**WEEK 6**

**6.ReactJS-HOL**

**OBJECTIVES:**

**1. Explain the Need and Benefits of React Router**

React Router is a standard routing library for React applications that enables navigation between different components without refreshing the page.

**🔹 Need for React Router:**

* In Single Page Applications (SPA), navigation must happen dynamically without reloading the entire page.
* React by itself doesn't provide routing — so React Router is used to simulate multiple "pages" in the app.
* Helps build a multi-view experience while keeping the app performant and responsive.

**🔹 Benefits:**

* Dynamic routing based on application state and structure
* Preserves UI state when navigating between components
* Enables parameter passing via URLs
* Supports nested routing, redirects, and route protection
* Works well with code splitting and lazy loading for performance.

**2. Identify the Components in React Router**

React Router includes several key components that make routing possible:

| **Component** | **Description** |
| --- | --- |
| <BrowserRouter> | The router container that uses HTML5 history API for clean URLs |
| <Routes> | A wrapper for all <Route> elements — matches one route at a time |
| <Route> | Defines a specific route and the component to render when path matches |
| <Link> | Used to navigate between routes without full page reload |
| <Navigate> | Redirects to another route programmatically |
| <useParams> | Hook to access dynamic URL parameters |
| <useNavigate> | Hook to navigate programmatically within components |

**3. List the Types of Router Components**

There are multiple types of router components provided by React Router depending on the environment:

| **Router Type** | **Description** |
| --- | --- |
| <BrowserRouter> | Most common; uses **clean URLs** via the browser history API |
| <HashRouter> | Uses **hash-based URLs** (#/route) — useful for static file hosting |
| <MemoryRouter> | Stores history in memory (no URL changes) — mainly used in tests or non-browser envs |
| <StaticRouter> | Used for **server-side rendering (SSR)** |

**4. Parameter Passing via URL**

React Router allows you to define **dynamic segments** in the URL using route parameters. These can be accessed via the useParams() hook.

**🔹 Step-by-step:**

* 1. **Define a dynamic route:**

<Route path="/user/:userId" element={<User />} />

* 1. **Create a link to it:**

<Link to="/user/42">View User</Link>

* 1. **Access parameter in component:**

import { useParams } from 'react-router-dom';

function User() {

const { userId } = useParams();

return <h3>User ID: {userId}</h3>;

}