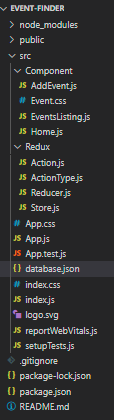
**Event-Finder**

**Source code**

**Project Structure**

****

**Package.json**

{

  "name": "event-finder",

  "version": "0.1.0",

  "private": true,

  "dependencies": {

    "@reduxjs/toolkit": "^1.9.3",

    "@testing-library/jest-dom": "^5.16.5",

    "@testing-library/react": "^13.4.0",

    "@testing-library/user-event": "^13.5.0",

    "axios": "^1.3.4",

    "bootstrap": "^5.2.3",

    "react": "^18.2.0",

    "react-bootstrap": "^2.7.4",

    "react-dom": "^18.2.0",

    "react-redux": "^8.0.5",

    "react-router-dom": "^6.10.0",

    "react-scripts": "5.0.1",

    "react-toastify": "^9.1.2",

    "redux": "^4.2.1",

    "redux-logger": "^3.0.6",

    "redux-thunk": "^2.4.2",

    "web-vitals": "^2.1.4"

  },

  "scripts": {

    "start": "react-scripts start",

    "build": "react-scripts build",

    "test": "react-scripts test",

    "eject": "react-scripts eject"

  },

  "eslintConfig": {

    "extends": [

      "react-app",

      "react-app/jest"

    ]

  },

  "browserslist": {

    "production": [

      ">0.2%",

      "not dead",

      "not op\_mini all"

    ],

    "development": [

      "last 1 chrome version",

      "last 1 firefox version",

      "last 1 safari version"

    ]

  }

}

**App.js**

import './App.css';

import { BrowserRouter, Link, Route, Routes } from 'react-router-dom';

import Home from './Component/Home';

// import Updateuser from './Component/Updateuser';

import { ToastContainer } from 'react-toastify';

import { Provider } from 'react-redux';

import Store from './Redux/Store';

import Adduser from './Component/AddEvent';

import EventsListing from './Component/EventsListing';

function App() {

  return (

    <Provider store={Store}>

    <div className="App">

      <BrowserRouter>

        <div className='header'>

          <Link to={'/'}>Event Finder</Link>

        </div>

        <Routes>

          <Route path='/' element={<EventsListing></EventsListing>}></Route>

          <Route path='/events' element={<EventsListing></EventsListing>}></Route>

          <Route path='/events/add' element={<Adduser></Adduser>}></Route>

          {/\* <Route path='/user/edit/:code' element={<Updateuser></Updateuser>}></Route> \*/}

        </Routes>

      </BrowserRouter>

      <ToastContainer className="toast-position"

        position="bottom-right"></ToastContainer>

    </div>

    </Provider>

  );

}

export default App;

**database.json**

{

  "events": [

    {

      "id": 1,

      "name": "Event 1",

      "date": "02-04-2001",

      "location": "HYD"

    },

    {

      "id": 2,

      "name": "Event 2",

      "date": "23-06-2021",

      "location": "HYD"

    },

    {

      "id": 3,

      "name": "Event 3",

      "date": "12-11-2024",

      "location": "HYD"

    },

    {

      "name": "Event 5",

      "date": "2023-04-18",

      "location": "Delhi",

      "id": 6

    }

  ]

}

**Components/AddEvent.js**

import { useState } from "react";

import { useDispatch } from "react-redux";

import { Link, useNavigate } from "react-router-dom";

import { FunctionAddEvents } from "../Redux/Action";

const AddEvents = () => {

    const [name, namechange] = useState('');

    const [date, datechange] = useState('');

    const [location, locationchange] = useState('');

    const dispatch=useDispatch();

    const navigate=useNavigate();

    const handlesubmit = (e) => {

        e.preventDefault();

        const eventobj = { name, date, location };

        dispatch(FunctionAddEvents(eventobj));

        navigate('/events');

    }

    return (

        <div>

            <form onSubmit={handlesubmit}>

                <div className="card">

                    <div className="card-header" style={{ textAlign: 'left' }}>

                        <h2>Add an Event</h2>

                    </div>

                    <div className="card-body" style={{ textAlign: 'left' }}>

                        <div className="row">

                            <div className="col-lg-12">

                                <div className="form-group">

                                    <label>Name</label>

                                    <input value={name} onChange={e => namechange(e.target.value)} className="form-control"></input>

                                </div>

                            </div>

                            <div className="col-lg-12">

                                <div className="form-group">

                                    <label>Date</label>

                                    <input value={date} type="date" onChange={e => datechange(e.target.value)} className="form-control"></input>

                                </div>

                            </div>

                            <div className="col-lg-12">

                                <div className="form-group">

                                    <label>Location</label>

                                    <input value={location} onChange={e => locationchange(e.target.value)} className="form-control"></input>

                                </div>

                            </div>

                            {/\* <div className="col-lg-12">

                                <div className="form-group">

                                    <label>Role</label>

                                    <select value={role} onChange={e => rolechange(e.target.value)} className="form-control">

                                        <option value="admin">Admin</option>

                                        <option value="staff">Staff</option>

                                    </select>

                                </div>

                            </div> \*/}

                        </div>

                    </div>

                    <div className="card-footer" style={{ textAlign: 'left' }}>

                        <button className="btn btn-primary" type="submit">Submit</button> |

                        <Link className="btn btn-danger" to={'/events'}>Back</Link>

                    </div>

                </div>

            </form>

        </div>

    );

}

export default AddEvents;

**EventListing.js**

import { useEffect } from "react";

import { connect } from "react-redux";

import { Link } from "react-router-dom";

import { toast } from "react-toastify";

import { FetchEventsList,Removeevent } from "../Redux/Action";

import "./Event.css";

import Button from 'react-bootstrap/Button';

import Card from 'react-bootstrap/Card';

const Eventslisting = (props) => {

    useEffect(() => {

        props.loadevents();

    }, [])

    const handledelete = (code) => {

        if (window.confirm('Do you want to remove?')) {

             props.Removeevent(code);

             props.loadevents();

             toast.success('Event Deleted successfully.')

        }

    }

    return (

        props.user.loading ? <div><h2>Loading...</h2></div> :

            props.user.errmessage ? <div><h2>{props.user.errmessage}</h2></div> :

                <div>

                    <div className="card text-center">

                        <div className="card-header" >

                            <Link to={'/events/add'} className="btn btn-success">Add an Event [+]</Link>

                        </div>

                            <h2 style={{fontSize:'70px'}}>Events</h2>

                        <div className="card-body">

                            <div className="container cards\_container">

                            {

                                        props.user.userlist && props.user.userlist.map(item =>

                                            <Card className="bg-primary" style={{ fontSize:"20px",width: '18rem',margin:'10px',color:'black',border:'5px solid black',borderRadius:'0px' }}>

                                            {/\* <Card.Img variant="top" src="holder.js/100px180" /> \*/}

                                            <Card.Body>

                                              <Card.Title><strong>{item.id}.{item.name}</strong></Card.Title>

                                              <Card.Text>

                                               <strong>Date</strong>: {item.date}

                                              </Card.Text>

                                              <Card.Text>

                                              <strong>Location:</strong>  {item.location}

                                              </Card.Text>

                                              <Button onClick={() => { handledelete(item.id) }} className="btn btn-danger">Delete</Button>

                                            </Card.Body>

                                          </Card>

                                        )

                                    }

                                    </div>

                        </div>

                    </div>

                </div>

    );

}

const mapStateToProps = (state) => {

    return {

        user: state.user

    }

}

const mapDispatchToProps = (dispatch) => {

    return {

        loadevents: () => dispatch(FetchEventsList()),

        Removeevent:(code)=>dispatch(Removeevent(code))

    }

}

export default connect(mapStateToProps, mapDispatchToProps)(Eventslisting);

**Event.css**

@import url("https://fonts.googleapis.com/css?family=Roboto+Slab:100,300,400,700");

@import url("https://fonts.googleapis.com/css?family=Raleway:300,300i,400,400i,500,500i,600,600i,700,700i,800,800i,900,900i");

\* {

  margin: 0;

  border: 0;

  box-sizing: border-box;

}

:root {

  --font-roboto: "Roboto Slab", serif;

  --font-raleway: "Raleway", sans-serif;

}

body {

  font-family: var(--font-roboto);

  background-color: #212426;

}

.app {

  padding: 4rem;

  display: flex;

  justify-content: center;

  align-items: center;

  flex-direction: column;

}

h1 {

  font-size: 3rem;

  letter-spacing: 0.9px;

  background: linear-gradient(

    90deg,

    rgba(249, 211, 180, 1) 0%,

    rgba(249, 211, 180, 0) 100%

  );

  background-clip: text;

  -webkit-background-clip: text;

  -webkit-text-fill-color: transparent;

  width: fit-content;

}

.search {

  width: 71%;

  margin: 4rem 0 2rem;

  display: flex;

  align-items: center;

  justify-content: center;

  padding: 1.5rem 1.75rem;

  border-radius: 3rem;

  background: #1f2123;

  -webkit-box-shadow: 5px 5px 7px #1c1d1f, -5px -5px 7px #222527;

  box-shadow: 5px 5px 7px #1c1d1f, -5px -5px 7px #222527;

}

.search input {

  flex: 1;

  border: none;

  font-size: 1.5rem;

  font-family: var(--font-raleway);

  font-weight: 500;

  padding-right: 1rem;

  outline: none;

  color: #a1a1a1;

  background: #1f2123;

}

.search img {

  width: 35px;

  height: 35px;

  cursor: pointer;

}

/\* .search button {

  padding: 20px 40px;

  border-radius: 0.5rem;

  margin-left: 15px;

  color: #a1a1a1;

  font-family: var(--font-raleway);

  font-weight: 900;

  letter-spacing: 0.75px;

  font-size: 1.25rem;

  cursor: pointer;

  background: #1f2123;

  -webkit-box-shadow: 5px 5px 7px #1c1d1f, -5px -5px 7px #222527;

  box-shadow: 5px 5px 7px #1c1d1f, -5px -5px 7px #222527;

} \*/

.empty {

  width: 100%;

  margin-top: 3rem;

  display: flex;

  justify-content: center;

  align-items: center;

}

.empty h2 {

  font-size: 1.25rem;

  color: #f9d3b4;

  font-family: var(--font-raleway);

}

.container {

  width: 100%;

  margin-top: 3rem;

  display: flex;

  justify-content: center;

  align-items: center;

  flex-wrap: wrap;

}

.movie {

  width: 310px;

  height: 460px;

  margin: 1.5rem;

  position: relative;

  border-radius: 12px;

  overflow: hidden;

  border: none;

  transition: all 0.4s cubic-bezier(0.175, 0.885, 0, 1);

  box-shadow: 0px 13px 10px -7px rgba(0, 0, 0, 0.1);

}

.movie div:nth-of-type(1) {

  position: absolute;

  padding: 16px;

  width: 100%;

  opacity: 0;

  top: 0;

  color: #f9d3b4;

}

.movie:hover {

  box-shadow: 0px 30px 18px -8px rgba(0, 0, 0, 0.1);

  transform: scale(1.05, 1.05);

}

.movie div:nth-of-type(2) {

  width: 100%;

  height: 100%;

}

.movie div:nth-of-type(2) img {

  height: 100%;

  width: 100%;

}

.movie div:nth-of-type(3) {

  z-index: 2;

  background-color: #343739;

  padding: 16px 24px 24px 24px;

  position: absolute;

  bottom: 0;

  right: 0;

  left: 0;

}

.movie div:nth-of-type(3) span {

  font-family: "Raleway", sans-serif;

  text-transform: uppercase;

  font-size: 13px;

  letter-spacing: 2px;

  font-weight: 500;

  color: #f0f0f0;

}

.movie div:nth-of-type(3) h3 {

  margin-top: 5px;

  font-family: "Roboto Slab", serif;

  color: #f9d3b4;

}

.movie:hover div:nth-of-type(2) {

  height: 100%;

  opacity: 0.3;

}

.movie:hover div:nth-of-type(3) {

  background-color: transparent;

}

.movie:hover div:nth-of-type(1) {

  opacity: 1;

}

@media screen and (max-width: 600px) {

  .app {

    padding: 4rem 2rem;

  }

  .search {

    padding: 1rem 1.75rem;

    width: 100%;

  }

  .search input {

    font-size: 1rem;

  }

  .search img {

    width: 20px;

    height: 20px;

  }

}

@media screen and (max-width: 400px) {

  .app {

    padding: 4rem 1rem;

  }

  h1 {

    font-size: 2rem;

  }

  .container {

    margin-top: 2rem;

  }

  .movie {

    width: "100%";

    margin: 1rem;

  }

}

.cards\_container{

  margin: 10;

}

**Redux/Action.js**

import axios from "axios"

import { toast } from "react-toastify"

import { ADD\_EVENT, DELETE\_EVENT, FAIL\_REQUEST, GET\_EVENT\_LIST, GET\_EVENT\_OBJ, MAKE\_REQUEST } from "./ActionType"

export const makeRequest=()=>{

    return{

        type:MAKE\_REQUEST

    }

}

export const failRequest=(err)=>{

    return{

        type:FAIL\_REQUEST,

        payload:err

    }

}

export const geteventList=(data)=>{

    return{

        type:GET\_EVENT\_LIST,

        payload:data

    }

}

export const deleteevent=()=>{

    return{

        type:DELETE\_EVENT

    }

}

export const addevent=()=>{

    return{

        type:ADD\_EVENT

    }

}

export const geteventObj=(data)=>{

    return{

        type:GET\_EVENT\_OBJ,

        payload:data

    }

}

export const FetchEventsList=()=>{

    return (dispatch)=>{

      dispatch(makeRequest());

      //setTimeout(() => {

        axios.get('http://localhost:8000/events').then(res=>{

            const eventlist=res.data;

            dispatch(geteventList(eventlist));

          }).catch(err=>{

            dispatch(failRequest(err.message))

          })

     // }, 2000);

    }

}

export const Removeevent=(code)=>{

    return (dispatch)=>{

      dispatch(makeRequest());

      //setTimeout(() => {

        axios.delete('http://localhost:8000/events/'+code).then(res=>{

            dispatch(deleteevent());

          }).catch(err=>{

            dispatch(failRequest(err.message))

          })

     // }, 2000);

    }

}

export const FunctionAddEvents=(data)=>{

    return (dispatch)=>{

      dispatch(makeRequest());

      //setTimeout(() => {

        axios.post('http://localhost:8000/events',data).then(res=>{

            dispatch(addevent());

            toast.success('Event Added successfully.')

          }).catch(err=>{

            dispatch(failRequest(err.message))

          })

     // }, 2000);

    }

}

export const FetchEventsObj=(code)=>{

    return (dispatch)=>{

      dispatch(makeRequest());

      //setTimeout(() => {

        axios.get('http://localhost:8000/events/'+code).then(res=>{

            const eventlist=res.data;

            dispatch(geteventObj(eventlist));

          }).catch(err=>{

            dispatch(failRequest(err.message))

          })

     // }, 2000);

    }

}

**Redux/ActionType.js**

export const MAKE\_REQUEST='MAKE\_REQUEST'

export const FAIL\_REQUEST='FAIL\_REQUEST'

export const GET\_EVENT\_LIST='GET\_EVENT\_LIST'

export const DELETE\_EVENT='DELETE\_EVENT'

export const ADD\_EVENT='ADD\_EVENT'

export const GET\_EVENT\_OBJ='GET\_EVENT\_OBJ'

**Redux/Reducer.js**

import { ADD\_EVENT, DELETE\_EVENT, FAIL\_REQUEST, GET\_EVENT\_LIST, GET\_EVENT\_OBJ, MAKE\_REQUEST } from "./ActionType"

const initialstate = {

    loading: true,

    eventlist: [],

    eventobj: {},

    errmessage: ''

}

export const Reducer = (state = initialstate, action) => {

    switch (action.type) {

        case MAKE\_REQUEST:

            return {

                ...state,

                loading: true

            }

        case FAIL\_REQUEST:

            return {

                ...state,

                loading: false,

                errmessage: action.payload

            }

        case GET\_EVENT\_LIST:

            return {

                loading: false,

                errmessage: '',

                eventlist:action.payload,

                eventobj:{}

            }

            case DELETE\_EVENT:return{

                ...state,

                loading:false

            }

            case ADD\_EVENT:return{

                ...state,

                loading:false

            }

            case GET\_EVENT\_OBJ:return{

                ...state,

                loading:false,

                eventobj:action.payload

            }

        default: return state

    }

}

**Redux/store.js**

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

import logger from "redux-logger";

import thunk from "redux-thunk";

import { Reducer } from "./Reducer";

const rootreducer=combineReducers({event:Reducer});

const Store=configureStore({reducer:rootreducer,middleware:[thunk,logger]})

export default Store;