**ReactJS - State Management using Redux Assignment Solutions**

**Problem Statement:**

You work in XYZ Company as a Front End Developer. The company officials want you to

build a Todo app using ReactJS

Tasks to be performed:

1. Create a React app using npx create-react-app.

2. Create an app in which there is an input to add todo

3. Create a button that will add todo to the list with the description added in the input

4. When a todo is clicked it should be marked as done and have a strike through it

5. The state should be managed with redux

***Solutions:-***

**Step1: Create a react project todo-redux by running below command on terminal**

npx create-react-app todo-redux

**Step2: Create a *redux* folder inside src directory. redux structure is shown below.**

*Inside redux folder, create below files -* Store.js*,* rootReducer.js

now inside redux folder, create another folder “todoapp”

*Inside todoapp folder, create another folder*

actions -> index.js , reducers -> operations.js



**Step3.1: install redux and react-redux by running below command on terminal**

npm install redux react-redux

**Step3.2: Also install bootstrap and react-icons-kit by running below commands.**

npm install bootstrap

npm i react-icons-kit

**Step4.1: Modify index,js file according to our app requirement.**

**index.js**

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.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();

**Step4.2: Modify App,js file according to our app requirement.**

App.js

function App() {

  return (

    <div className="wrapper">

      <br></br>

      <h1 className="text-center">Todo App | React-Redux</h1>

    </div>

  );

}

export default App;

**Step5.1: create a folder “components” inside src . inside this components folder , create below files. then folder structure will look like**

**Form.js**

import React from 'react';

export const Form = () => {

    return (

        <div>Form</div>

    )

}

**Todos.js**

import React from 'react';

export const Todos = () => {

    return (

        <div>Todos</div>

    )

}

**Step5.2: render Form and Todos component inside App.js**

import { Form } from "./components/Form";

import { Todos } from "./components/Todos";

function App() {

  return (

    <div className="wrapper">

      <br></br>

      <h1 className="text-center">Todo App | React-Redux</h1>

      <Form />

      <Todos />

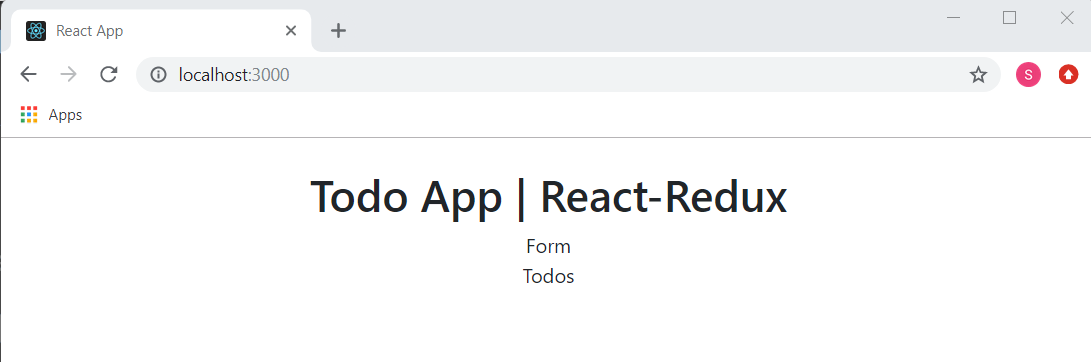
    </div>

  );

}

export default App;

**Step5.3: Open browser and type** <http://localhost:3000/> , **the output will look like as shown below**

****

**Todo App folder structure is shown below.**

****

**Step6: creating form component**

import React,{useState} from 'react';

export const Form = () => {

    const [todoValue, setTodoValue]=useState('');

    const handleSubmit=(e)=>{

        e.preventDefault();

        //creating unique id

        let date = new Date();

        let time = date.getTime();

        //creating todo object

        let todoObj={

            id: time,

            todo: todoValue,

            completed: false

        }

        setTodoValue('');

    }

    return (

        <form className='form-group custom-form' onSubmit={handleSubmit}>

        <label>Add your todo-items</label>

        <div className='input-and-btn'>

          <input type="text" className='form-control' required

            value={todoValue} onChange={(e)=>setTodoValue(e.target.value)}/>

          <button type="submit" className='btn btn-secondary btn-md'>ADD</button>

        </div>

      </form>

    )

}

**Step7: creating dummy todos**

**reducers/operations.js**

const initialState=[

    {id: 1, todo: 'Cooking Food', completed: false},

    {id: 2, todo: 'Learning react-redux', completed: false},

    {id: 3, todo: 'Reading Book', completed: false},

];

export const operationsReducer=(state=initialState, action)=>{

    switch(action.type){

        default: return state;

    }

}

**Step8: rootReducer.js**

import { combineReducers } from "redux";

import { operationsReducer } from "./todoapp/reducers/operations";

export const rootReducer = combineReducers({

    operationsReducer,

    // more reducers can be added here

})

**Step9: store.js**

import {createStore} from 'redux';

import {rootReducer} from './rootReducer';

export const store = createStore(

    rootReducer,

    //redux dev tools extension

    window.\_\_REDUX\_DEVTOOLS\_EXTENSION\_\_ && window.\_\_REDUX\_DEVTOOLS\_EXTENSION\_\_()

)

**Step10: import store , provider component in index.js .**

**Add below code snippets**

*import {store} from './redux/store';*

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

*<Provider store={store}>*

*<App />*

*</Provider>*

**Index.js file code will be**

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.css';

import {store} from './redux/store';

import {Provider} from 'react-redux';

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();

**Step11: to get dummy todos data with the help of reducers**

**Todos.js**

import React from 'react';

import { useSelector } from 'react-redux';

import { Icon } from 'react-icons-kit';

import {trash} from 'react-icons-kit/feather/trash';

export const Todos = () => {

    const todos = useSelector((state)=>state.operationsReducer);

    return todos.map((todo)=>(

        <div key={todo.id} className='todo-box'>

            <div className='content'>

                <input type="checkbox" checked={todo.completed}></input>

                <p style={todo.completed===true?{textDecoration:'line-through'}:{textDecoration:'none'}}>

                    {todo.todo}

                </p>

            </div>

            <div className='actions-box'>

              <span><Icon icon={trash}/></span>

            </div>

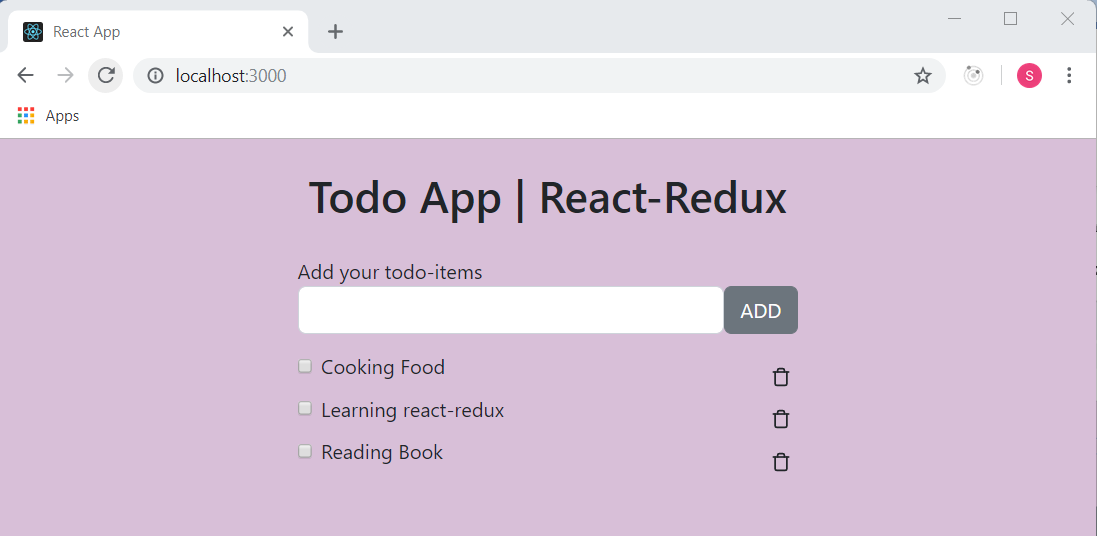
        </div>

    ))

}

**Step12:** After appling some css styling in index.css, the app will look as shown below

**Open browser and type** <http://localhost:3000/> , **the output will look like as shown below**

****

**Step13.1: now to add todo in input field and render it , add below three line of code**

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

*import { addTodo } from '../redux/todoapp/actions';*

*const [todoValue, setTodoValue]=useState('');*

*dispatch(addTodo(todoObj))*

**now Form.js code will be**

import React,{useState} from 'react';

import { useDispatch } from 'react-redux';

import { addTodo } from '../redux/todoapp/actions';

export const Form = () => {

    const dispatch = useDispatch();

    const [todoValue, setTodoValue]=useState('');

    const handleSubmit=(e)=>{

        e.preventDefault();

        //creating unique id

        let date = new Date();

        let time = date.getTime();

        //creating todo object

        let todoObj={

            id: time,

            todo: todoValue,

            completed: false

        }

        setTodoValue('');

        dispatch(addTodo(todoObj))

    }

    return (

        <form className='form-group custom-form' onSubmit={handleSubmit}>

        <label>Add your todo-items</label>

        <div className='input-and-btn'>

          <input type="text" className='form-control' required

            value={todoValue} onChange={(e)=>setTodoValue(e.target.value)}/>

          <button type="submit" className='btn btn-secondary btn-md'>ADD</button>

        </div>

      </form>

    )

}

**Step13.2: todoaap/actions/index.js**

export const ADD\_TODO = 'ADD\_TODO';

export const addTodo=(payload)=>{

    return{

        type: ADD\_TODO,

        payload

    }

}

**Step13.3: import** ADD\_TODO **in reducers/operations.js**

*import { ADD\_TODO } from "../actions";*

*case ADD\_TODO:*

*return [...state, action.payload];*

**operations.js file code will be**

import { ADD\_TODO } from "../actions";

const initialState=[

    {id: 1, todo: 'Cooking Food', completed: false},

    {id: 2, todo: 'Learning react-redux', completed: false},

    {id: 3, todo: 'Reading Book', completed: false},

];

export const operationsReducer=(state=initialState, action)=>{

    switch(action.type){

        case ADD\_TODO:

            return [...state, action.payload];

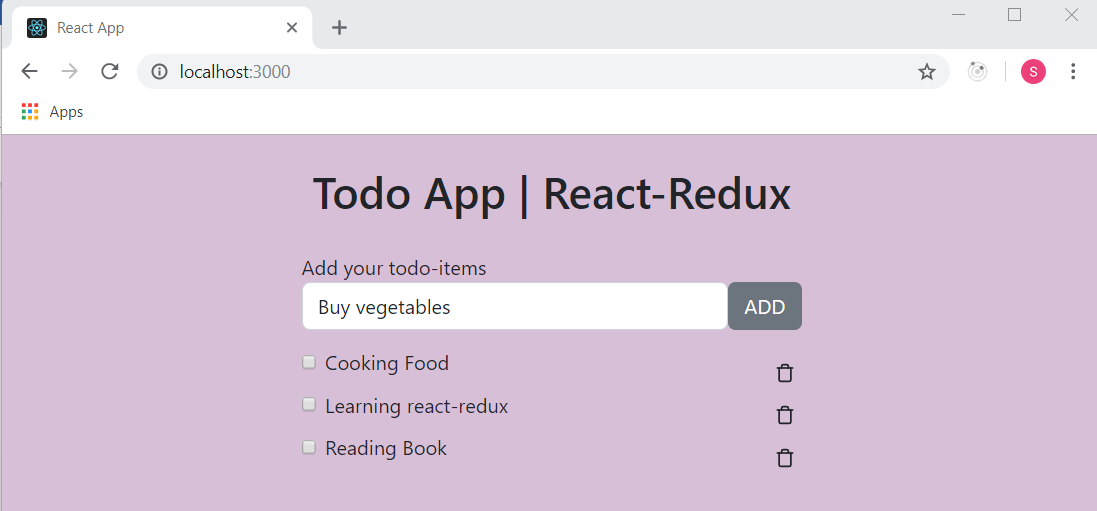
        default: return state;

    }

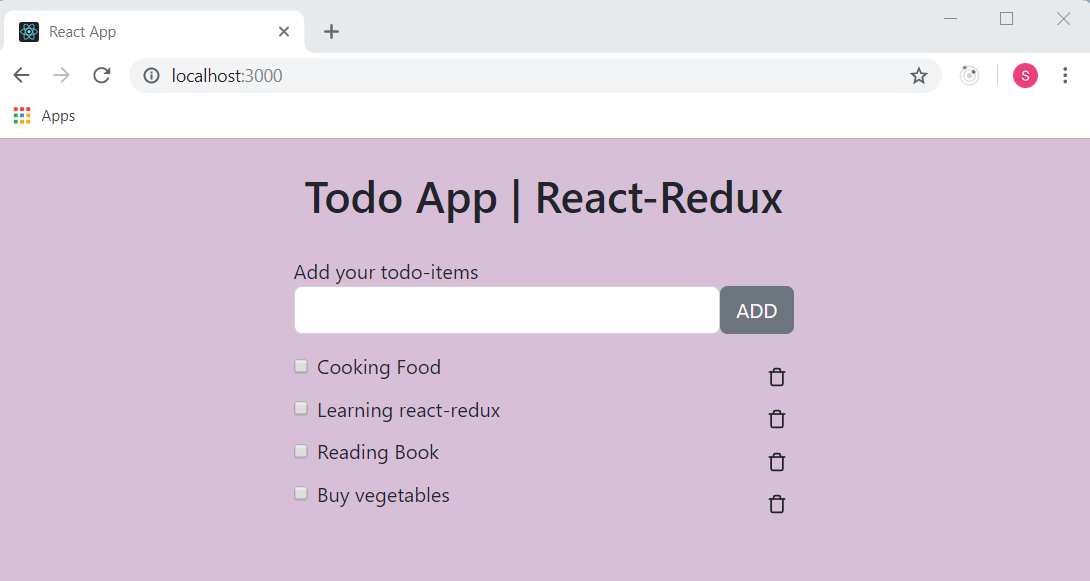
}

**Step14: Open browser and type** <http://localhost:3000/>

*Adding todo in the input field*

****

*When we click on ADD button then “Buy vegetables” todo got added in the todo list*

****

**Step15.1: Now to implement Delete todo, add below code snippets in actions/index.js**

*export const DELETE\_TODO = 'DELETE\_TODO';*

*//for deleting a todo*

*export const deleteTodo=(payload)=>{*

*return{*

*type: DELETE\_TODO,*

*payload*

*}*

*}*

**Index.js file code will be**

export const ADD\_TODO = 'ADD\_TODO';

export const DELETE\_TODO = 'DELETE\_TODO';

//for adding todo

export const addTodo=(payload)=>{

    return{

        type: ADD\_TODO,

        payload

    }

}

//for deleting a todo

export const deleteTodo=(payload)=>{

    return{

        type: DELETE\_TODO,

        payload

    }

}

**Step15.2: add below code snippets in components/Todos.js**

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

*// dispatch function to dispatch an action*

*const dispatch = useDispatch();*

*import { deleteTodo } from '../redux/todoapp/actions';*

*<span onClick={()=>dispatch(deleteTodo(todo.id))}><Icon icon={trash}/></span>*

**Todos.js file code will be**

import React from 'react';

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

import { Icon } from 'react-icons-kit';

import {trash} from 'react-icons-kit/feather/trash';

import { deleteTodo } from '../redux/todoapp/actions';

export const Todos = () => {

    // dispatch function to dispatch an action

    const dispatch = useDispatch();

    // getting todos from the store

    const todos = useSelector((state)=>state.operationsReducer);

    return todos.map((todo)=>(

        <div key={todo.id} className='todo-box'>

            <div className='content'>

                <input type="checkbox" checked={todo.completed}></input>

                <p style={todo.completed===true?{textDecoration:'line-through'}:{textDecoration:'none'}}>

                    {todo.todo}

                </p>

            </div>

            <div className='actions-box'>

            <span onClick={()=>dispatch(deleteTodo(todo.id))}><Icon icon={trash}/></span>

            </div>

        </div>

    ))

}

**Step15.3: add below code snippets in reducers/operations.js**

*import { DELETE\_TODO } from "../actions";*

*case DELETE\_TODO:*

*const filteredTodos = state.filter((todo)=>todo.id!==action.payload);*

*return filteredTodos;*

**operation.js file code will be**

import { ADD\_TODO, DELETE\_TODO } from "../actions";

const initialState=[

    {id: 1, todo: 'Cooking Food', completed: false},

    {id: 2, todo: 'Learning react-redux', completed: false},

    {id: 3, todo: 'Reading Book', completed: false},

];

export const operationsReducer=(state=initialState, action)=>{

    switch(action.type){

        case ADD\_TODO:

            return [...state, action.payload];

        case DELETE\_TODO:

            const filteredTodos = state.filter((todo)=>todo.id!==action.payload);

            return filteredTodos;

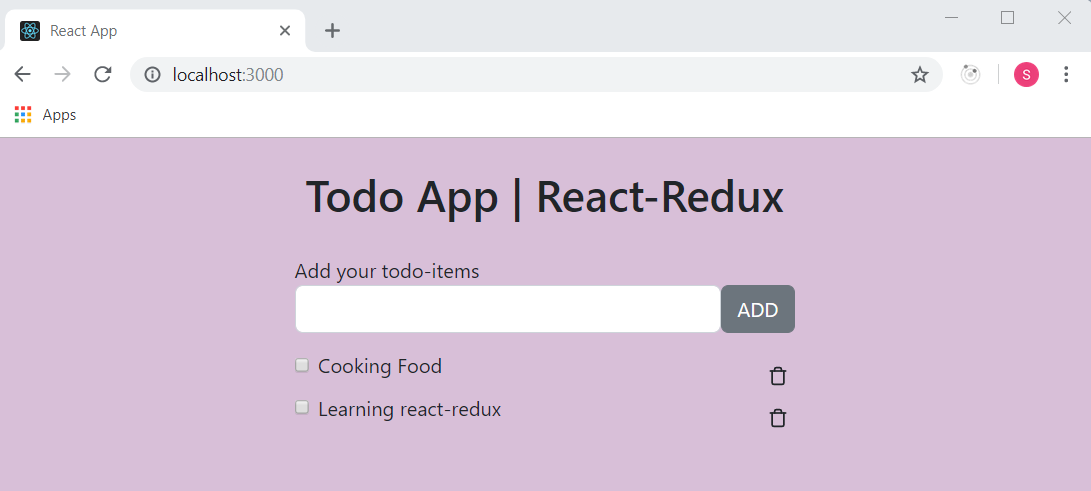
        default: return state;

    }

}

**Step15.4: Open browser and type** <http://localhost:3000/>

**When we click on delete icon, todo get deleted**

****

**Step16.1: Now to impement this case – *When a todo is clicked it should be marked as done and have a strike through it.***

**Add below code snippets in** actions/index.js

*export const UPDATE\_CHECKBOX = 'UPDATE\_CHECKBOX';*

*//handle strike through checkbox*

*export const handleCheckbox=(payload)=>{*

*return{*

*type: UPDATE\_CHECKBOX,*

*payload*

*}*

*}*

**index.js file code will be**

export const ADD\_TODO = 'ADD\_TODO';

export const DELETE\_TODO = 'DELETE\_TODO';

export const UPDATE\_CHECKBOX = 'UPDATE\_CHECKBOX';

//for adding todo

export const addTodo=(payload)=>{

    return{

        type: ADD\_TODO,

        payload

    }

}

//for deleting a todo

export const deleteTodo=(payload)=>{

    return{

        type: DELETE\_TODO,

        payload

    }

}

//handle strike through checkbox

export const handleCheckbox=(payload)=>{

    return{

        type: UPDATE\_CHECKBOX,

        payload

    }

}

**Step16.2: Add below code snippets in Todos.js**

*import { deleteTodo, handleCheckbox } from '../redux/todoapp/actions';*

*<input type="checkbox" onChange={()=>dispatch(handleCheckbox(todo.id))} checked={todo.completed}></input>*

**Todos.js** **file code will be**

import React from 'react';

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

import { Icon } from 'react-icons-kit';

import {trash} from 'react-icons-kit/feather/trash';

import { deleteTodo, handleCheckbox } from '../redux/todoapp/actions';

export const Todos = () => {

    // dispatch function to dispatch an action

    const dispatch = useDispatch();

    // getting todos from the store

    const todos = useSelector((state)=>state.operationsReducer);

    return todos.map((todo)=>(

        <div key={todo.id} className='todo-box'>

            <div className='content'>

                <input type="checkbox" onChange={()=>dispatch(handleCheckbox(todo.id))} checked={todo.completed}></input>

                <p style={todo.completed===true?{textDecoration:'line-through'}:{textDecoration:'none'}}>

                    {todo.todo}

                </p>

            </div>

            <div className='actions-box'>

            <span onClick={()=>dispatch(deleteTodo(todo.id))}><Icon icon={trash}/></span>

            </div>

        </div>

    ))

}

**Step16.3: Add below code snippets in operations.js**

*import { UPDATE\_CHECKBOX } from "../actions";*

*case UPDATE\_CHECKBOX:*

*let todoArray=[];*

*state.map((item)=>{*

*if(item.id===action.payload){*

*item.completed = !item.completed;*

*}*

*todoArray.push(item);*

*})*

*return todoArray;*

**operation.js file code will be**

import { ADD\_TODO, DELETE\_TODO, UPDATE\_CHECKBOX } from "../actions";

const initialState=[

    {id: 1, todo: 'Cooking Food', completed: false},

    {id: 2, todo: 'Learning react-redux', completed: false},

    {id: 3, todo: 'Reading Book', completed: false},

];

export const operationsReducer=(state=initialState, action)=>{

    switch(action.type){

        case ADD\_TODO:

            return [...state, action.payload];

        case DELETE\_TODO:

            const filteredTodos = state.filter((todo)=>todo.id!==action.payload);

            return filteredTodos;

        case UPDATE\_CHECKBOX:

            let todoArray=[];

            state.map((item)=>{

                if(item.id===action.payload){

                    item.completed = !item.completed;

                }

                todoArray.push(item);

            })

            return todoArray;

        default: return state;

    }

}

**Step16.4: Open browser and type** <http://localhost:3000/>

When we click on checkbox , then todo have a strike through it.

