

| Title: Implementation of react concept |
| --- |

**AIM:** To demonstrate the working of React.

**Problem Definition:**

Demonstrate the use of different concept of React on the basis of following points

1. Function Component
2. Styling/ Bootstrap
3. React JSX
4. Expressions in JSX
5. React Props
6. React state
7. React Component Lifecycle
8. React Events
9. Event Binding

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Expected OUTCOME of Experiment:

. **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

1. Express .js Deep API reference, by Azat Marden, Apress, 2nd edition, 2015.
2. <https://codeburst.io/building-a-rest-api-using-mongo-db-75cac3403fab>
3. <https://www.edureka.co/blog/rest-api-with-node-js/>
4. https://bezkoder.com/node-express-mongodb-crud-rest-api/

**Pre Lab/ Prior Concepts:**

* **Function Component**: Simpler, stateless or stateful via hooks.
* **Styling/Bootstrap**: Use Bootstrap classes for styling; add Bootstrap CSS via CDN or npm.
* **React JSX**: Write HTML-like code inside JavaScript using JSX syntax.
* **Expressions in JSX**: Use curly braces {} to embed JavaScript expressions in JSX.
* **React Props**: Pass data to components using props.
* **React State**: Manage component state using the useState hook.
* **React Component Lifecycle**: Use useEffect for lifecycle management in function components.
* **React Events**: Handle events using camelCase syntax.
* **Event Binding**: Bind event handlers either directly or in the constructor (for class components).

**Implementation Details:**

Function Component:ItemList and App are function components. They return JSX to render UI.

* Styling/Bootstrap:Applied Bootstrap classes like list-group, list-group-item, btn, and container for styling.
* React JSX:Used JSX to define how the components should be rendered. Example: <ul className="list-group">.
* Expressions in JSX: {items.map((item, index) => (<li key={index} className="list-group-item">{item}</li>))} dynamically generates list items based on state.
* React Props:Passed items and onAddItem as props from App to ItemList.
* React State:Managed the state of items in the App component using useState.
* React Component Lifecycle:Used useEffect in ItemList to log when the component mounts and unmounts.
* React Events:Attached an onClick event handler to the button in ItemList.
* Event Binding:Bound the handleAddItem function to the button click event.

**import React, { useState, useEffect } from 'react';**

**function ItemList({ items, onAddItem }) {**

**useEffect(() => {**

**console.log('ItemList component mounted');**

**return () => {**

**console.log('ItemList component unmounted');**

**};**

**}, []);**

**return (**

**<div>**

**<ul className="list-group">**

**{items.map((item, index) => (**

**<li key={index} className="list-group-item">**

**{item}**

**</li>**

**))}**

**</ul>**

**<button className="btn btn-primary mt-3" onClick={onAddItem}>**

**Add Item**

**</button>**

**</div>**

**);**

**}**

**function App() {**

**const [items, setItems] = useState(['Item 1', 'Item 2']);**

**const handleAddItem = () => {**

**setItems([...items, ` Item ${items.length + 1} added`]);**

**};**

**return (**

**<div className="container mt-5">**

**<h1 className="mb-4">Item List</h1>**

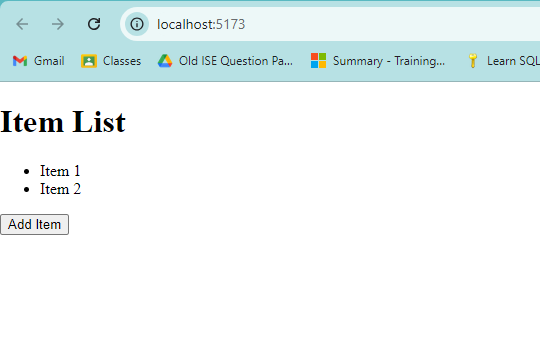
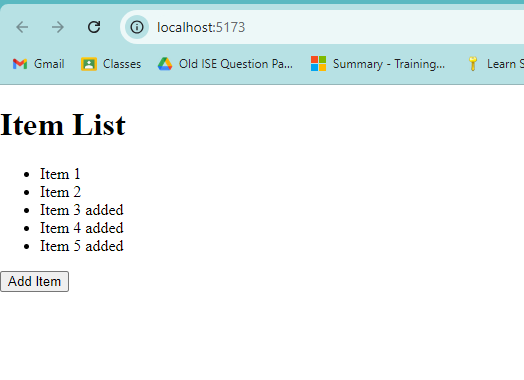
**<ItemList items={items} onAddItem={handleAddItem} />**

**</div>**

**);**

**}**

**export default App;**

**  
**

**Conclusion:**

Learned react fundamentals.