**Lesson 08 Demo 01**

**Implementing CRUD Operations Using JSON Server API**

**Objective:** To implement a CRUD operation in React application using JSON server API

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

**Prerequisites:** None

Steps to be followed:

1. Create and set up the React project
2. Create the database
3. Create the components file
4. Configure the Redux
5. Modify the index.js and App.js file
6. Run the application

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

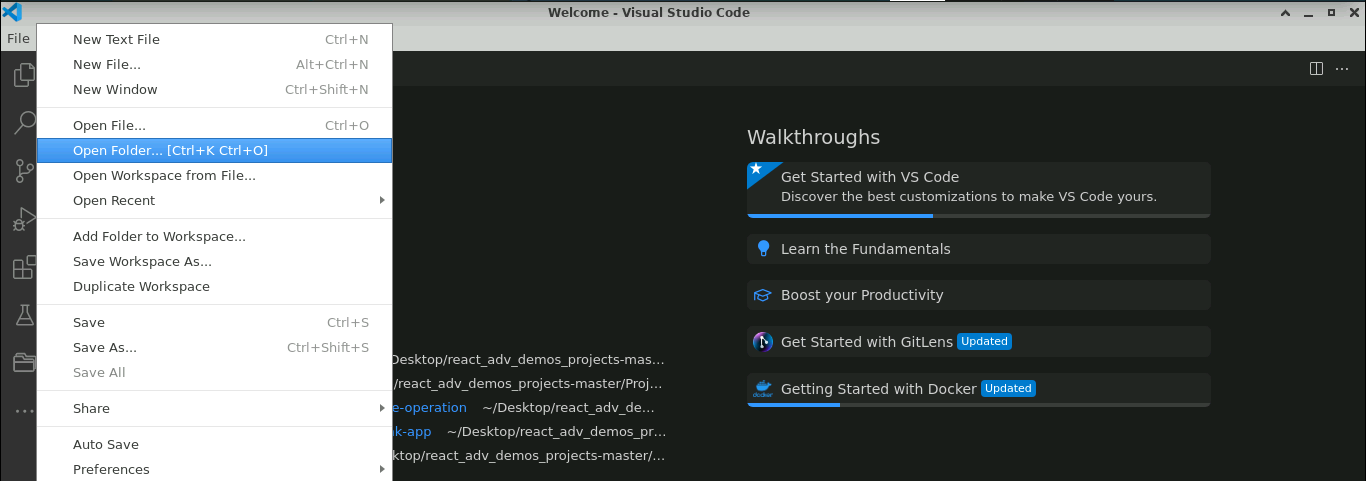
1. Open a terminal window and run the following command to create a React application:

**npx create-react-app reduxcrudoperation**

**A screenshot of a computer

Description automatically generated**

1. Open the created React application folder (**reduxcrudoperation**) in VS Code by clicking on **File** in the top left corner and selecting **Open Folder**

****

1. Click on **Open** button

**A screenshot of a computer

Description automatically generated**

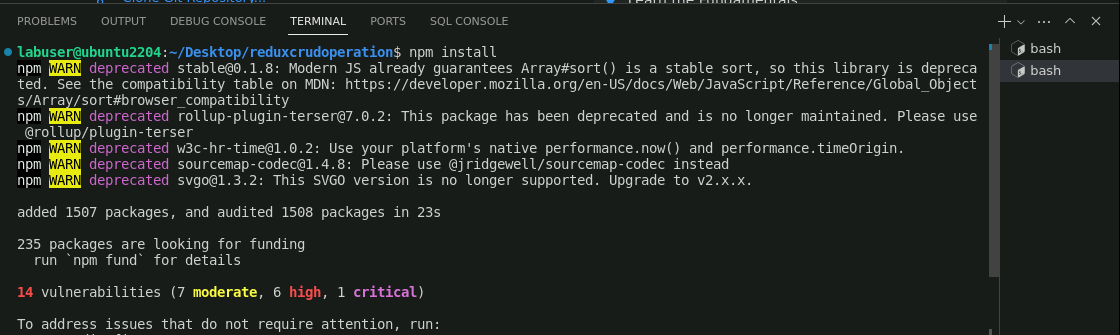
The folder structure appears as follows:

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**

****

**Note**: This command helps you install all the required dependencies mentioned in **package.json** file on the local machine in the form of a **node\_module** folder.

1. Run the following command to install the Bootstrap dependencies:

**npm install bootstrap**

**A screen shot of a computer

Description automatically generated**

1. Run the following command to install the React router dependencies:

**npm install react-router-dom**

**A screen shot of a computer

Description automatically generated**

1. Run the following command to install the React notification dependencies:

**npm i react-toastify**

**A screen shot of a computer

Description automatically generated**

1. Run the following command to install the React Redux dependencies:

**npm i redux react-redux redux-thunk axios redux-logger @reduxjs/toolkit**

**A screen shot of a computer

Description automatically generated**

1. Open the **package.json** file to view the dependencies

A screenshot of a computer

Description automatically generated

**Step 2: Create the database**

* 1. Inside the **src** folder, create a file named **Data** and run the following command to create a **db.json** file:

**json-server --watch src/Data/db.json --port 8000**

**A screen shot of a computer

Description automatically generated**

**A screen shot of a computer

Description automatically generated**

* 1. Modify the **db.json** file by adding the user details

A screenshot of a computer

Description automatically generated

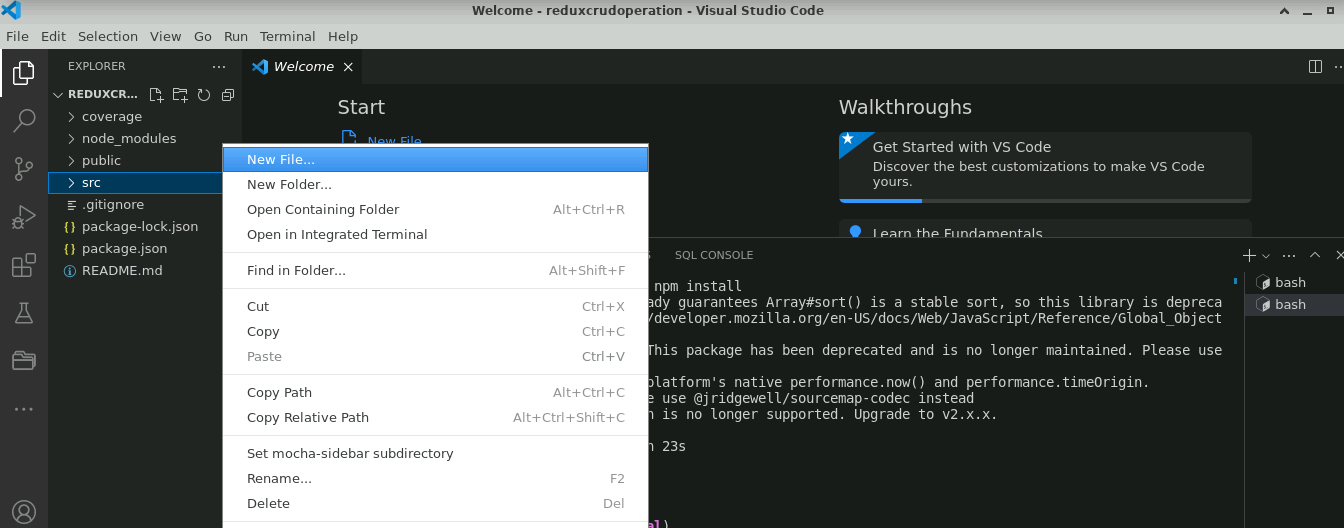
* 1. Click on [**http://localhost:8000/user**](http://localhost:8000/user) to view the user details in the database

A blue line on a white background

Description automatically generated

**Step 3: Create the components file**

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

****

* 1. Create the **Component** folder inside the **src** folder, then inside it, create a **Home.js** file, and enter the following code:

**const Home = () => {**

**return (**

**<div>**

**<h2>React JS CRUD using REDUX & JSON Server</h2>**

**</div>**

**);**

**}**

**export default Home;**

**A screenshot of a computer

Description automatically generated**

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

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

**import { connect } from "react-redux";**

**import { Link } from "react-router-dom";**

**import { toast } from "react-toastify";**

**import { FetchUserList,Removeuser} from "../Redux/Action";**

**const Userlisting = (props) => {**

**useEffect(() => {**

**props.loaduser();**

**}, [])**

**const handledelete = (code) => {**

**if (window.confirm('Do you want to remove?')) {**

**props.removeuser(code);**

**props.loaduser();**

**toast.success('User removed successfully.')**

**}**

**}**

**return (**

**props.user.loading ? <div><h2>Loading...</h2></div> :**

**props.user.errmessage ? <div><h2>{props.user.errmessage}</h2></div> :**

**<div>**

**<div className="card">**

**<div className="card-header" >**

**<Link to={'/user/add'} className="btn btn-success">Add User [+]</Link>**

**</div>**

**<div className="card-body">**

**<table className="table table-bordered">**

**<thead className="bg-dark text-white">**

**<tr>**

**<td>Code</td>**

**<td>Name</td>**

**<td>Email</td>**

**<td>Phone</td>**

**<td>Role</td>**

**<td>Action</td>**

**</tr>**

**</thead>**

**<tbody>**

**{**

**props.user.userlist && props.user.userlist.map(item =>**

**<tr key={item.id}>**

**<td>{item.id}</td>**

**<td>{item.name}</td>**

**<td>{item.email}</td>**

**<td>{item.phone}</td>**

**<td>{item.role}</td>**

**<td>**

**<Link to={'/user/edit/' + item.id} className="btn btn-primary">Edit</Link> |**

**<button onClick={() => { handledelete(item.id) }} className="btn btn-danger">Delete</button>**

**</td>**

**</tr>**

**)**

**}**

**</tbody>**

**</table>**

**</div>**

**</div>**

**</div>**

**);**

**}**

**const mapStateToProps = (state) => {**

**return {**

**user: state.user**

**}**

**}**

**const mapDispatchToProps = (dispatch) => {**

**return {**

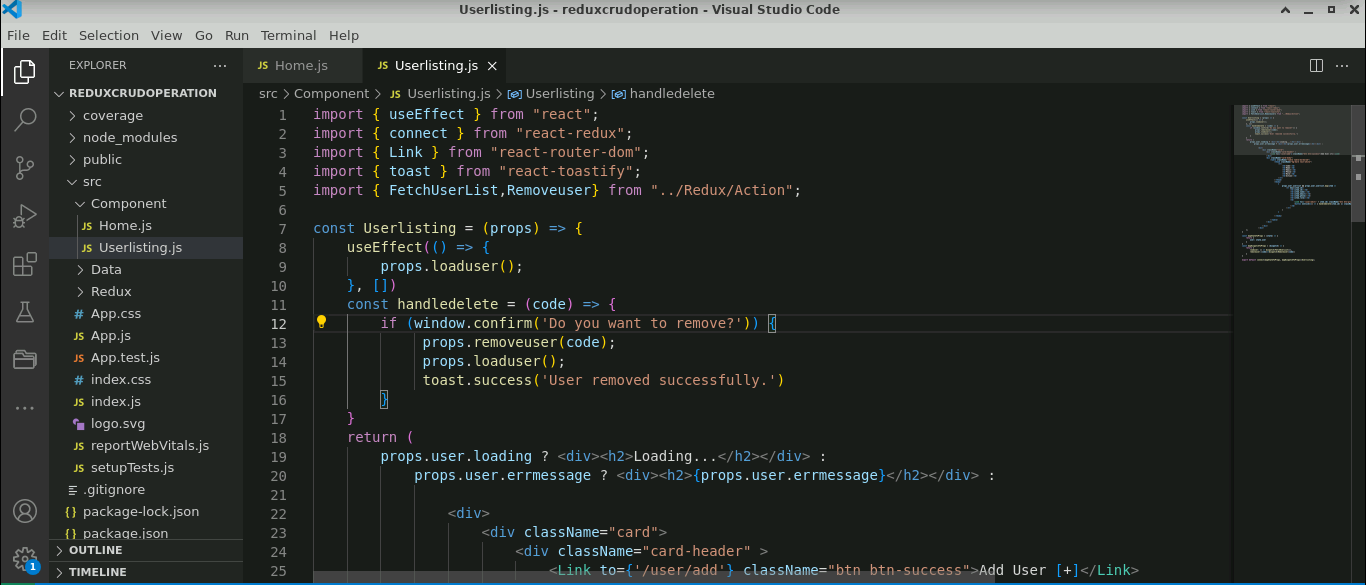
**loaduser: () => dispatch(FetchUserList()),**

**removeuser:(code)=>dispatch(Removeuser(code))**

**}**

**}**

**export default connect(mapStateToProps, mapDispatchToProps)(Userlisting);**

****

* 1. Create an **Addduser.js** file and enter the following code:

**import { useState } from "react";**

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

**import { Link, useNavigate } from "react-router-dom";**

**import { FunctionAddUser } from "../Redux/Action";**

**const Adduser = () => {**

**const [name, namechange] = useState('');**

**const [email, emailchange] = useState('');**

**const [phone, phonechange] = useState('');**

**const [role, rolechange] = useState('staff');**

**const dispatch=useDispatch();**

**const navigate=useNavigate();**

**const handlesubmit = (e) => {**

**e.preventDefault();**

**const userobj = { name, email, phone, role };**

**dispatch(FunctionAddUser(userobj));**

**navigate('/user');**

**}**

**return (**

**<div>**

**<form onSubmit={handlesubmit}>**

**<div className="card">**

**<div className="card-header" style={{ textAlign: 'left' }}**

**<h2>Add User</h2>**

**</div>**

**<div className="card-body" style={{ textAlign: 'left' }}>**

**<div className="row">**

**<div className="col-lg-12">**

**<div className="form-group">**

**<label>Name</label>**

**<input value={name} onChange={e => namechange(e.target.value)} className="form-control"></input>**

**</div>**

**</div>**

**<div className="col-lg-12">**

**<div className="form-group">**

**<label>Email</label>**

**<input value={email} onChange={e => emailchange(e.target.value)} className="form-control"></input>**

**</div>**

**</div>**

**<div className="col-lg-12">**

**<div className="form-group">**

**<label>Phone</label>**

**<input value={phone} onChange={e => phonechange(e.target.value)} className="form-control"></input>**

**</div>**

**</div>**

**<div className="col-lg-12">**

**<div className="form-group">**

**<label>Role</label>**

**<select value={role} onChange={e => rolechange(e.target.value)} className="form-control">**

**<option value="admin">Admin</option>**

**<option value="staff">Staff</option>**

**</select>**

**</div>**

**</div>**

**</div>**

**</div>**

**<div className="card-footer" style={{ textAlign: 'left' }}>**

**<button className="btn btn-primary" type="submit">Submit</button> |**

**<Link className="btn btn-danger" to={'/user'}>Back</Link>**

**</div>**

**</div>**

**</form>**

**</div>**

**);**

**}**

**export default Adduser;**

**A screenshot of a computer screen

Description automatically generated**

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

**import { useEffect, useState } from "react";**

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

**import { Link, useNavigate, useParams } from "react-router-dom";**

**import { FetchUserObj, FunctionUpdateUser } from "../Redux/Action";**

**const Updateuser = () => {**

**const [id, idchange] = useState(0);**

**const [name, namechange] = useState('');**

**const [email, emailchange] = useState('');**

**const [phone, phonechange] = useState('');**

**const [role, rolechange] = useState('staff');**

**const dispatch = useDispatch();**

**const navigate = useNavigate();**

**const { code } = useParams();**

**const userobj=useSelector((state)=>state.user.userobj)**

**const handlesubmit = (e) => {**

**e.preventDefault();**

**const userobj = { id, name, email, phone, role };**

**dispatch(FunctionUpdateUser(userobj,id));**

**navigate('/user');**

**}**

**useEffect(() => {**

**dispatch(FetchUserObj(code));**

**}, [])**

**useEffect(() => {**

**if(userobj){**

**idchange(userobj.id);**

**namechange(userobj.name);**

**emailchange(userobj.email);**

**phonechange(userobj.phone);**

**rolechange(userobj.role);**

**}**

**}, [userobj])**

**return (**

**<div>**

**<form onSubmit={handlesubmit}>**

**<div className="card">**

**<div className="card-header" style={{ textAlign: 'left' }}>**

**<h2>Add User</h2>**

**</div>**

**<div className="card-body" style={{ textAlign: 'left' }}>**

**<div className="row">**

**<div className="col-lg-12">**

**<div className="form-group">**

**<label>Id</label>**

**<input value={id || ''} disabled="disabled" className="form-control"></input>**

**</div>**

**</div>**

**<div className="col-lg-12">**

**<div className="form-group">**

**<label>Name</label>**

**<input value={name || ''} onChange={e => namechange(e.target.value)} className="form-control"></input>**

**</div>**

**</div>**

**<div className="col-lg-12">**

**<div className="form-group">**

**<label>Email</label>**

**<input value={email || ''} onChange={e => emailchange(e.target.value)} className="form-control"></input>**

**</div>**

**</div>**

**<div className="col-lg-12">**

**<div className="form-group">**

**<label>Phone</label>**

**<input value={phone || ''} onChange={e => phonechange(e.target.value)} className="form-control"></input>**

**</div>**

**</div>**

**<div className="col-lg-12">**

**<div className="form-group">**

**<label>Role</label>**

**<select value={role || ''} onChange={e => rolechange(e.target.value)} className="form-control">**

**<option value="admin">Admin</option>**

**<option value="staff">Staff</option>**

**</select>**

**</div>**

**</div>**

**</div>**

**</div>**

**<div className="card-footer" style={{ textAlign: 'left' }}>**

**<button className="btn btn-primary" type="submit">Submit</button> |**

**<Link className="btn btn-danger" to={'/user'}>Back</Link>**

**</div>**

**</div>**

**</form>**

**</div>**

**);**

**}**

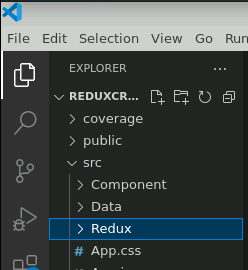
**export default Updateuser;**

**A screenshot of a computer program

Description automatically generated**

**Step 4: Configure the Redux**

1. Inside the **src** folder, create a folder named **Redux**

****

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

**import axios from "axios"**

**import { toast } from "react-toastify"**

**import Updateuser from "../Component/Updateuser"**

**import { ADD\_USER, DELETE\_USER, FAIL\_REQUEST, GET\_USER\_LIST, GET\_USER\_OBJ, MAKE\_REQUEST, UPDATE\_USER } from "./ActionType"**

**export const makeRequest=()=>{**

**return{**

**type:MAKE\_REQUEST**

**}**

**}**

**export const failRequest=(err)=>{**

**return{**

**type:FAIL\_REQUEST,**

**payload:err**

**}**

**}**

**export const geUserList=(data)=>{**

**return{**

**type:GET\_USER\_LIST,**

**payload:data**

**}**

**}**

**export const deleteUser=()=>{**

**return{**

**type:DELETE\_USER**

**}**

**}**

**export const addUser=()=>{**

**return{**

**type:ADD\_USER**

**}**

**}**

**export const updateUser=()=>{**

**return{**

**type:UPDATE\_USER**

**}**

**}**

**export const getUserObj=(data)=>{**

**return{**

**type:GET\_USER\_OBJ,**

**payload:data**

**}**

**}**

**export const FetchUserList=()=>{**

**return (dispatch)=>{**

**dispatch(makeRequest());**

**//setTimeout(() => {**

**axios.get('http://localhost:8000/user').then(res=>{**

**const userlist=res.data;**

**dispatch(geUserList(userlist));**

**}).catch(err=>{**

**dispatch(failRequest(err.message))**

**})**

**// }, 2000);**

**}**

**}**

**export const Removeuser=(code)=>{**

**return (dispatch)=>{**

**dispatch(makeRequest());**

**//setTimeout(() => {**

**axios.delete('http://localhost:8000/user/'+code).then(res=>{**

**dispatch(deleteUser());**

**}).catch(err=>{**

**dispatch(failRequest(err.message))**

**})**

**// }, 2000);**

**}**

**}**

**export const FunctionAddUser=(data)=>{**

**return (dispatch)=>{**

**dispatch(makeRequest());**

**//setTimeout(() => {**

**axios.post('http://localhost:8000/user',data).then(res=>{**

**dispatch(addUser());**

**toast.success('User Added successfully.')**

**}).catch(err=>{**

**dispatch(failRequest(err.message))**

**})**

**// }, 2000);**

**}**

**}**

**export const FunctionUpdateUser=(data,code)=>{**

**return (dispatch)=>{**

**dispatch(makeRequest());**

**//setTimeout(() => {**

**axios.put('http://localhost:8000/user/'+code,data).then(res=>{**

**dispatch(updateUser());**

**toast.success('User Updated successfully.')**

**}).catch(err=>{**

**dispatch(failRequest(err.message))**

**})**

**// }, 2000);**

**}**

**}**

**export const FetchUserObj=(code)=>{**

**return (dispatch)=>{**

**dispatch(makeRequest());**

**//setTimeout(() => {**

**axios.get('http://localhost:8000/user/'+code).then(res=>{**

**const userlist=res.data;**

**dispatch(getUserObj(userlist));**

**}).catch(err=>{**

**dispatch(failRequest(err.message))**

**})**

**// }, 2000);**

**}**

**}**

A screen shot of a computer

Description automatically generated

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

**export const MAKE\_REQUEST='MAKE\_REQUEST'**

**export const FAIL\_REQUEST='FAIL\_REQUEST'**

**export const GET\_USER\_LIST='GET\_USER\_LIST'**

**export const DELETE\_USER='DELETE\_USER'**

**export const ADD\_USER='ADD\_USER'**

**export const UPDATE\_USER='UPDATE\_USER'**

**export const GET\_USER\_OBJ='GET\_USER\_OBJ'**

**A screenshot of a computer

Description automatically generated**

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

**import { ADD\_USER, DELETE\_USER, FAIL\_REQUEST, GET\_USER\_LIST, GET\_USER\_OBJ, MAKE\_REQUEST, UPDATE\_USER } from "./ActionType"**

**const initialstate = {**

**loading: true,**

**userlist: [],**

**userobj: {},**

**errmessage: ''**

**}**

**export const Reducer = (state = initialstate, action) => {**

**switch (action.type) {**

**case MAKE\_REQUEST:**

**return {**

**...state,**

**loading: true**

**}**

**case FAIL\_REQUEST:**

**return {**

**...state,**

**loading: false,**

**errmessage: action.payload**

**}**

**case GET\_USER\_LIST:**

**return {**

**loading: false,**

**errmessage: '',**

**userlist:action.payload,**

**userobj:{}**

**}**

**case DELETE\_USER:return{**

**...state,**

**loading:false**

**}**

**case ADD\_USER:return{**

**...state,**

**loading:false**

**}**

**case UPDATE\_USER:return{**

**...state,**

**loading:false**

**}**

**case GET\_USER\_OBJ:return{**

**...state,**

**loading:false,**

**userobj:action.payload**

**}**

**default: return state**

**}**

**}**

**A screenshot of a computer program

Description automatically generated**

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

**import { configureStore,combineReducers } from "@reduxjs/toolkit";**

**import logger from "redux-logger";**

**import thunk from "redux-thunk";**

**import { Reducer } from "./Reducer";**

**const rootreducer=combineReducers({user:Reducer});**

**const Store=configureStore({reducer:rootreducer,middleware:[thunk,logger]})**

**export default Store;**

**A screenshot of a computer

Description automatically generated**

**Step 5: Modify the index.js and App.js file**

* 1. Inside the **src** folder, modify the **index.js** file as shown below:

**import React from 'react';**

**import ReactDOM from 'react-dom/client';**

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

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

**import reportWebVitals from './reportWebVitals';**

**import '../node\_modules/bootstrap/dist/css/bootstrap.min.css';**

**import 'react-toastify/dist/ReactToastify.css'**

**const root = ReactDOM.createRoot(document.getElementById('root'));**

**root.render(**

**<React.StrictMode>**

**<App />**

**</React.StrictMode>**

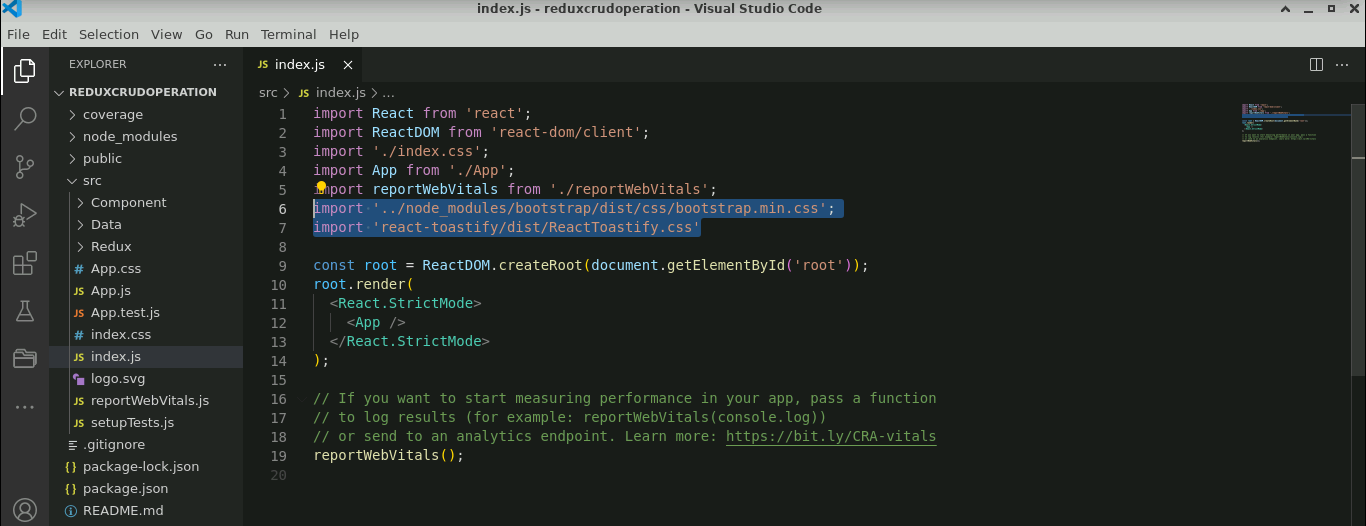
**);**

**// If you want to start measuring performance in your app, pass a function**

**// to log results (for example reportWebVitals(console.log))**

**// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals**

**reportWebVitals();**

****

* 1. Modify the **App.js** file as shown below:

**import logo from './logo.svg';**

**import './App.css';**

**import { BrowserRouter, Link, Route, Routes } from 'react-router-dom';**

**import Home from './Component/Home';**

**import Userlisting from './Component/Userlisting';**

**import Adduser from './Component/Adduser';**

**import Updateuser from './Component/Updateuser';**

**import { ToastContainer } from 'react-toastify';**

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

**import Store from './Redux/Store';**

**function App() {**

**return (**

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

**<div className="App">**

**<BrowserRouter>**

**<div className='header'>**

**<Link to={'/'}>Home</Link>**

**<Link to={'/user'}>User</Link>**

**</div>**

**<Routes>**

**<Route path='/' element={<Home></Home>}></Route>**

**<Route path='/user' element={<Userlisting></Userlisting>}></Route>**

**<Route path='/user/add' element={<Adduser></Adduser>}></Route>**

**<Route path='/user/edit/:code' element={<Updateuser></Updateuser>}></Route>**

**</Routes>**

**</BrowserRouter>**

**<ToastContainer className="toast-position"**

**position="bottom-right"></ToastContainer>**

**</div>**

**</Provider>**

**);**

**}**

**export default App;**

**A screen shot of a computer screen

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Step 6: Run 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

The output appears as shown below:

A screen shot of a computer

Description automatically generated

* 1. Navigate to **Home User** from the above output to get the user details

A screenshot of a computer

Description automatically generated

* 1. Click the **Add User [+]** button, enter the **Name**, **Email**, **Phone,** and **Role**,and click on the **Submit** button

A screenshot of a cell phone

Description automatically generated

The user details have been successfully added as shown below:

A screenshot of a computer

Description automatically generated

* 1. Click on the **Edit** button, enter the **Name**, **Email**, **Phone,** and **Role,** and click on the **Submit** button to edit the existing details

A screenshot of a computer

Description automatically generated

The user details have been successfully updated as shown below:

A screenshot of a computer

Description automatically generated

* 1. Click on the **Delete** button and click on **OK** to delete the details

A screenshot of a computer

Description automatically generated

The user details have been removed successfully.

**A screenshot of a computer

Description automatically generated**

With this, you have successfully implemented a CRUD operation in the React application using JSON server API.