**Lesson 08 Demo 01**

**react-** **react-redux-toolkit-tech**

**Objective:** To demonstrate the react with redux interact with static json file with proper way to handle data and handling error.

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

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

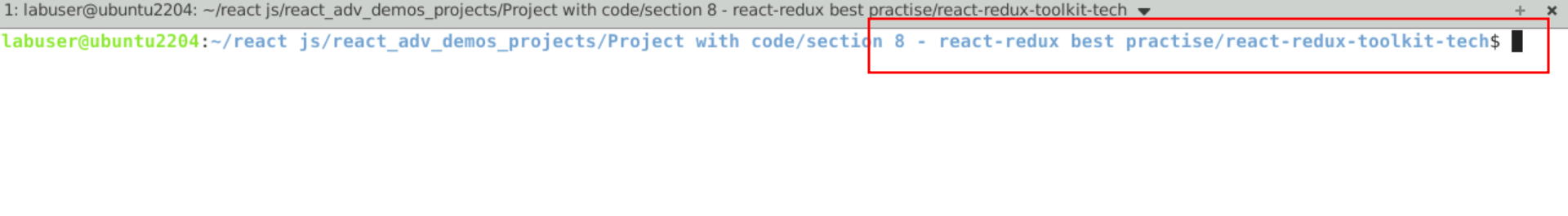
**Note** : All react js project already created with version 18.x with Sample App.js file

**Steps to be followed:**

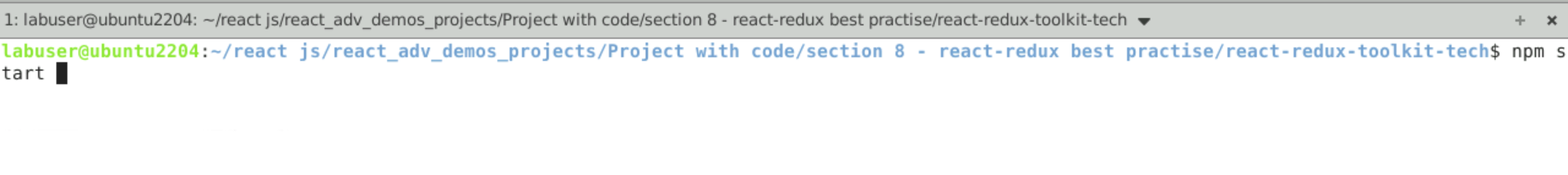
1. Set up for react js project
2. Create db.json file which contains few static employee details in the form of json file.
3. Create redux folder which contains store.js file and techSlice.js file
4. Now create Component folder and inside this folder create Technologeis.js files
5. Create reducers folder which contains employeeReducer.js file
6. In index.js file provide the store configuration details.
7. Now we run the application using npm start

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

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

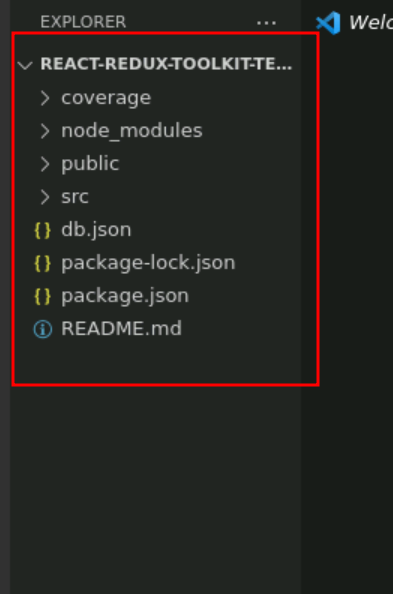


1. Now you need to run the command as **npm install.** This command helps us to installed all required dependencies mention in package.json file in local machine in the form of node\_module folder.

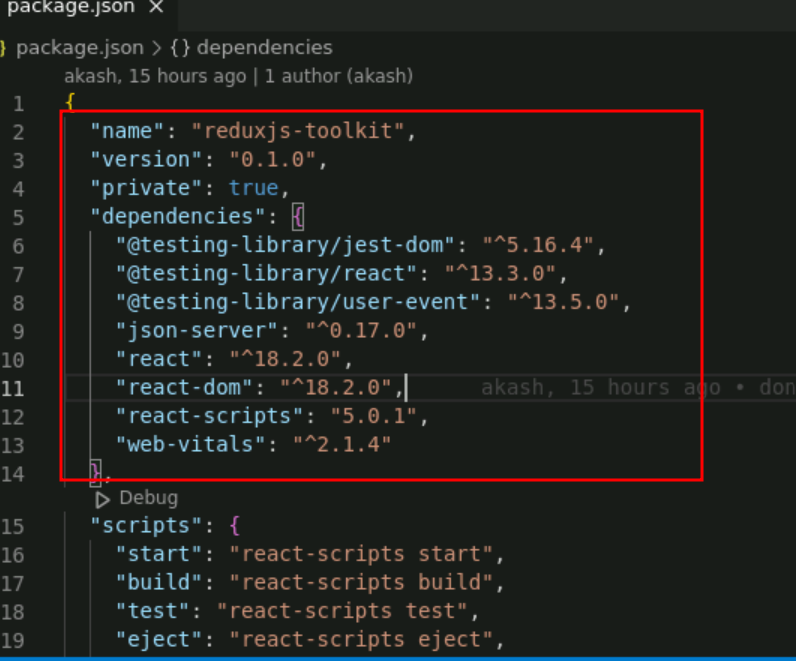


1. Now open **react-redux-toolkit-tech** folder in VS Code Editor

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

****

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

****

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

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

**db.json**

{

"technologies": [

{

"id": 1,

"name": "Java : Java is platform independent programming langauge",

"like": 20

},

{

"id": 2,

"name": "Python : Python is an open source interpreter scripting langauge",

"like": 16

},

{

"id": 3,

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

"like": 11

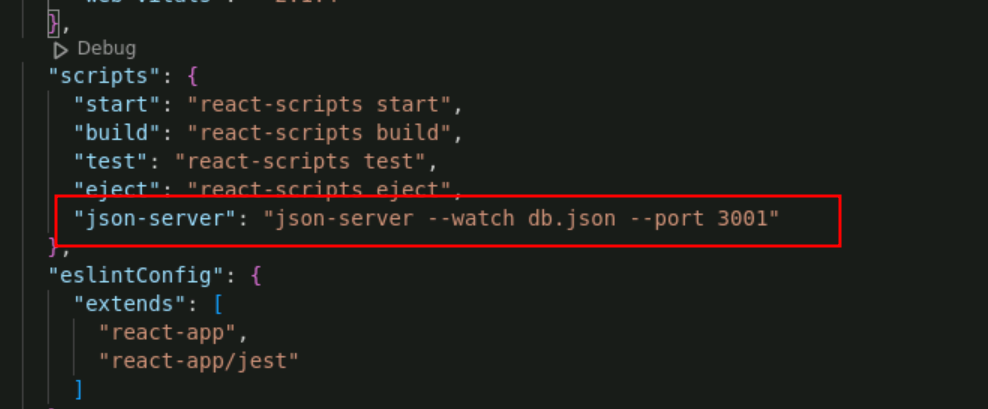
}

]

}

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.**

So here good practise you provide this command details in pakage.json file



Right now don’t run db.json file.

**Step 3:** Create redux folder which contains store.js file and techSlice.js file

3.1 techSlice.js file which contains action, reducer, createAsyncThnk 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;

3.2 store.js. This file contains store details with the help of reducer.

**store.js**

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

import techReducer from "./techSlice";

const store = configureStore({

reducer: { technologies: techReducer },

});

export default store;

**Step 4: Now create Component folder and inside this folder create Technologeis.js files**

4.1 Insider component folder create Technologies.js file which is responsible to interact with redux with useSelector and useDispatch to provide action and access global state variable.

**Technologies.js**

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 5: In index.js file provide the store configuration details.**

5.1 index.js contain all configuration details.

index.js

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

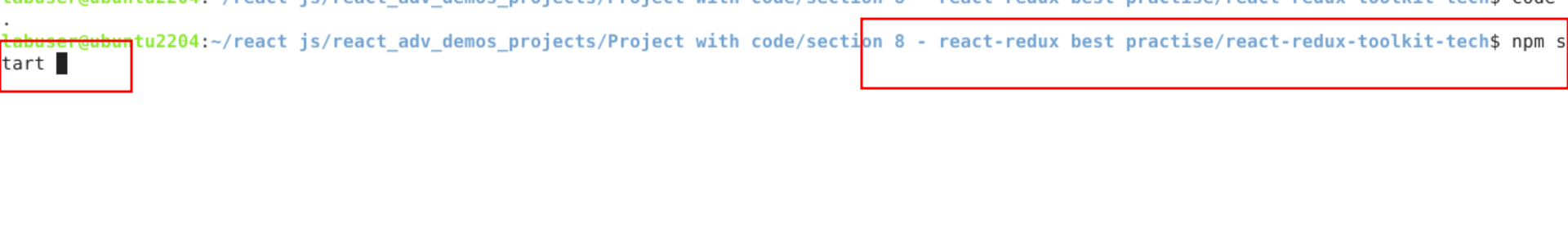
// to log results (for example: reportWebVitals(console.log))

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

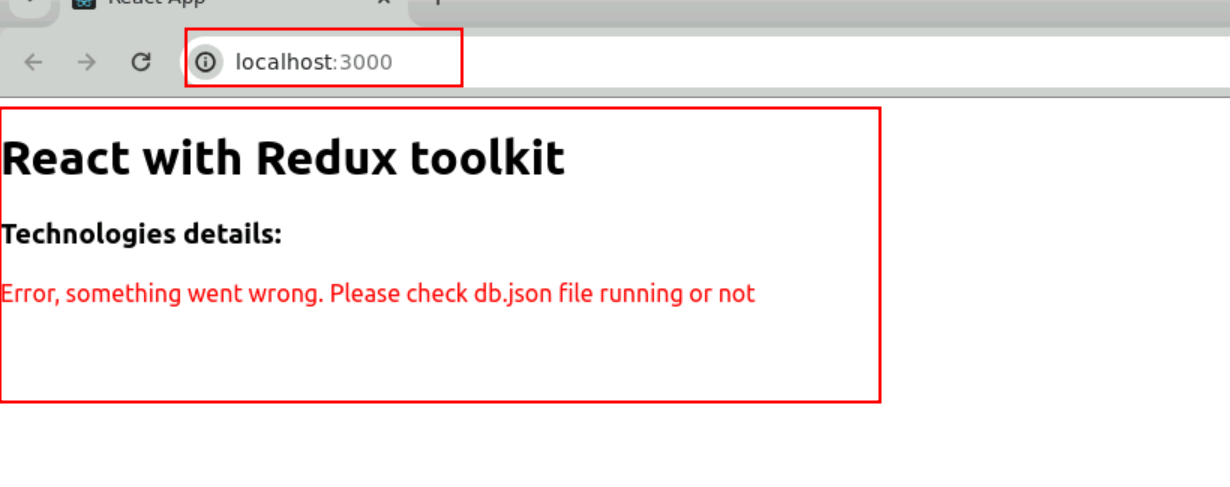
reportWebVitals();

**Step 6 : Now we run the application using npm start**

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

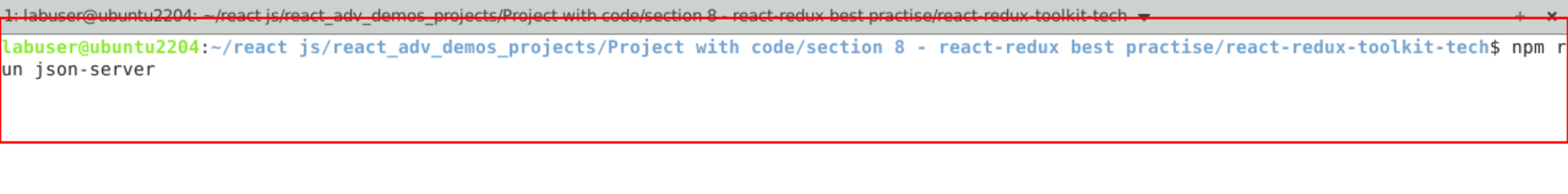


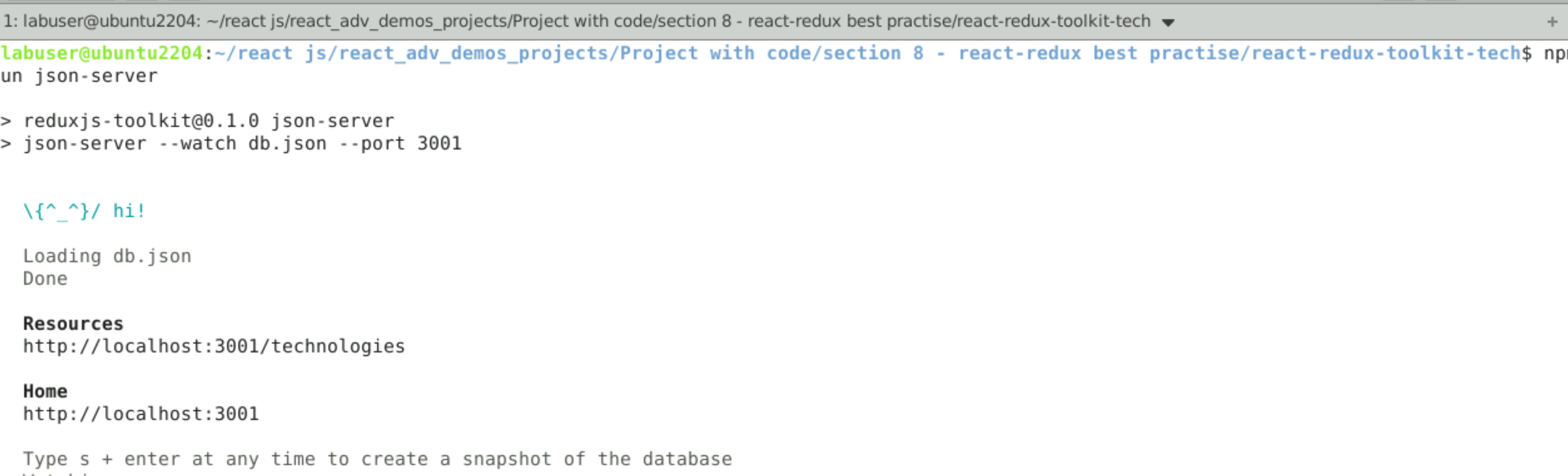
**6.2** If you see the output on browser



6.3 Now to get output on another terminal run the command as

**npm run json-server**





6.4 Now you can refresh the browser to view the output

