

## Day 18 Revisit

- State & Props – Props read only properties, state – private to components.
- Props is used for inter component communication.
- State is used to handle dynamic contents.
- State changes re-render react component.
- State can be changed with the help of setState() method only.
- Virtual & Actual DOM
- React Forms
- Class based Component (Stateful)/ function based component (stateless)
- Adding third party styling library with react components
- Added bootstrap library (npm I bootstrap, add link/script in html or import it in index/app.js)
- Each React component needs to be properly closed.
- Redering List using simple arrays
- Created customerList component which read a json file and renders the content as a dynamic html table.
- Styled that table using bootstrap
- Created two more child components in customerList namely customerForm, customerDetails.
- Both customerList and customerForm are class based components (stateful)
- customerDetails is a function based component (stateless)
- Discussed and saw a demo for parent to child communication using props.
- Discussed and saw a demo for child to parent communication using state and props
- Component Life Cycle methods and different phases of components life cycle
- Mount Phase, Update Phase, Unmount phase
- componentDidMount(), componentDidUpdate(), componentWillUnmount()

## Day 19 Agenda

- React Forms, Hooks
- Error Boundaries
- Validations, Error Handling in Forms
- HOC (Higher Order Components)
- React.lazy and Suspense for CSR/SSR
- Handling Events in React
- Reacting to User action in React
- Advanced React JS
  - Using pure functions as components
  - Custom middleware

- ◆ `<> </>`
- ◆ Container tags
- ◆ Nesting Element (Embedding (adding) element inside another element)
- ◆ Getter method is also called as accessor (Allows to access the property)
- ◆ Setter method is also called as mutator (Allows to set to a new value)

Async Method Call

- Call back function.
- Promise – Promising the result not immediately but after sometime. It always gives response. Response can be success/error.
- Observables

Lazy Loading – lazy(load)

HOC -- **a higher-order component is a function that takes a component and returns a new component.**

<https://legacy.reactjs.org/docs/higher-order-components.html>

Hooks Ref -- <https://react.dev/reference/react/hooks>

CSR – Client Side Rendering

SSR – Server Side Rendering

Mvn clean install

Mvn package -DskipTests

Java -jar target\<jarfile\_name>

Use commandlinerunner

[Post Request] Create a savings bank account for customer.

[Post Request] Create account object using the id of prev inserted customer. Type : SAVING, balance : 1000\$

Deposit – Put request -balance (existing balance + deposit\_amount)

Withdraw – Put request – updatedBalance = (existingBalance – withdraw\_amount)