ASSIGNMENT NO :- 4

Aim :- Enhance web page developed in earlier assignment by rendering Lists and Portals, Error Handling, Routers and style with React CSS also make it a responsive design to a scale well across PC tablet and Mobile phone.

Objective :-

- Enhance User Interface and Experience
- Improve Application Robustness and Navigation.

Theory :-

Q1] How do Lists and keys work in React?

→ Lists : Render dynamic arrays of elements using map().

Keys : Unique identifiers for elements in a list, helping React track changes, additions, and removals.

Key Matters because it ensure efficient updates and prevent bugs

Ex:-

```
Const items = [{id: 1, name: "Item 1'}, {id: 2, name:
'Item 2' }];
Const listItems = items.map(item}=> (
  <li key= {item. id}> {item.name} </li>
));
```

Q2) What is a React Portal and when would you use one?

→ Portal : A way to render children into a DOM node outside the parent component's hierarchy.
Use Cases:
1) Modals
2) Tooltips
3) Overlays.

It prevents styling issues and ensure proper layering

Q3) Discuss the importance of Error Boundaries in React.

→ Error Boundary is a component that catches Java-script errors in its child tree and display a fallback UI.

- Importance:

1) Prevent app crashes
2) Provides a better user experience
3) Helps with debugging.

Implement static
getDerivedStateFromError() and/or componentDidCatch().

Q4] How does React Router enable Single Page Application [SPA] functionality?

→ React Router enables Client-side routing allowing users to navigate between views without full page reloads.

How:-

1) Routes : Define paths and components to render
2) Link : Navigate between routes.
3) useParams, useLocation : Access route parameters and query settings.

Ex:-

```
import {Browser Router, Routes, Route, Link} from 'react-router-dom';
function App() {
return (
<Browser Router>
<Link to ="/home" >Home </link>
<Routes>
<Route path="/home" element={<Home />} />
<Route path = "/about" element = {<About />} />
</Routes>
</Browser Router>
);
}
```

Page No.
Date

Q5) Explain the different ways to style a React applicatio

→ • Inline Styles : Use the style attribute with a JavaScript object.

• CSS Modules : import CSS files as modules to avoid naming conflicts.

• Styled Components : Use libraries like styled-components for CSS-in-JS.

• Sass/SCSS : Use preprocessors for more powerful Styling.

• Utility Libraries : Tailwind CSS for utility-first Styling.

13/10/25