**Lesson 08 Demo 03**

**Creating a React Application with Redux Toolkit**

**Objective:** To create a React application with Redux to interact with a static JSON file for handling data and error

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

**Prerequisites:** None

Steps to be followed:

1. Create and set up the React project
2. Create the db.json file
3. Create components and redux folder under the src folder
4. Configure the index.js file by providing store details
5. 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-tech**

A screenshot of a computer

Description automatically generated

* 1. Open the created React app folder (**react-redux-toolkit-tech**) 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 the **Open** button

**A screen shot 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:

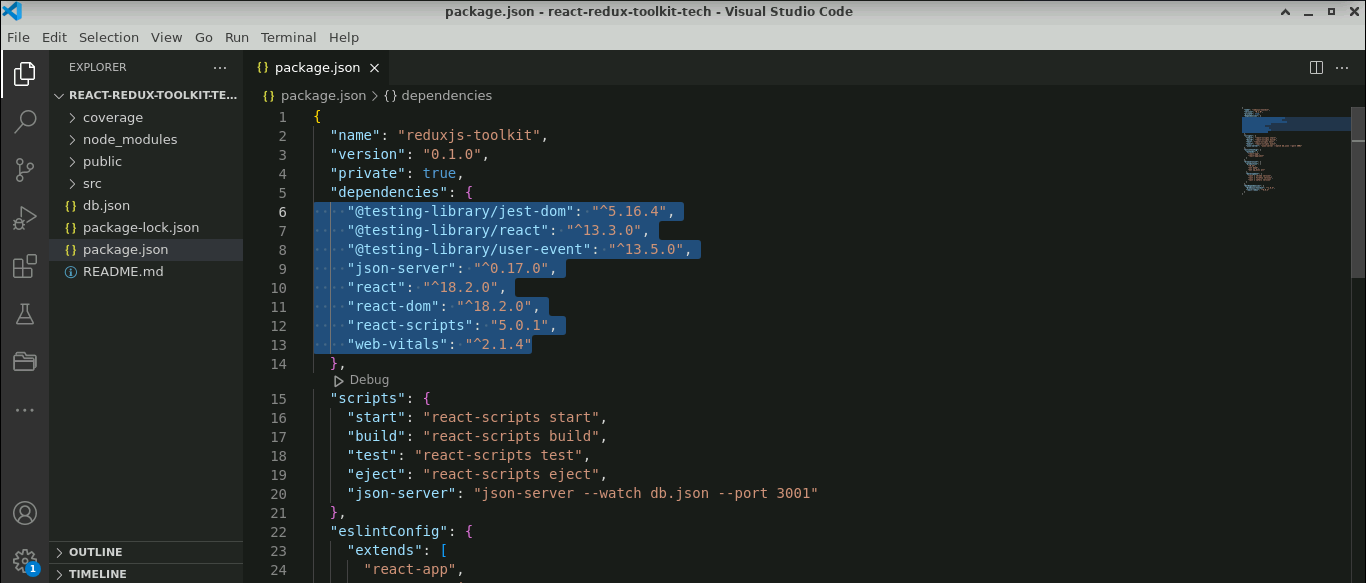
**npm install**

A screenshot of a computer program

Description automatically generated

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

* 1. Open the **package.json** file and view the external dependencies

****

**Step 2: Create a db.json file**

1. Create a **db.json** file that contains static JSON data

**{**

**"technologies": [**

**{**

**"id": 1,**

**"name": "Java : Java is platform independent programming language",**

**"like": 20**

**},**

**{**

**"id": 2,**

**"name": "Python : Python is an open source interpreter scripting language",**

**"like": 16**

**},**

**{**

**"id": 3,**

**"name": "React : React JS is an open source library provided by facebook to develop Web UI Application",**

**"like": 11**

**}**

**]**

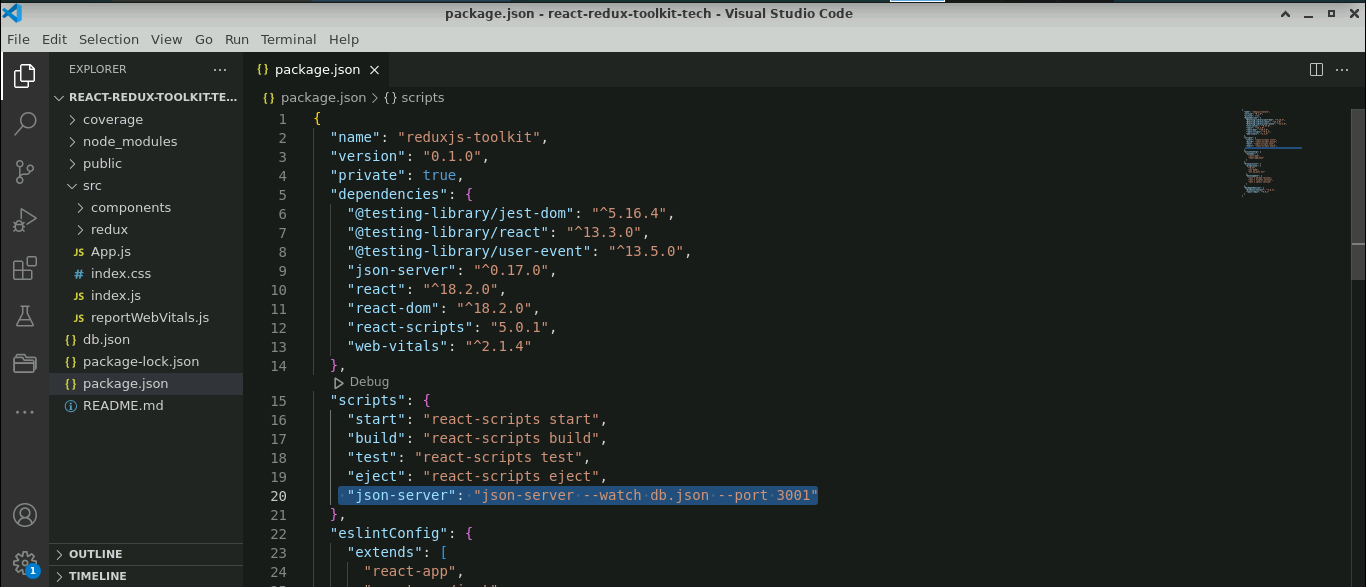
**}**

**A screenshot of a computer

Description automatically generated**

1. Enter the following command in the **package.json** file to install the **json-server** module as shown below:

**json-server --watch db.json --port 3001**



**Step 3: Create components and redux folder under the src folder**

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

**A computer screen shot of a computer

Description automatically generated**

1. Create two folders named **components** and **redux**

A screen shot of a computer

Description automatically generated

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

**import {**

**createAction,**

**createAsyncThunk,**

**createReducer,**

**} from "@reduxjs/toolkit";**

**/\* Actions \*/**

**export const likeForTech = createAction("tech/hit", (tid) => {**

**return { payload: { id: tid } };**

**});**

**export const loadTechs = createAsyncThunk("tech/load", async () => {**

**const response = await fetch("http://localhost:3001/technologies");**

**const technologies = await response.json();**

**return { technologies };**

**});**

**/\* Selectors \*/**

**const selectTechState = (rootState) => rootState.technologies;**

**export const selectTechList = (rootState) => selectTechState(rootState).technologies;**

**export const selectTechLoading = (rootState) => selectTechState(rootState).techLoading;**

**export const selectTechLoadError = (rootState) => selectTechState(rootState).error;**

**/\* Reducer \*/**

**const initialState = {**

**technologies: [],**

**};**

**const techReducer = createReducer(initialState, {**

**[likeForTech]: (state, action) => {**

**const techHit = state.technologies.find(**

**(tech) => tech.id === action.payload.id**

**);**

**techHit.like += 1;**

**},**

**[loadTechs.pending]: (state) => {**

**state.techLoading = true;**

**},**

**[loadTechs.fulfilled]: (state, action) => {**

**state.techLoading = false;**

**state.technologies = action.payload.technologies;**

**},**

**[loadTechs.rejected]: (state) => {**

**state.techLoading = false;**

**state.error =**

**"Error, something went wrong. Please check db.json file running or not";**

**},**

**});**

**export default techReducer;**

**A screenshot of a computer program

Description automatically generated**

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

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

**import techReducer from "./techSlice";**

**const store = configureStore({**

**reducer: { technologies: techReducer },**

**});**

**export default store;**

****

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

**import React from "react";**

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

**import {**

**likeForTech,**

**loadTechs,**

**selectTechList,**

**selectTechLoadError,**

**selectTechLoading,**

**} from "../redux/techSlice";**

**const Technologies = () => {**

**const technologies = useSelector(selectTechList);**

**const loading = useSelector(selectTechLoading);**

**const loadError = useSelector(selectTechLoadError);**

**const dispatch = useDispatch();**

**const handleLike = (storyId) => {**

**dispatch(likeForTech(storyId));**

**};**

**React.useEffect(() => {**

**dispatch(loadTechs());**

**}, [dispatch]);**

**return loading ? (**

**<p>loading...</p>**

**) : loadError ? (**

**<p style={{ color: "red" }}>{loadError}</p>**

**) : (**

**<ul>**

**{technologies.map((tech) => (**

**<li key={tech.id}>**

**<span>**

**{tech.name}**

**</span>**

**<a onClick={() => handleLike(tech.id)}>**

**<img src="data:image/jpeg;base64,**

**"**

**width="50px" height="50px"/>**

**</a>**

**<i>{tech.like}</i>**

**</li>**

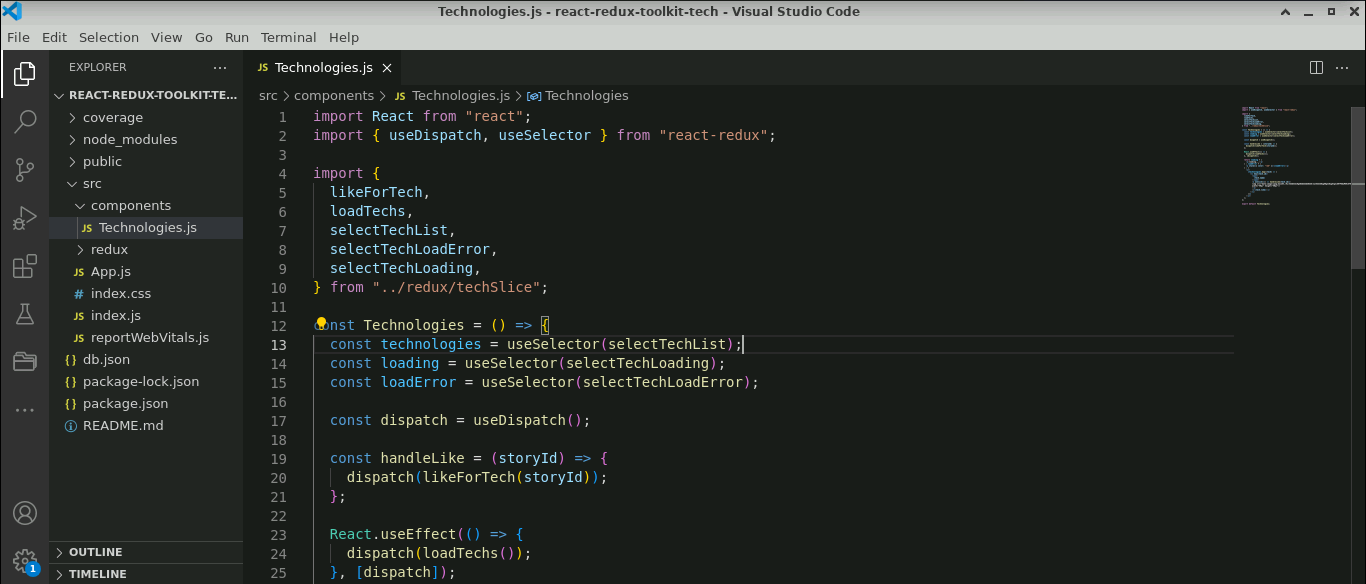
**))}**

**</ul>**

**);**

**};**

**export default Technologies;**

****

**Step 4: Configure the index.js file by providing store details**

1. All configuration information is contained in the **index.js** file, as shown below:

**import React from "react";**

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

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

**import App from "./App";**

**import store from "./redux/store";**

**import "./index.css";**

**import reportWebVitals from "./reportWebVitals";**

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

**root.render(**

**<React.StrictMode>**

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

**<App />**

**</Provider>**

**</React.StrictMode>**

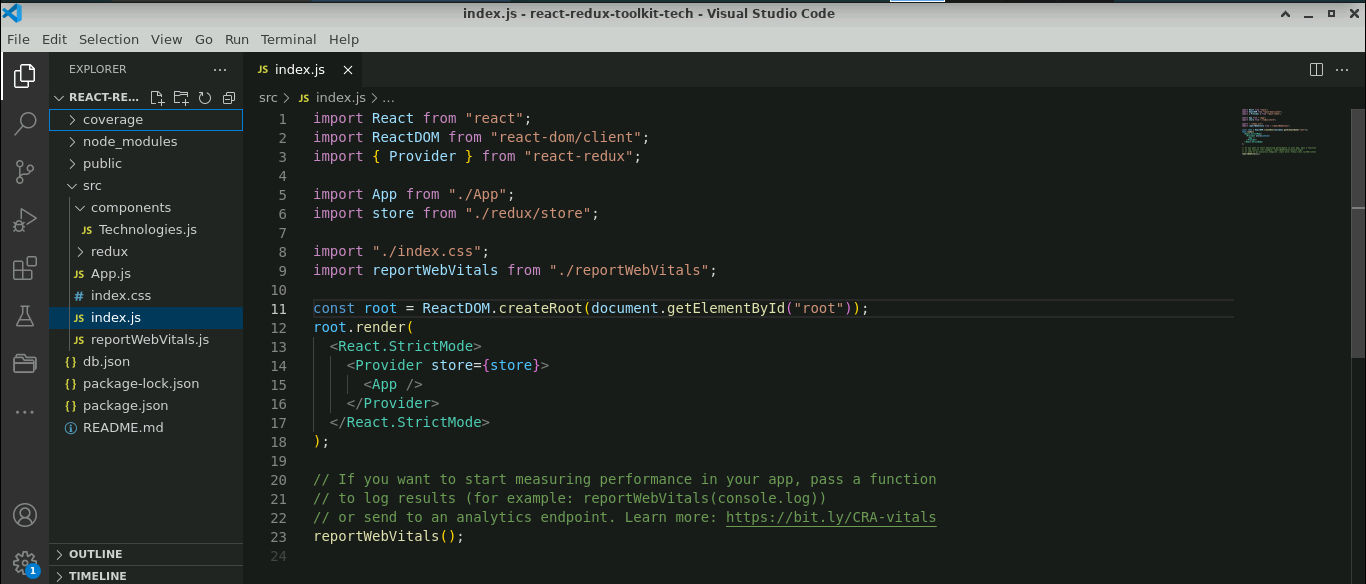
**);**

**// 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**

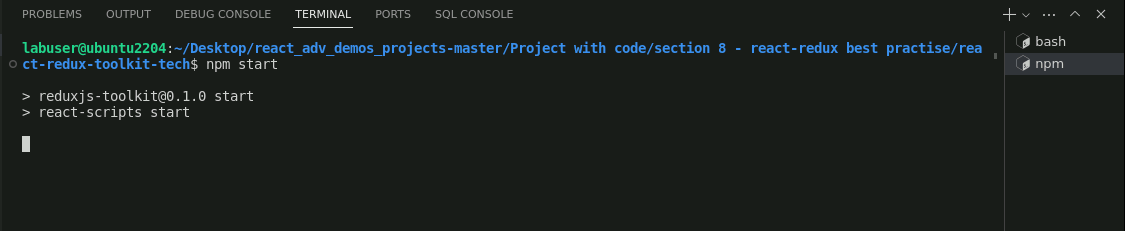
**reportWebVitals();**

****

**Step 5: Test the application**

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

**npm start**

****

The output appears as shown below:

A screenshot of a computer

Description automatically generated

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

**npm run json-server**

A screenshot of a computer

Description automatically generated

* 1. Click on <http://localhost:3001/technologies> to view the technologies details in the database

A screenshot of a computer

Description automatically generated

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

**npm start**

A screen shot 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 to interact with a static JSON file for handling data and errors for state management and asynchronous operations in a structured manner.