

Index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';
import '../node_modules/font-awesome/css/font-awesome.min.css';
import reportWebVitals from './reportWebVitals';

ReactDOM.render(
  <App />,
  document.getElementById('root'));

reportWebVitals();
```

context.js

```
import { render } from '@testing-library/react';
import React, {Component} from 'react';
const Context = React.createContext();

const reducer = (state, action) => {
  switch (action.type) {
    case 'DELETE_EVENTS':
      return{
        ...state,
        eventss: state.eventss.filter(events => events.id !== action.payload)
      };
    /******For ADD Content */
    case 'ADD_EVENTS':
      return{
        ...state,
        eventss: [action.payload, ...state.eventss]
      };
    default:
      return state;
  }
};

export class Provider extends Component{
  state = {
```

```

    eventss: [
      {
        id: 1,
        title: 'Researching ',
        scheduled: '2:00pm',
        period: '20/02/2021'
      },
      {
        id: 2,
        title: 'Training ',
        scheduled: '7:00pm',
        period: '20/02/2021'
      },
      {
        id: 3,
        title: 'Hangout ',
        scheduled: '2:00pm',
        period: '21/09/2021'
      }
    ],
    dispatch: action =>
      this.setState(state => reducer(state, action))

  };

  render(){
    return (
<Context.Provider value= {this.state}>
  {this.props.children}
</Context.Provider>
    )
  }
}

export const Consumer = Context.Consumer;

// this file holds the application level state

```

App.test.js

```

import { render, screen } from '@testing-library/react';
import App from './App';

```

```
test('renders learn react link', () => {
  render(<App />);
  const linkElement = screen.getByText(/learn react/i);
  expect(linkElement).toBeInTheDocument();
});
```

App.js

```
import React, { Component } from 'react';
import Eventss from './components/Eventss';
import Header from './components/Header';

import AddEvents from './components/AddEvents';

import { Provider } from './context';
import 'bootstrap/dist/css/bootstrap.min.css';
import './App.css';

class App extends Component {
  render(){
    return(
      <Provider>
        <div className="App">

          <Header branding="Find MyEvent"/>
          <div className="container">
            <AddEvents />
            <Eventss />

          </div>
        </div>
      </Provider>
    );
  }
}

export default App;
```

Header.js

```
import React from 'react'
```

```

import PropTypes from 'prop-types';

const Header = props => {
  const { branding } = props;
  return (
    <nav className="navbar navbar-expand-sm navbar-dark bg-success mb-3 py-0" >
      <div className="container">
        <a href="/" className="navbar-brand">{branding}</a>
        <div>
          <ul className="navbar-nav mr-auto">
            <li className="nav-item">
              <a href="/" className="nav-link">Home</a>
            </li>

            </ul>
          </div>
        </div>
      </nav>
    );
};

Header.defaultProps = {
  branding: 'My App'
};

Header.propTypes = {
  branding: PropTypes.string.isRequired
};
export default Header;

```

Eventss.js

```

import React, { Component } from 'react'
import Events from './Events';
import { Consumer } from '../context'

class Eventss extends Component {

  //state is called here

  render() {
    return (
      <Consumer>

```

```

        {value => {
            const { eventss } = value;
            return (<React.Fragment>
                {value.eventss.map(events => (
                    <Events
                        key={events.id}
                        events={events}
                    />
                ))}
            </React.Fragment>
        )
    }}
</Consumer>
//this value is coming from the context provider
)
}
}

export default Eventss;

```

Events.js

```

import React, { Component } from 'react'
import PropTypes from 'prop-types';
import { Consumer } from '../context';

class Events extends Component {
    state = {
        showEventsInfo: false
    };

    onDeleteClick = (id, dispatch) => {
        dispatch({type: 'DELETE_EVENTS', payload: id});
    };

    render() {
        const { id, title, scheduled, period } = this.props.events;
        const { showEventsInfo } = this.state;
    }
}

```

```

    return(
      <Consumer>
        {value => {
          const { dispatch } = value;
          return(
            <div className="card card-body mb-3">
              <h4>{title}{' '}
                {/* for dropdown */}
                <i onClick={() => this.setState({ showEventsInfo: !this.state.showEventsInfo })} className="fa fa-caret-down" style={{ cursor: 'pointer' }} />
                {/* for delete button */}
                <i className="fa fa-trash" style={{ cursor: 'pointer', float: 'right', color: 'red' }}
                  onClick={this.onDeleteClick.bind(this, id, dispatch)}
                />

              </h4>

              {showEventsInfo ?
                (<ul className="list-group">
                  <li className="list-group-item">Scheduled: {scheduled}</li>
                  <li className="list-group-item">Period: {period}</li>
                </ul>) : null}

              </div>
            );
          }}
        </Consumer>
      );
    }
  }
Events.propTypes = {
  events: PropTypes.object.isRequired
};

export default Events;

```

AddEvents.js

```
import React, { Component } from 'react'
import { Consumer } from '../context';
// import uuid from 'uuid';

class AddEvents extends Component {
  //The way you labelled the state must be the way you label the "name value in the form"
  state = {
    title: '',
    scheduled: '',
    period: ''
  };

  onSubmit = (dispatch, e) => {
    e.preventDefault();
    //***** install uuid in your vscode terminal npm i uuid because this is used to generate id */
    const
      { title, scheduled, period } = this.state;
    const NewEvents = {

      // id: uuid(),
      title ,
      scheduled ,
      period
    };

    dispatch({ type: 'ADD_EVENTS', payload: NewEvents});
  };

  onChange = e => this.setState({ [e.target.name]: e.target.value });
  render() {

    const { title, scheduled, period } = this.state;

    return(
      <Consumer>
        {value => {
```

```

const { dispatch } = value;
return(
  <div className="card mb-3">
    <div className="card-header">
      Add Events
    </div>

    <div className="card-body">
      <form onSubmit={this.onSubmit.bind(this, dispatch)}>
<div className="form-group">
  <label htmlFor="title">Title</label>
  <input type="text" name="title" className="form-control form-control-
lg" placeholder="Enter Title..."
  value={title}
  onChange={this.onChange}
  />
</div>
<div className="form-group">
  <label htmlFor="scheduled">Scheduled</label>
  <input type="Scheduled"
  name="scheduled" className="form-control form-control-
lg" placeholder="dd/mm/yyyy"
  value={scheduled}
  onChange={this.onChange}/>
</div>
<div className="form-group">
  <label htmlFor="period">Period</label>
  <input type="Period" name="period" className="form-control form-control-
lg"
  value={period}
  onChange={this.onChange}/>

</div>

  <input type="submit" value="Add Event" className="btn btn-
success btn-block"/>
    </form>
  </div>

</div>
  )
}
</Consumer>
)
}

```



```
}  
  
export default AddEvents;
```

AddEvent_ref.js

```
import React, { Component } from 'react'  
  
class AddEvents extends Component {  
  constructor(props) {  
    super(props);  
  
    this.titleInput = React.createRef();  
    this.scheduledInput = React.createRef();  
    this.periodInput = React.createRef();  
  }  
  
  onSubmit = e => {  
    e.preventDefault();  
    const events = {  
      title: this.titleInput.current.value,  
      scheduled: this.scheduledInput.current.value,  
      period: this.periodInput.current.value  
    };  
    console.log(events);  
  };  
  
  static defaultProps = {  
    title: 'Staff Party',  
    scheduled: '23/02/2021',  
    period: '3:30pm'  
  };  
  
  render() {  
  
    const { title, scheduled, period } = this.props;  
  
    return (  
      <div className="card mb-3">  
        <div className="card-header">
```

```

        Add Events
    </div>

    <div className="card-body">
        <form onSubmit={this.onSubmit}>
            <div className="form-group">
                <label htmlFor="title">Title</label>
                <input type="text" name="title" className="form-
control form-control-lg" placeholder="Enter Title..."
                    defaultValue={title}
ref={this.titleInput}
                />
            </div>
            <div className="form-group">
                <label htmlFor="scheduled">Scheduled</label>
                <input type="Scheduled"
                    name="scheduled" className="form-control form-
control-lg" placeholder="dd/mm/yyyy"
                    defaultValue={scheduled}
                    ref={this.scheduledInput}
                />
            </div>
            <div className="form-group">
                <label htmlFor="period">Period</label>
                <input type="Period" name="period" className="form-
control form-control-lg"
                    defaultValue={period}
                    ref={this.periodInput}
                />
            </div>
            <input type="submit" value="Add Event" className="btn btn
-success btn-block" />
        </form>
    </div>

    </div>
    );
}
}

export default AddEvents;

```