

SOURCECODE

- EventManager.js

```
• import React from 'react';
• import ReactDOM from 'react-dom';
• import { Redirect } from 'react-router-dom'
• import {store, logout} from '../sessionStore'
•
• class EventManager extends React.Component{
•   state = {events:[]}
•   logoutUser() {
•     store.dispatch(logout())
•     return <Redirect to ="/logout" />
•   }
•   componentDidMount(){
•     var url = "http://localhost:3000/events";
•
•     fetch(url)
•     .then(response=>response.json())
•     .then(response=>this.setState({events:response}))
•   }
•
•   componentDidUpdate(){
•     var url = "http://localhost:3000/events";
•     fetch(url)
•     .then(response=>response.json())
•     .then(response =>
this.setState({events:response}));
•   }
•
•   AddEvents(){
•     let event = {
•       "id": Number(this.refs.id.value),
•       "eventName":this.refs.eventName.value,
•       "location":this.refs.location.value,
•       "guest":this.refs.guest.value,
•       "date":this.refs.date.value
•     };
•     let url = "http://localhost:3000/events";
•     fetch(url,{
•       method:'POST',
•       headers:{'content-type':'application/json'},
•       body:JSON.stringify(event)
•     })
•     .then(response=>response.json())
```

```

    .then(() => this.setState({msg:'Row added....'}));
  }

  EditEvents(){
    let event = {
      "id": Number(this.refs.id.value),
      "eventName":this.refs.eventName.value,
      "location":this.refs.location.value,
      "guest":this.refs.guest.value,
      "date":this.refs.date.value
    };
    let url = `http://localhost:3000/events/${event.id}`;
    fetch(url,{
      method:'PUT',
      headers:{'content-type':'application/json'},
      body:JSON.stringify(event)
    })
    .then(response=>response.json())
    .then(() => this.setState({msg:'Row updated....'}));
  }

  DeleteEvents(){
    let id =Number(this.refs.id.value);
    let url = `http://localhost:3000/events/${id}`;
    fetch(url,{
      method:'DELETE'
    })
    .then(response=>response.json())
    .then(() => this.setState({msg:'Row deleted....'}));
  }

  render(){
    var event = this.state.events;
    return(
      <div className="container">
        <button onClick = {() =>
this.logoutUser()}>Logout</button> <br/>
        <div className="wrapper fadeInDown">

          <div><div className="divEvent">

```

```

•         <input type="number" id="login" className="form-
control psswd" name="login" ref="id" placeholder="Id"/>
•         <input type="text" id="eventName" className="form-
control psswd" name="eventName" ref="eventName" placeholder="eventName"/>
•         <input type="text" id="location" className="form-
control psswd" name="location" ref="location" placeholder="location"/>
•         <input type="text" id="guest" className="form-
control psswd" name="guest" ref="guest" placeholder="guest name"/>
•         <input type="date" id="date" className="form-
control psswd" name="date" ref="date" placeholder="DATE"/>
•         </div>
•         <div>
•             <input type="submit" value="Add Events"
onClick={()=>this.AddEvents()}/>
•             <input type="submit" value="Edit Events"
onClick={()=>this.EditEvents()}/>
•             <input type="submit" value="Delete Events"
onClick={()=>this.DeletEvents()}/>
•         </div>
•
•         </div>
•
•         </div>
•
•         <div>
•             <table className="table table-bordered table-hover">
•                 <thead>
•                     <tr>
•                         <th>ID</th> <th>Name of the event</th>
• <th>location</th> <th>guest</th> <th>date</th>
•                     </tr>
•                 </thead>
•
•                 <tbody>
•                     {event.map(p => <tr><td>{p.id}</td>
•                         <td>{p.eventName}</td>
•                         <td>{p.location}</td>
•                         <td>{p.guest}</td>
•                         <td>{p.date}</td>
•                     </tr>
•                     )}
•                 </tbody>
•             </table>
•         </div>
•     )
• }

```

- }
-
- `export default EventManager;`

- Route.js

```
import React from 'react'
import {BrowserRouter as Router, Route, Switch, Link} from 'react-router-dom'
import UserComponent from './UserComponent'
import EventManager from './EventManager'

class EventRoute extends React.Component{
  render(){
    return (
      <div>
        <h1>My Event Finder</h1>

        <Router>

          { /* <Link to="/login">Login</Link>| */ }
          { /* <Link to="/people">People</Link>| */ }

          <Switch>
            <Route exact path="/">
              <UserComponent/>
            </Route>
            <Route exact path="/dashboard">
              <EventManager/>
            </Route>
            <Route exact path="/logout">
              <UserComponent/>
            </Route>

          </Switch>

        </Router>
      </div>
    )
  }
}
export default EventRoute;
```

- UserComponent.js

```

• import React from 'react';
• import '../src/login.css'
• import { Redirect } from 'react-router-dom'
• import {store, login, logout} from '../sessionStore'
•
• class UserComponent extends React.Component{
•
•     state = {msg: '',
•             status: '',
•             users: [
•                 {'uname': 'admin', 'pwd': 'admin'},
•                 {'uname': 'user', 'pwd': 'password'},
•                 {'uname': 'test', 'pwd': 'user'}
•             ]
•         }
•
•         // state={msg:'',status:false}
•         validateUser(){
•             var uname = this.refs.uname.value;
•             var pwd = this.refs.pwd.value;
•
•             // if ( uname=="admin" && pwd=="1234"){
•             //     this.setState({msg:'Credentials found correct...'});
•             //     this.setState({status:true});
•             // }
•             // else
•             //     this.setState({msg:'Invalid input credentials'});
•             if (this.state.users.find( x => x.uname === uname && x.pwd ===
pwd)) {
•                 console.log("welcome.....")
•                 store.dispatch(login(uname))
•                 this.setState({msg: 'Credentials Validated.....',
status:true})
•             }
•             else {
•                 store.dispatch(logout())
•                 this.setState({msg: 'Invalid Credentials.....', status:false})
•             }
•         }
•         render(){
•             if ( this.state.status)
•                 return <Redirect to ="/dashboard" />
•             return(
•                 <div className="container">
•                     <div id="formContent">

```

```

•
•
•       <div className="container fa-stack-2x">
•           
•           </div>
•
•
•
•       <div>
•           <input type="text" id="login" className="fadeIn
second" name="login" ref="uname" placeholder="login"/>
•           <input type="password" id="password"
className="fadeIn third form-control psswd" name="password" ref="pwd"
placeholder="password"/>
•           <input type="submit" className="fadeIn fourth"
value="Log In" onClick={()=>this.validateUser()} />
•       </div>
•
•       <div className="alert alert-danger">
•           <strong>{this.state.msg}</strong>
•       </div>
•
•
•       </div>
•   </div>
•   )
•   }
•   }
•
•   export default UserComponent

```

- App.js

```

import React from 'react';
import logo from './logo.svg';
import './App.css';

import EventManager from './Component/EventManager';
import UserComponent from './Component/UserComponent';
import Route from './Component/Route';

function App() {
  return (
    <div className="App">
      { /* <EventManager/> */ }
      <Route/>
    </div>
  );
}

```

```

    </div>
  );
}

export default App;

```

- Login.css

```

•
•  /* BASIC */
•
•  html {
•    background-color: #56baed;
•  }
•
•  body {
•    font-family: "Poppins", sans-serif;
•    height: 100vh;
•  }
•
•  a {
•    color: #92badd;
•    display:inline-block;
•    text-decoration: none;
•    font-weight: 400;
•  }
•
•  h2 {
•    text-align: center;
•    font-size: 16px;
•    font-weight: 600;
•    text-transform: uppercase;
•    display:inline-block;
•    margin: 40px 8px 10px 8px;
•    color: #cccccc;
•  }
•
•  .psswd{
•
•    height: 60px;
•    background-color: #f6f6f6;
•    border: none;
•    color: #0d0d0d;
•    padding: 15px 32px;
•    text-align: center;

```

```

•   text-decoration: none;
•   display: inline-block;
•   font-size: 16px;
•   margin: 5px;
•   width: 85%;
•   border: 2px solid #f6f6f6;
•   transition: all 0.5s ease-in-out;
•   border-radius: 5px 5px 5px 5px;
• }
•
•   .divEvent{
•       display: inline-flex;
•   }
•
•   /* STRUCTURE */
•
•   .wrapper {
•       display: flex;
•       align-items: center;
•       flex-direction: column;
•       justify-content: center;
•       width: 100%;
•       min-height: 100%;
•       padding: 20px;
•   }
•
•   #formContent {
•       -webkit-border-radius: 10px 10px 10px 10px;
•       border-radius: 10px 10px 10px 10px;
•       background: #fff;
•       padding: 30px;
•       width: 90%;
•       max-width: 450px;
•       position: relative;
•       padding: 0px;
•       -webkit-box-shadow: 0 30px 60px 0 rgba(0,0,0,0.3);
•       box-shadow: 0 30px 60px 0 rgba(0,0,0,0.3);
•       text-align: center;
•   }
•
•   #formFooter {
•       background-color: #f6f6f6;
•       border-top: 1px solid #dce8f1;
•       padding: 25px;
•       text-align: center;

```



```

•   -webkit-border-radius: 0 0 10px 10px;
•   border-radius: 0 0 10px 10px;
•   }
•
•
•
•
•   /* TABS */
•
•   h2.inactive {
•       color: #cccccc;
•   }
•
•   h2.active {
•       color: #0d0d0d;
•       border-bottom: 2px solid #5fbae9;
•   }
•
•
•
•   /* FORM TYPOGRAPHY*/
•
•   input[type=button], input[type=submit], input[type=reset] {
•       background-color: #56baed;
•       border: none;
•       color: white;
•       padding: 15px 80px;
•       text-align: center;
•       text-decoration: none;
•       display: inline-block;
•       text-transform: uppercase;
•       font-size: 13px;
•       -webkit-box-shadow: 0 10px 30px 0 rgba(95,186,233,0.4);
•       box-shadow: 0 10px 30px 0 rgba(95,186,233,0.4);
•       -webkit-border-radius: 5px 5px 5px 5px;
•       border-radius: 5px 5px 5px 5px;
•       margin: 5px 20px 40px 20px;
•       -webkit-transition: all 0.3s ease-in-out;
•       -moz-transition: all 0.3s ease-in-out;
•       -ms-transition: all 0.3s ease-in-out;
•       -o-transition: all 0.3s ease-in-out;
•       transition: all 0.3s ease-in-out;
•   }
•
•   input[type=button]:hover, input[type=submit]:hover,
input[type=reset]:hover {

```

```

    background-color: #39ace7;
}

input[type=button]:active, input[type=submit]:active,
input[type=reset]:active {
    -moz-transform: scale(0.95);
    -webkit-transform: scale(0.95);
    -o-transform: scale(0.95);
    -ms-transform: scale(0.95);
    transform: scale(0.95);
}

input[type=text] {
    background-color: #f6f6f6;
    border: none;
    color: #0d0d0d;
    padding: 15px 32px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-size: 16px;
    margin: 5px;
    width: 85%;
    border: 2px solid #f6f6f6;
    -webkit-transition: all 0.5s ease-in-out;
    -moz-transition: all 0.5s ease-in-out;
    -ms-transition: all 0.5s ease-in-out;
    -o-transition: all 0.5s ease-in-out;
    transition: all 0.5s ease-in-out;
    -webkit-border-radius: 5px 5px 5px 5px;
    border-radius: 5px 5px 5px 5px;
}

input[type=text]:focus {
    background-color: #fff;
    border-bottom: 2px solid #5fbae9;
}

input[type=text]:placeholder {
    color: #cccccc;
}

/* ANIMATIONS */

```

```

•
•   /* Simple CSS3 Fade-in-down Animation */
•   .fadeInDown {
•       -webkit-animation-name: fadeInDown;
•       animation-name: fadeInDown;
•       -webkit-animation-duration: 1s;
•       animation-duration: 1s;
•       -webkit-animation-fill-mode: both;
•       animation-fill-mode: both;
•   }
•
•   @-webkit-keyframes fadeInDown {
•       0% {
•           opacity: 0;
•           -webkit-transform: translate3d(0, -100%, 0);
•           transform: translate3d(0, -100%, 0);
•       }
•       100% {
•           opacity: 1;
•           -webkit-transform: none;
•           transform: none;
•       }
•   }
•
•   @keyframes fadeInDown {
•       0% {
•           opacity: 0;
•           -webkit-transform: translate3d(0, -100%, 0);
•           transform: translate3d(0, -100%, 0);
•       }
•       100% {
•           opacity: 1;
•           -webkit-transform: none;
•           transform: none;
•       }
•   }
•
•   /* Simple CSS3 Fade-in Animation */
•   @-webkit-keyframes fadeIn { from { opacity:0; } to { opacity:1; } }
•   @-moz-keyframes fadeIn { from { opacity:0; } to { opacity:1; } }
•   @keyframes fadeIn { from { opacity:0; } to { opacity:1; } }
•
•   .fadeIn {
•       opacity:0;
•       -webkit-animation:fadeIn ease-in 1;

```

```

•     -moz-animation:fadeIn ease-in 1;
•     animation:fadeIn ease-in 1;
•
•     -webkit-animation-fill-mode:forwards;
•     -moz-animation-fill-mode:forwards;
•     animation-fill-mode:forwards;
•
•     -webkit-animation-duration:1s;
•     -moz-animation-duration:1s;
•     animation-duration:1s;
• }
•
• .fadeIn.first {
•     -webkit-animation-delay: 0.4s;
•     -moz-animation-delay: 0.4s;
•     animation-delay: 0.4s;
• }
•
• .fadeIn.second {
•     -webkit-animation-delay: 0.6s;
•     -moz-animation-delay: 0.6s;
•     animation-delay: 0.6s;
• }
•
• .fadeIn.third {
•     -webkit-animation-delay: 0.8s;
•     -moz-animation-delay: 0.8s;
•     animation-delay: 0.8s;
• }
•
• .fadeIn.fourth {
•     -webkit-animation-delay: 1s;
•     -moz-animation-delay: 1s;
•     animation-delay: 1s;
• }
•
• /* Simple CSS3 Fade-in Animation */
• .underlineHover:after {
•     display: block;
•     left: 0;
•     bottom: -10px;
•     width: 0;
•     height: 2px;
•     background-color: #56baed;
•     content: "";

```

```

•   transition: width 0.2s;
•   }
•
•   .underlineHover:hover {
•       color: #0d0d0d;
•   }
•
•   .underlineHover:hover:after{
•       width: 100%;
•   }
•
•
•
•
•   /* OTHERS */
•
•   *:focus {
•       outline: none;
•   }
•
•   #icon {
•       width:60%;
•   }
•

```

• EventApi.json

```

•   {
•       "events": [
•           {
•               "id": 2,
•               "eventName": "new event",
•               "location": "Bangalore",
•               "guest": "simplilearn",
•               "date": "2021-12-22"
•           },
•           {
•               "id": 123,
•               "eventName": "changes done",
•               "location": "Chennai",
•               "guest": "Me",
•               "date": "2021-12-31"
•           },
•           {
•               "id": 1,
•               "eventName": "testing simpli",
•               "location": "Bangalore",

```

```
•      "guest": "simplilearn",
•      "date": ""
•    }
•  ]
• }
```