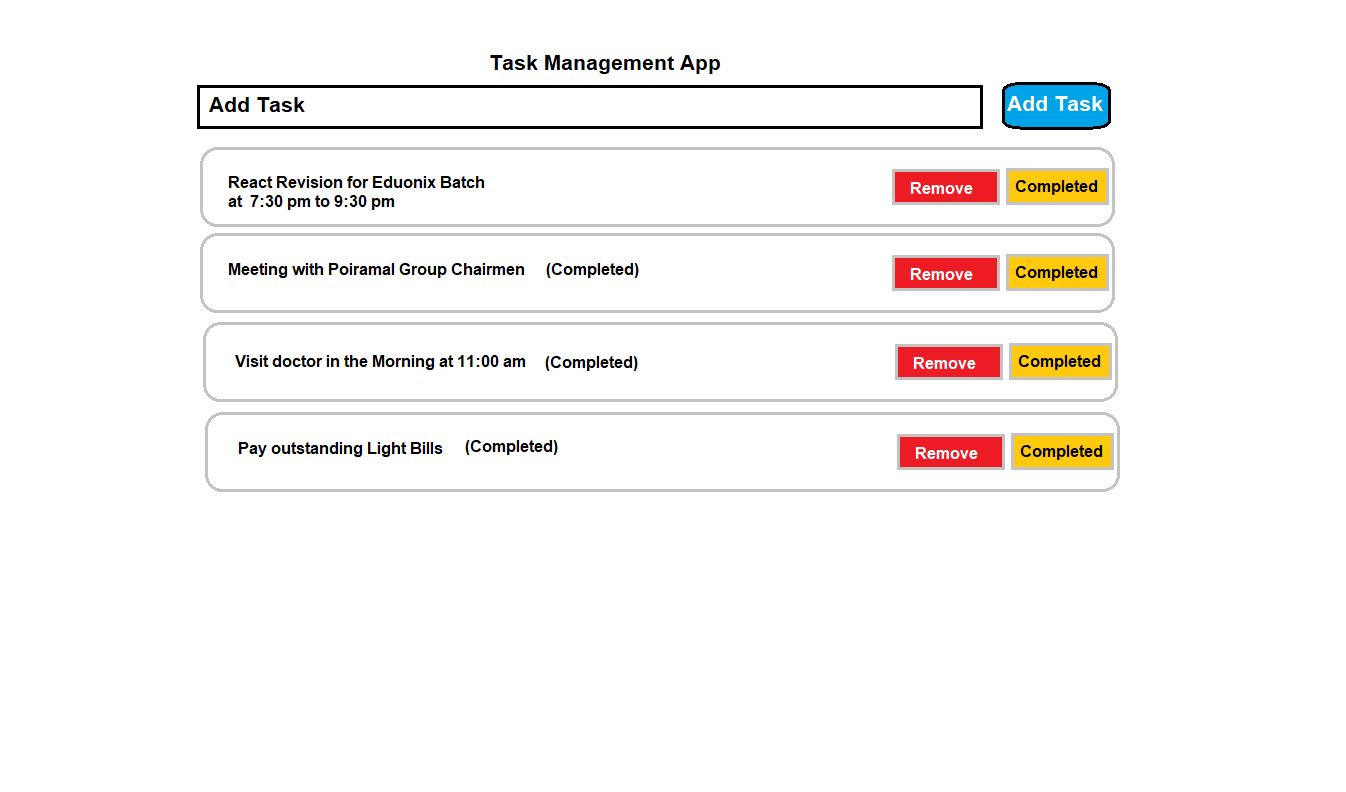
# **React Revision**

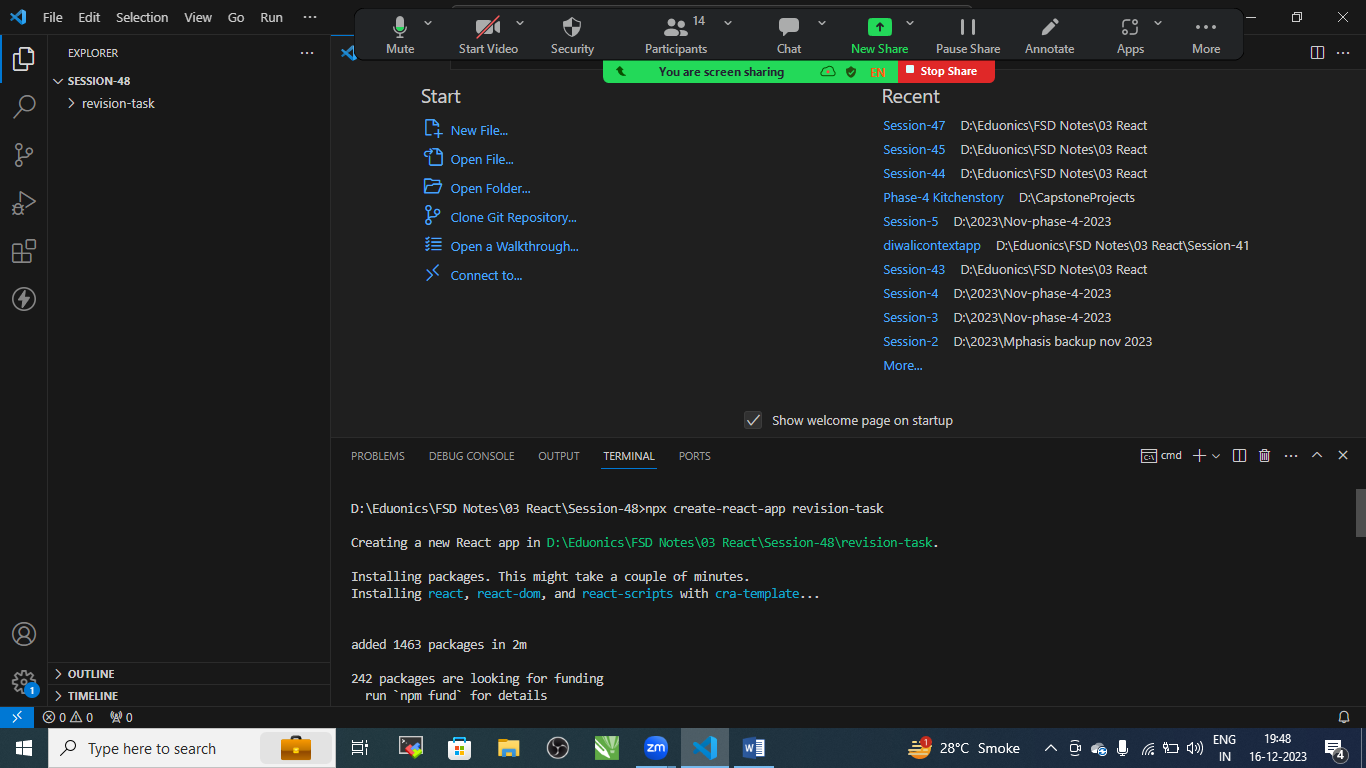


Concepts to be used:

1. useState
2. useEffect
3. Redux
4. useReducer
5. useContext
6. react-redux

using all of the above Concepts we will create React App

**Step:1 Create React Application**

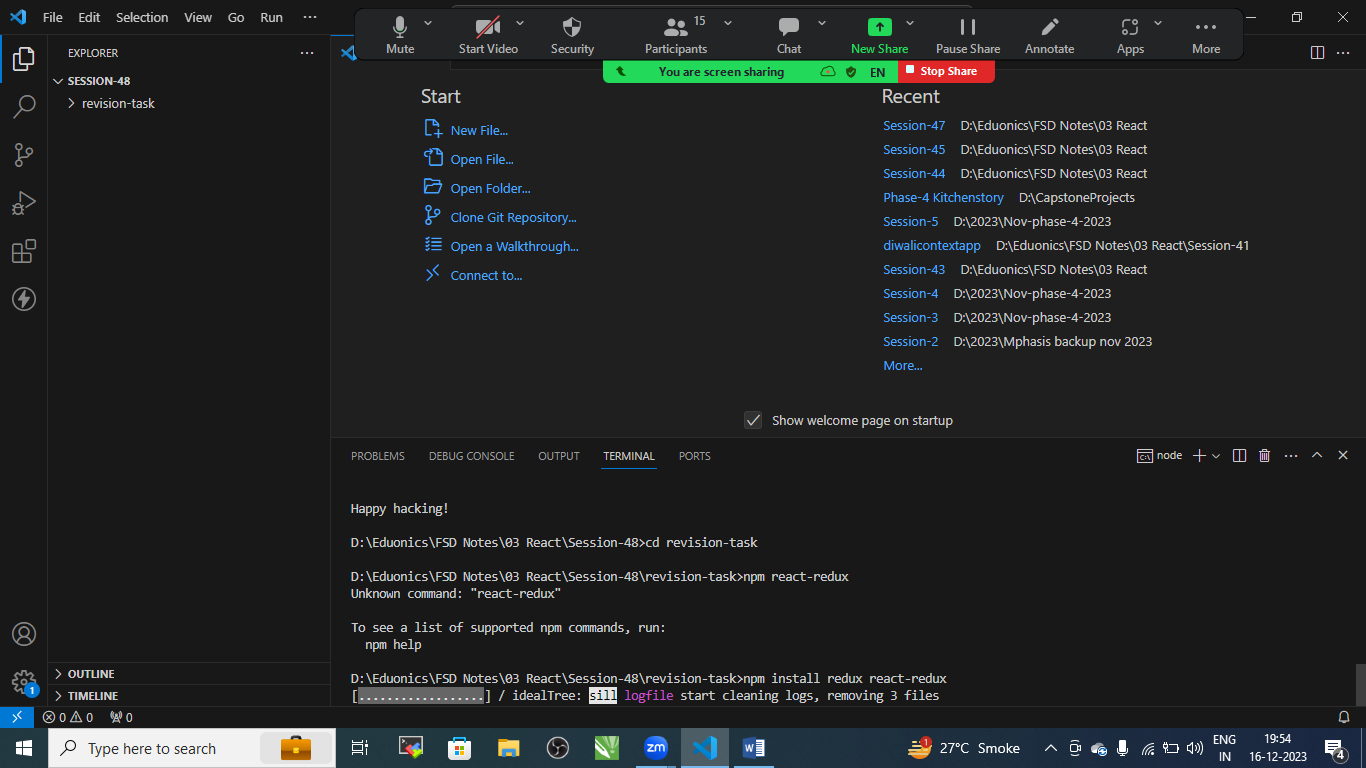
* **npm create-react-app revision-task**

Change the directory

* cd revision-task

Install React Redux

* install install redux react-redux



Let’s Create Folder Structure

/src

/component

TaskForm.js

TaskList.js

/redux

action.js : it tells reducer to manipulate data when user performs some action

reducers.js : it manipulated the data when action is receive

store.js : is what holds all data of your application

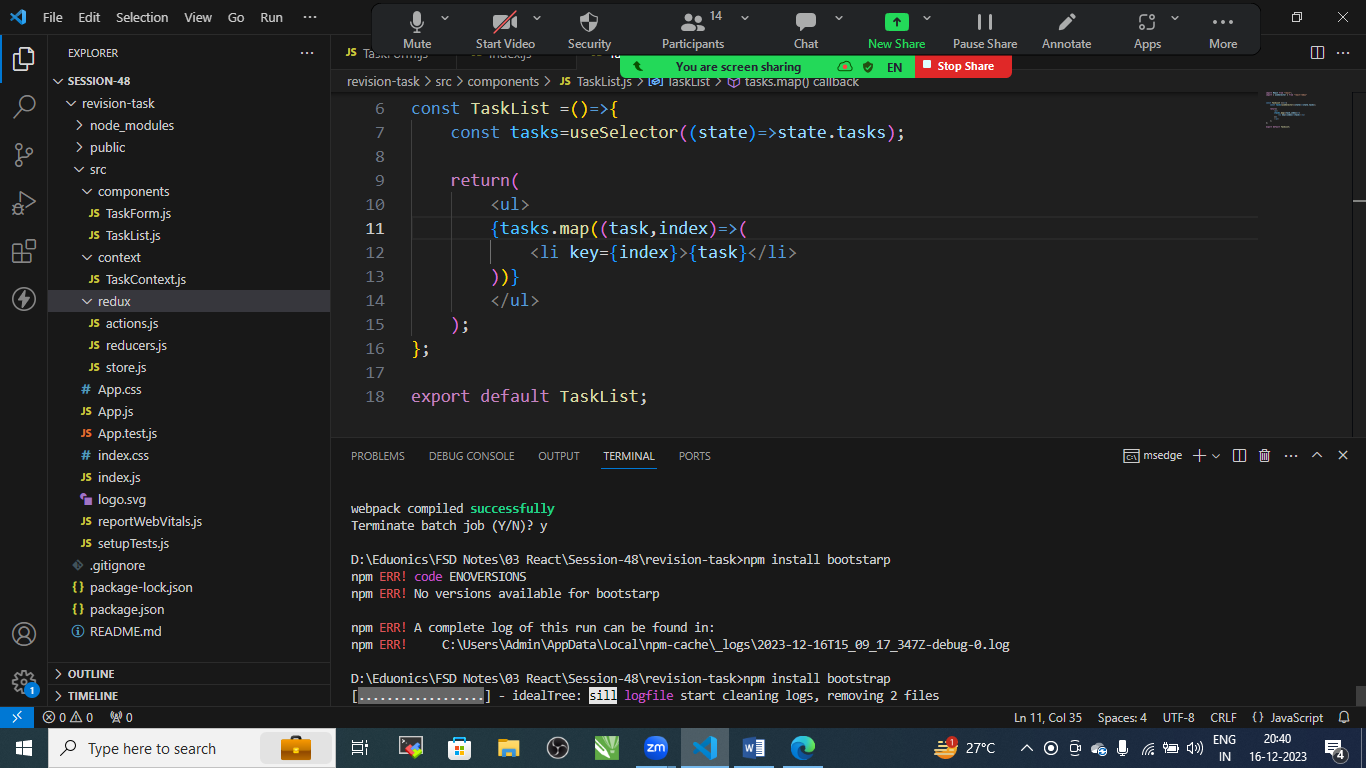
/context

TaskContext.js

App.js

Index.js

Install bootstarp



Go to component folder

TaskForm.js

import React,{useState} from 'react';

import {useDispatch} from 'react-redux'

import { addTask } from '../redux/actions';

const TaskForm=()=>{

    const [task,setTask]=useState('');

    const dispatch= useDispatch();

    const handleSubmit=(e)=>{

        e.preventDefault();

        dispatch(addTask(task));

        setTask('');

    };

    return (

        <form onSubmit={handleSubmit}>

            <input

            type='text'

            placeholder='Add Task'

            value={task}

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

            <button className='btn btn-primary' type='submit'>Add Task</button>

        </form>

    );

};

export default TaskForm;

TaskList

import React from "react";

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

import { removeTask,toggleTask } from "../redux/actions";

const TaskList =()=>{

    const tasks=useSelector((state)=>state.tasks);

    const dispatch=useDispatch();

    const handleRemove=(index)=>{

        dispatch(removeTask(index));

    }

    const handleToggle=(index)=>{

        dispatch(toggleTask(index));

    }

    return(

        <ul className="list-group">

        {tasks.map((task,index)=>(

            <li key={index} className="list-group-item d-flex justify-content-between alighn-items-center">

                {task}

                <div>

                    <button className="btn btn-danger mr-2" onClick={()=>handleRemove(index)}>

                        Remove

                    </button>

                    <button className="btn btn-warning mr-2" onClick={()=>handleToggle(index)}>

                        Completed

                    </button>

                </div>

            </li>

        ))}

        </ul>

    );

};

export default TaskList;

redux folder

action.js

export const ADD\_TASK='ADD\_TASK';

export const REMOVE\_TASK='REMOVE\_TASK';

export const TOGGLE\_TASK='TOGGLE\_TASK';

export const addTask= (task) => ({

    type: ADD\_TASK,

    payload: task,

});

export const removeTask=(index)=>({

    type: REMOVE\_TASK,

    payload: index,

});

export const toggleTask=(index)=>({

    type: TOGGLE\_TASK,

    payload: index,

});

Reducer.js

import { ADD\_TASK, REMOVE\_TASK, TOGGLE\_TASK } from "./actions";

const initialState={

    tasks:[],

};

const reducer=(state=initialState, action)=>{

    switch(action.type){

        case ADD\_TASK:

            return{

                ...state,

                tasks:[...state.tasks,action.payload],

            };

        case REMOVE\_TASK:

            return{

                ...state,

                tasks:state.tasks.filter((\_,index)=>index !== action.payload),

            };

        case TOGGLE\_TASK:

            return{

                ...state,

                tasks:state.tasks.map((task,index)=>

                index === action.payload ? toggleTaskStatus(task):task),

            };

        default:

            return state;

    }

};

const toggleTaskStatus=(task)=>{

    return task.includes('(completed)') ? task.replace('(completed)',''): `${task} (completed)`;

}

export default reducer;

store.js

import { createStore } from "redux";

import reducer from "./reducers";

const store=createStore(reducer);

export default store;

context folder

TaskContext.js

import { createContext, useContext, useReducer } from "react";

import reducer from "../redux/reducers";

const TaskContext=createContext();

export const TaskProvider=({children})=>{

    const [state,dispatch]=useReducer(reducer,{tasks:[]});

    return(

        <TaskContext.Provider value={{state,dispatch}}>

            {children}

        </TaskContext.Provider>

    );

};

export const useTaskContext=()=>{

    const context=useContext(TaskContext);

    if(!context){

        throw new Error('useTaskContext must be used within a TaskProvider');

    }

    return context;

};

App.js

import logo from './logo.svg';

import './App.css';

import { Provider } from 'react-redux';

import store from './redux/store';

import { TaskProvider } from './context/TaskContext';

import TaskForm from './components/TaskForm';

import TaskList from './components/TaskList';

import 'bootstrap/dist/css/bootstrap.min.css';

function App() {

  return (

     <Provider store={store}>

      <TaskProvider>

        <div className='container mt-5'>

          <h1 className='mb-4'>Task Management App</h1>

          <TaskForm/>

          <TaskList/>

        </div>

      </TaskProvider>

     </Provider>

  );

}

export default App;

index .js

import React from 'react';

import ReactDOM from 'react-dom';

import './index.css';

import App from './App';

import reportWebVitals from './reportWebVitals';

 ReactDOM.render(

  <React.StrictMode>

    <App/>

  </React.StrictMode>,

  document.getElementById('root')

 );

reportWebVitals();

start the Application

* npm start

