**Lesson 06 Demo 01**

**Creating a React Application with Redux Toolkit**

**Objective:** To create a React app with a Redux toolkit to store, delete, edit, and display course details

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

**Prerequisites:** None

Steps to be followed:

1. Create and set up the React project
2. Create the courseSlice.js file
3. Create the store
4. Configure the store in the index.js file
5. Create an AddCourse.js file
6. Modify the App.js file
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 app:

**npx create-react-app react-redux-toolkit-course-operation**



* 1. Open the created React app folder (**react-redux-toolkit-course-operation**) 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**

**A screenshot of a computer

Description automatically generated**

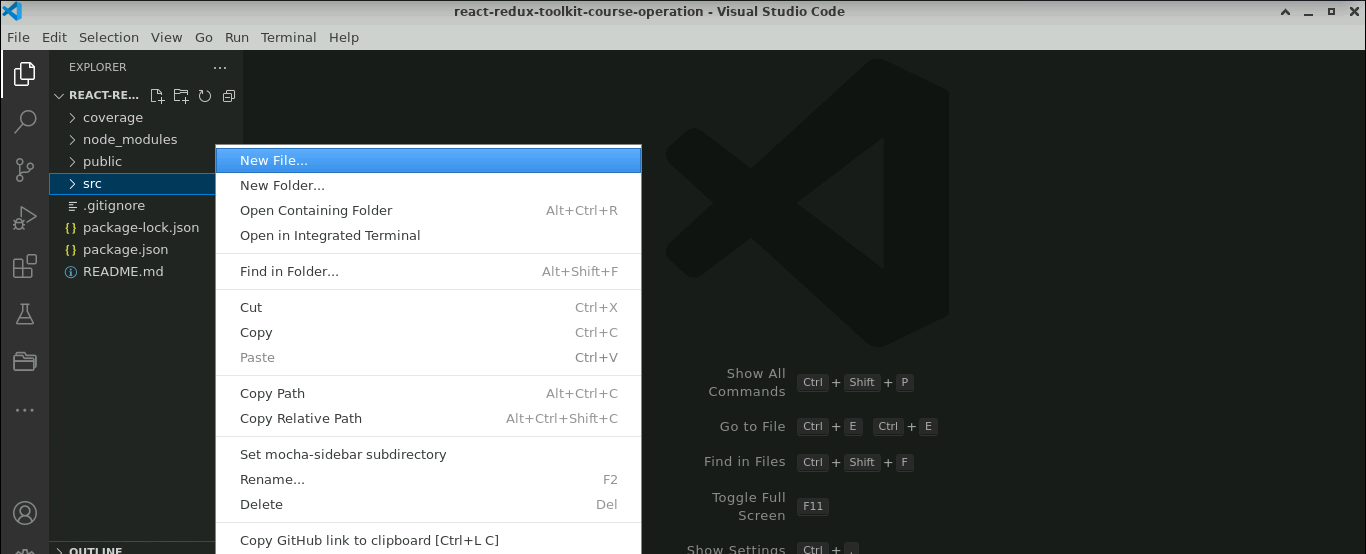
* 1. Open the **package.json** fileand view the external dependencies

**A screen shot of a computer

Description automatically generated**

**Step 2: Create the courseSlice.js file**

1. Right-click on the **src** folder, select **New File**,and create a file named **courseList.js**

****

**A screen shot of a computer

Description automatically generated**

1. Inside the **courseList.js** file, enter the following code:

**import React,{useState} from 'react';**

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

**import { editCourseDetails, deleteCourseDetails } from './courseSlice';**

**const CourseList = ()=> {**

**const courses = useSelector((state)=> state.course);**

**const dispatch = useDispatch();**

**const [editableCourseId,setEditableCourseId]=useState(null);**

**const [editName,setEditName]=useState("");**

**const handleDelete= (id)=> {**

**dispatch(deleteCourseDetails(id));**

**}**

**const handleEdit = (course)=> {**

**setEditName(course.name);**

**setEditableCourseId(course.id);**

**}**

**const handleUpdate = (course)=> {**

**dispatch(editCourseDetails(**

**{**

**id:course.id,**

**name:editName**

**}**

**));**

**setEditableCourseId(null);**

**}**

**return(**

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

**<div>**

**<div className='row offset-3'>**

**<div className='col-7'>**

**<h4>All Course Details</h4>**

**</div>**

**</div>**

**<div className='row offset-3'>**

**<div className='col-9'>**

**<table className='table table-stripped'>**

**{**

**courses.map((course)=> (**

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

**{**

**editableCourseId===course.id ? (**

**<div>**

**<td><input type="text" value={editName}**

**onChange={(e)=>setEditName(e.target.value)} /></td>**

**<td><input type="button" value="Update" onClick={(e)=>handleUpdate(course)} className='btn btn-success'/></td>**

**<td><input type="button" value="Delete" onClick={(e)=>handleDelete(course.id)} className='btn btn-secondary'/></td>**

**</div>**

**):**

**(**

**<div>**

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

**<td><input type="button" value="Edit"**

**onClick={(e)=>handleEdit(course)} className='btn btn-success'/></td>**

**<td><input type="button" value="Delete" onClick={(e)=>handleDelete(course.id)} className='btn btn-secondary'/></td>**

**</div>**

**)**

**}**

**</tr>**

**))}**

**</table>**

**</div>**

**</div>**

**</div>**

**{/\* <h2>Course List</h2>**

**<ul>**

**{**

**courses.map((course)=> (**

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

**{**

**editableCourseId===course.id ? (**

**<div>**

**<input type="text" value={editName}**

**onChange={(e)=>setEditName(e.target.value)} />**

**<input type="button" value="Update" onClick={(e)=>handleUpdate(course)}/>**

**<input type="button" value="Delete" onClick={(e)=>handleDelete(course.id)}/>**

**</div>**

**):**

**(**

**<div>**

**{course.name}**

**<br/>**

**<input type="button" value="Edit"**

**onClick={(e)=>handleEdit(course)}/>**

**<input type="button" value="Delete" onClick={(e)=>handleDelete(course.id)}/>**

**</div>**

**)**

**}**

**</li>**

**))}**

**</ul> \*/}**

**</div>**

**);**

**}**

**export default CourseList;**

**A screen shot of a computer screen

Description automatically generated**

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

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

**const courseSlice = createSlice({**

**name:"course",**

**initialState:[],**

**reducers: {**

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

**state.push(action.payload);**

**},**

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

**const {id,name}= action.payload;**

**const course = state.find((cc)=>cc.id==id);**

**if(course){**

**course.name = name;**

**}**

**},**

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

**return state.filter((course)=>course.id!==action.payload);**

**}**

**}**

**});**

**export const {addCourseDetails,editCourseDetails,deleteCourseDetails}=courseSlice.actions;**

**export default courseSlice.reducer;**

**A screen shot of a computer

Description automatically generated**

**Step 3: Create a store**

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

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

**import courseSlice from "./courseSlice";**

**const store = configureStore({**

**reducer:{**

**course: courseSlice**

**}**

**});**

**export default store;**

**A screenshot of a computer

Description automatically generated**

**Step 4: Configure the store in the index.js file**

1. Inside the **src** folder, open the **index.js** file and configure the store details 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 {Provider} from 'react-redux';**

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

**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();**

**A screenshot of a computer program

Description automatically generated**

**Step 5: Create an AddCourse.js file**

1. Inside the **src** folder,create an **AddCourse.js** file and enter the following code:

**import React, {useState} from 'react';**

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

**import { addCourseDetails } from './courseSlice';**

**const AddCourse = ()=> {**

**const dispatch = useDispatch();**

**const [name,setName]=useState("");**

**const handleSubmit = ()=> {**

**dispatch(addCourseDetails({id:Date.now(),name}));**

**setName("");**

**}**

**return(**

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

**<div className='row offset-4'>**

**<div className='col-4'>**

**<h4>Add Course</h4>**

**</div>**

**</div>**

**<div className='row offset-4'>**

**<div className='col-4'>**

**<input type="text"**

**placeholder='Add Course'**

**value={name}**

**onChange={e=>setName(e.target.value)} className='form-control'**

**/>**

**</div>**

**<div className='col-4'>**

**<input type="button" value="Add Course" onClick={handleSubmit} className='btn btn-primary'/>**

**</div>**

**</div>**

**</div>**

**)**

**}**

**export default AddCourse;**

**A computer screen shot of a program

Description automatically generated**

**Step 6: Modify the App.js file**

* 1. Inside the **src** folder, modify the **App.js** file by adding the user-defined components **AddCourse** and **CourseList**, as shown below code:

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

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

**import AddCourse from './AddCourse';**

**import CourseList from './CourseList';**

**function App() {**

**return (**

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

**<AddCourse></AddCourse>**

**<hr/>**

**<CourseList></CourseList>**

**</div>**

**);**

**}**

**export default App;**

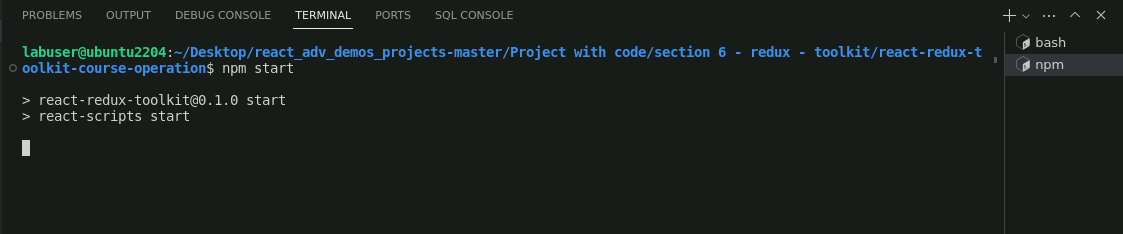
**A screenshot of a computer

Description automatically generated**

**Step 7: Test the application**

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

**npm start**



The output appears as shown below:

A screenshot of a computer

Description automatically generated

* 1. Under the **Add Course** tab, enter the course names (Data Structure, Computer Networks, and Java) and click on the **Add Course** button

A screenshot of a computer

Description automatically generated

The output appears as shown below:

A screenshot of a computer

Description automatically generated

* 1. Click on the **Edit** button (refer to the above screenshot), enter the course name (Computer Graphics) in the **Add Course** tab, and click on the **Update** button

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

The output appears as shown below:

A screenshot of a computer

Description automatically generated

* 1. Click on the **Delete** button (refer to the above screenshot) to delete the course from the **All Course Details** section

A screenshot of a computer

Description automatically generated

With this, you have successfully created a React app with a Redux toolkit to store, delete, edit, and display course details.