**Lesson 06 Demo 01**

**react-redux-toolkit-course-operation**

**Objective:** To demonstrate the react with redux toolkit which help to store, delete, edit and display course details.

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

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

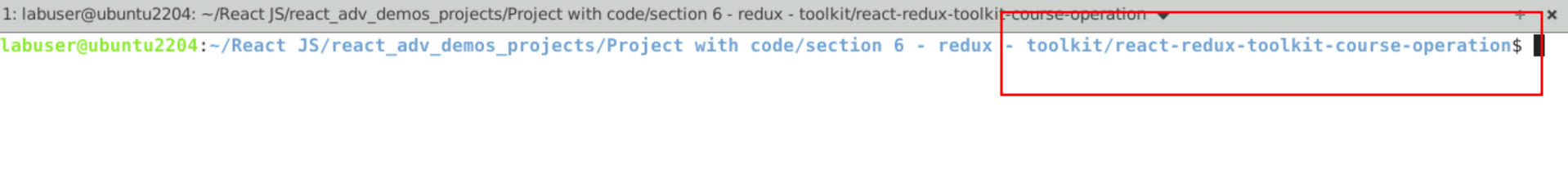
**Note** : All react js project already created with version 18.x with Sample App.js file

**Steps to be followed:**

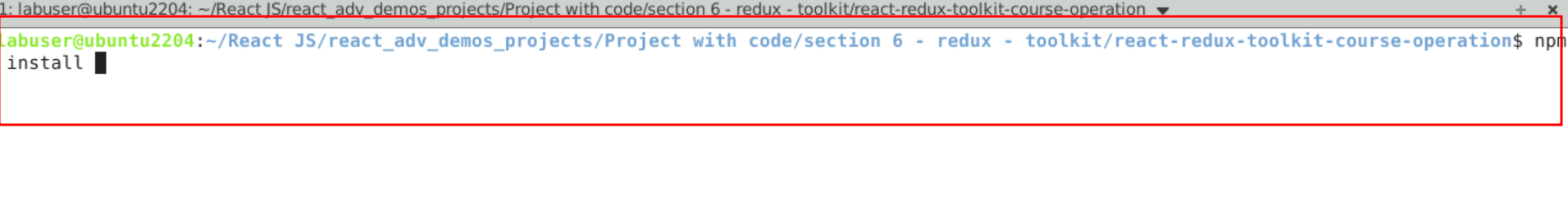
1. Set up for react js project
2. courseSlice.js file which contains createSlice pre defined modules port of redux - toolkit.
3. Create store file with help of slice file.
4. Configure store in index.js file
5. Create AddCourse and CourseList component
6. In App.js file add both AddCouse and CourseList component.
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-course-operation**

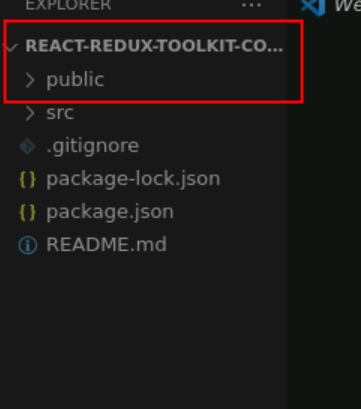


1. Now you need to run the command as **npm install.** This command helps us to installed all required dependencies mention in package.json file in local machine in the form of node\_module folder.

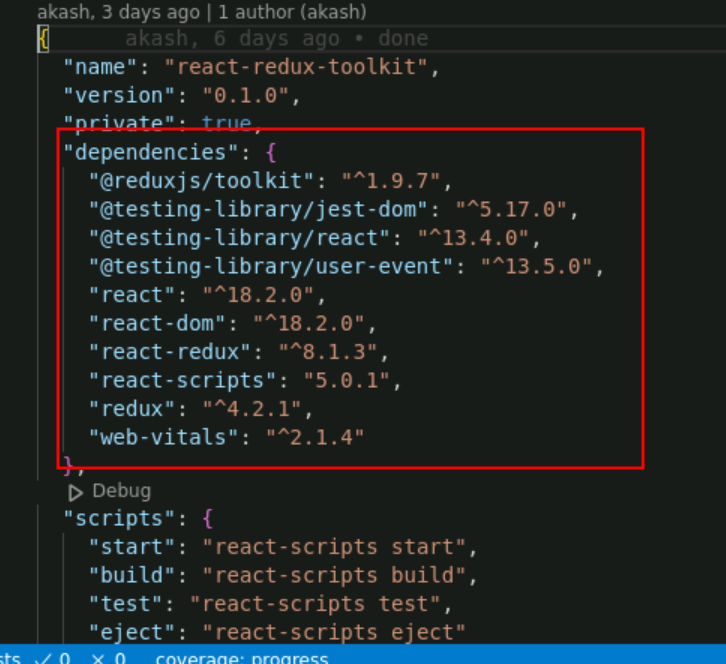


1. Now open **react-redux-toolkit-course-operation** folder in VS Code Editor

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

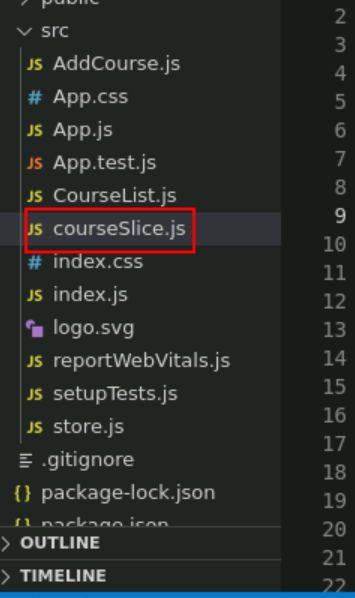
****

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

****

**Step 2:** courseSlice.js file which contains createSlice pre defined modules port of redux - toolkit.

2.1 react courseSlice.js file



2.2 This file take the help of createSlice pre defined module part of redux toolkit. Which provide slice name, initial state and reducer which contains more the one user defined reducer function which takes initial state, and action. Base upon action will do the changes on state variable.

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);

}

}

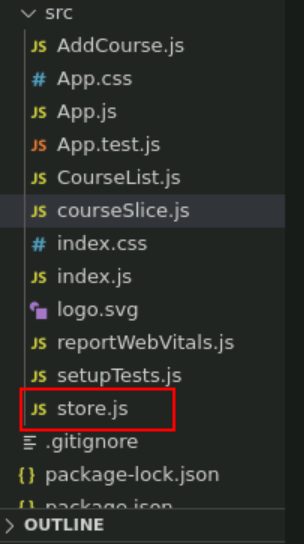
});

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

export default courseSlice.reducer;

**Step 3 : Create store file with help of slice file.**

3.1 create the store.js file



3.2 this file take the help of slice and reduxjs/toolkit module. Create the store.

**store.js**

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

import courseSlice from "./courseSlice";

const store = configureStore({

reducer:{

course: courseSlice

}

});

export default store;

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

4.1 In index.js file configure the store details.

import React from 'react';

import ReactDOM from 'react-dom/client';

import './index.css';

import App from './App';

import reportWebVitals from './reportWebVitals';

i**mport {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();

**Step 5 : Create AddCourse and CourseList component**

5.1 create the AddCourse.js file which contains component which is take the help of useDispatch hook to add the course details.

**AddCourse.js**

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;

5.2 create the CourseList.js file which contains component which is take the help of useDispatch and useSelector hook. It is use to display all details with two button to delete and edit the course name.

**CourseList.js**

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;

**Step 6 : In App.js file add both AddCouse and CourseList component.**

6.1 in App.js file add the user defined component ie AddCourse and CourseList components.

**App.js**

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>

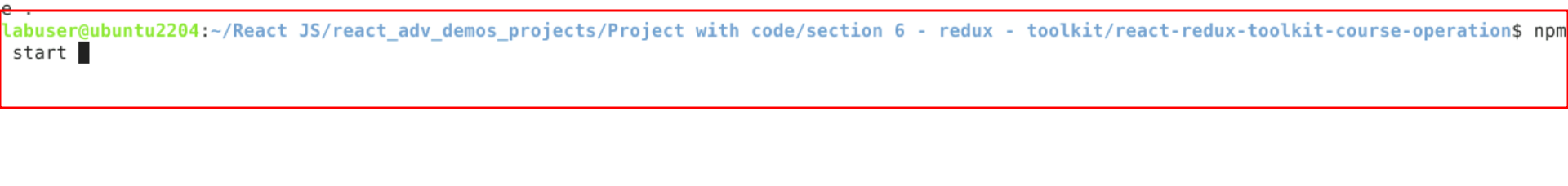
);

}

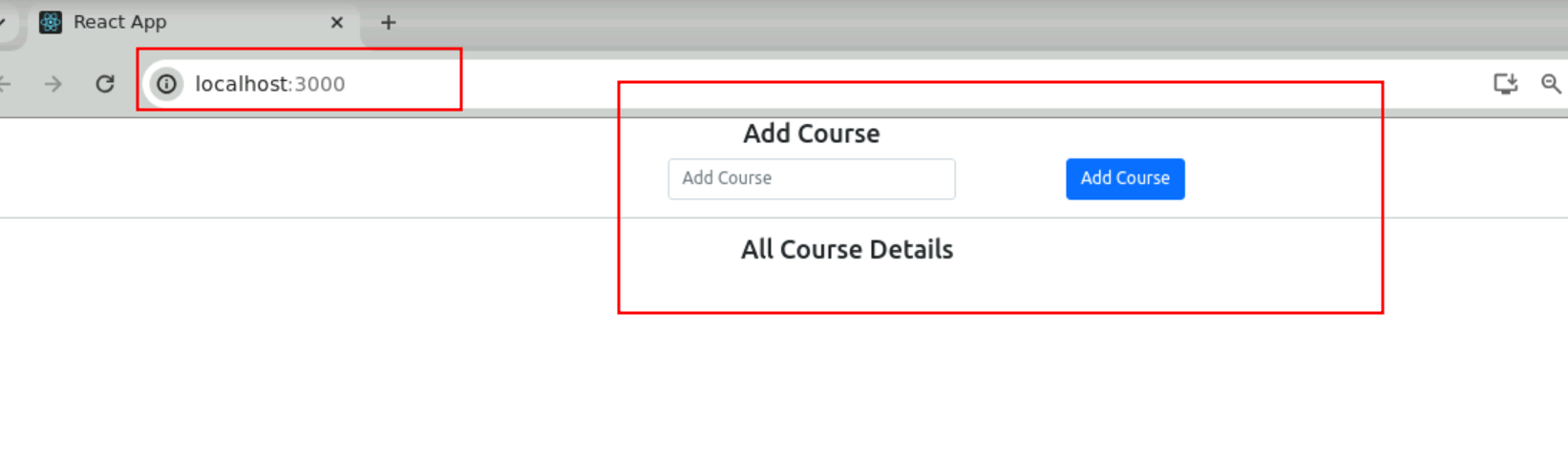
export default App;

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

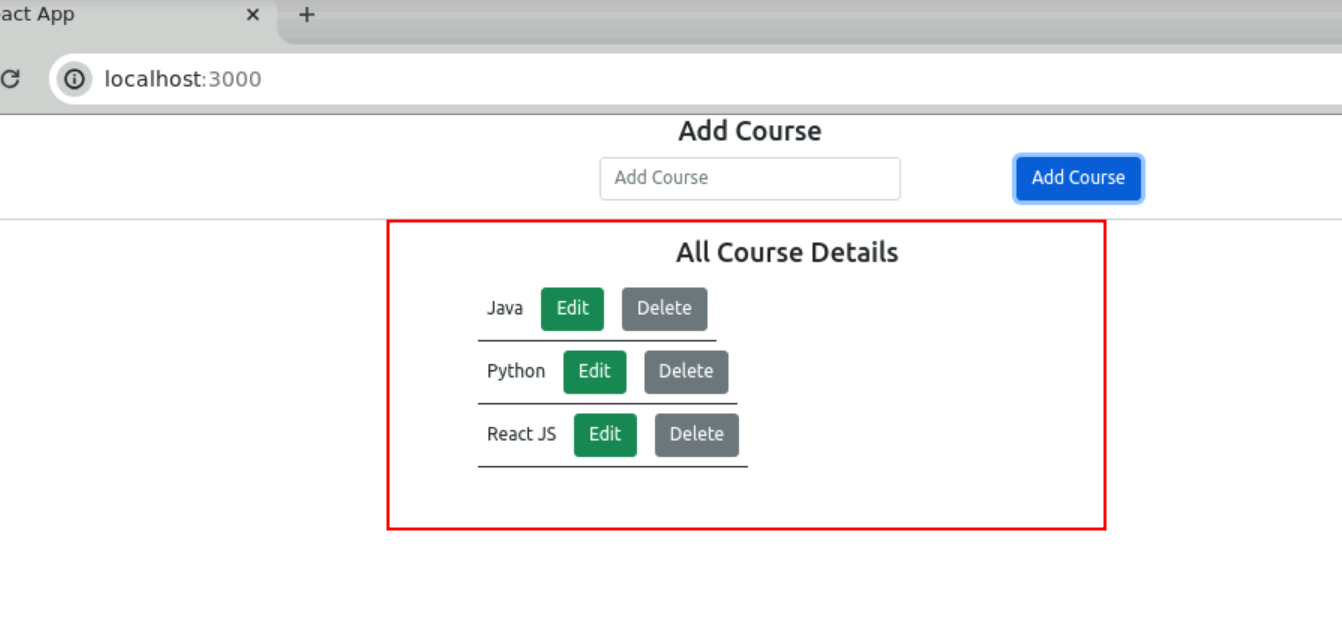
**6.1** Now run the application usiing command as **npm start**



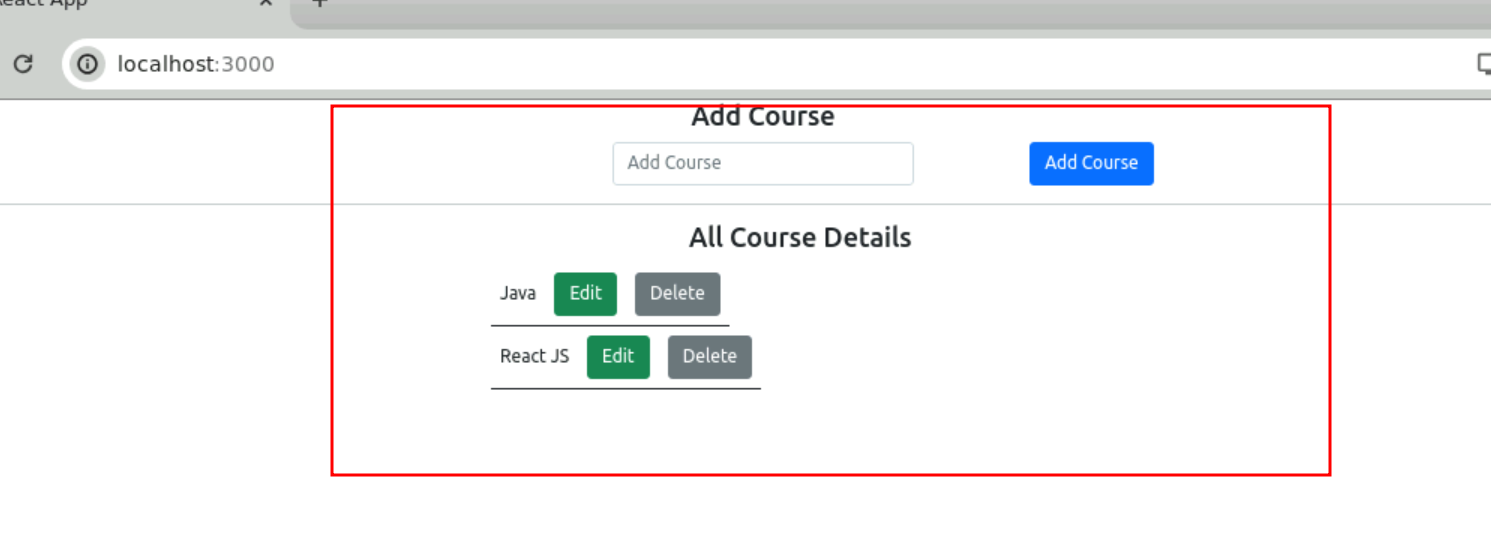
**6. 2**  Now you can view the output on browser. You can see the search text field.



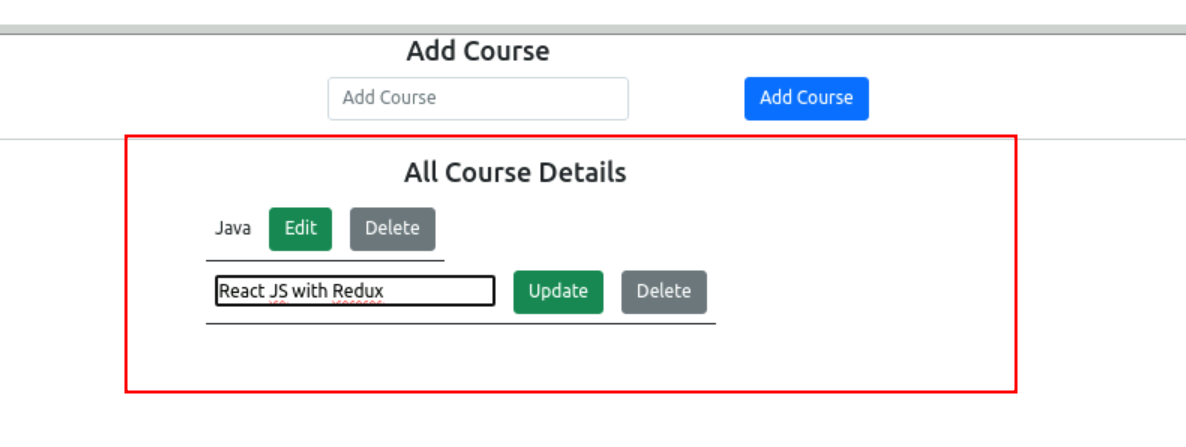
**6.3** Please add few course details.



6.4 Delete the specific course details.

****

6.5 Edit the course details

****

6.6 after updated course details

