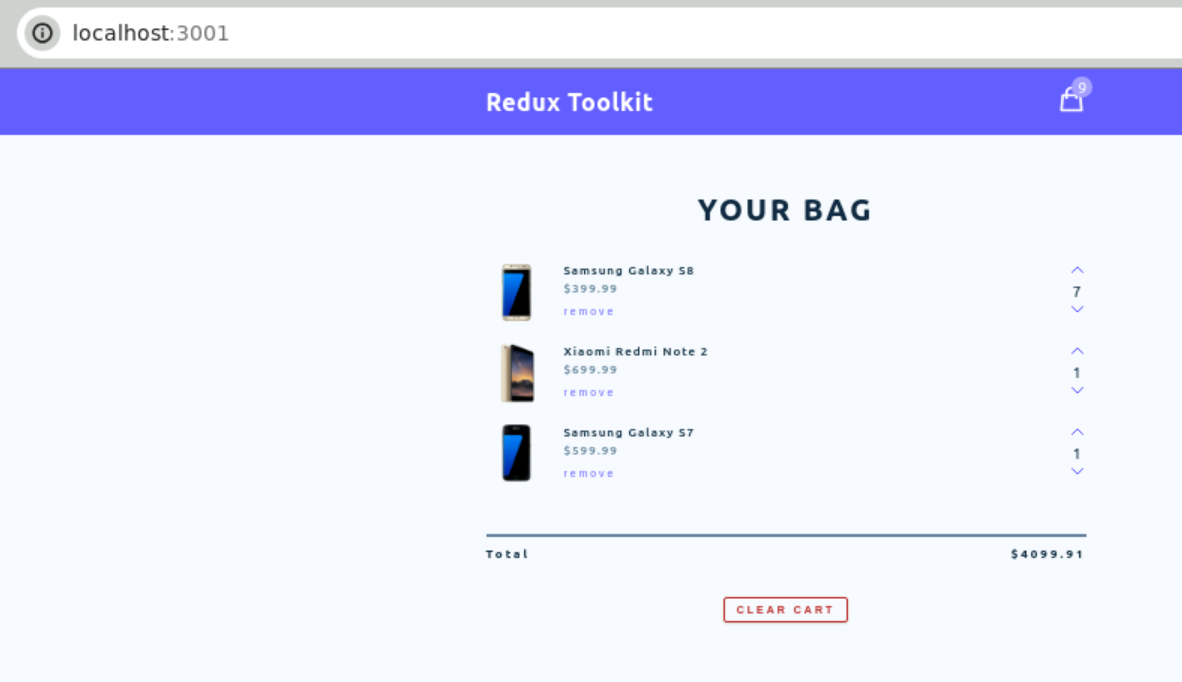
7.4 after running this application on browse



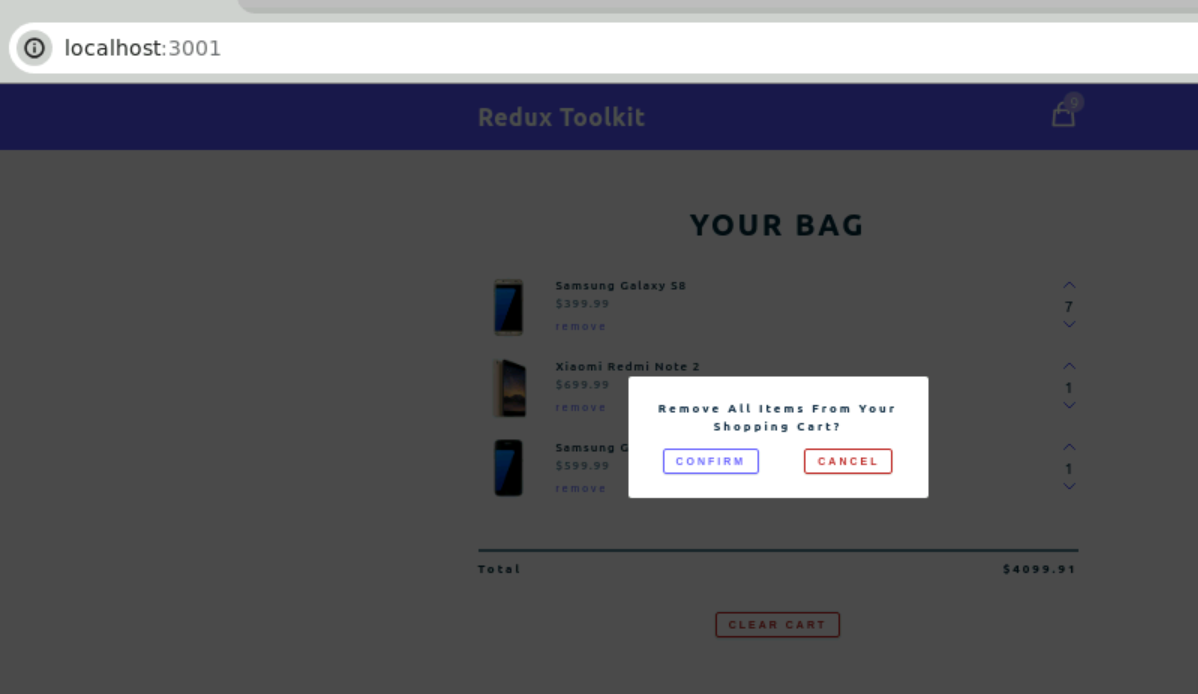
7.5 You can increment qty using up arrow as well as decrement qty using below arrow.



7.6 you can see the total number of qty in cart as well as total amount.



7.7 you can clear the cart by clicking clear cart button.



7.8 after removed the cart

