1. Programmatic Navigation in React Router

 What is Programmatic Navigation?
Programmatic navigation in React Router allows you to navigate between pages dynamically **without using <Link> or <NavLink> components**. It is useful for:

- Redirecting users after login or logout.
- Navigating based on user actions (e.g., button clicks).
- Handling conditional navigation.

 Example of Programmatic Navigation Using useNavigate (React Router v6+)

```
import { useNavigate } from 'react-router-dom'; const Home = () => { const navigate = useNavigate(); const goToDashboard = () => { navigate('/
dashboard'); }; return <button onClick={goToDashboard}>Go to Dashboard</button>; };
```

---

2. useHistory vs. useNavigate for Navigation

React Router introduced **useNavigate in v6**, replacing the older **useHistory from v5**.

Feature	useHistory (v5)	useNavigate (v6)
Import Path	import { useHistory } from 'react-router-dom';	import { useNavigate } from 'react-router-dom';
Navigation Function	history.push('/path');	navigate('/path');
Go Back / Forward	history.goBack(); history.goForward();	navigate(-1); navigate(1);
State Passing	history.push('/path', { state: data });	navigate('/path', { state: data });

Example: Migrating useHistory (v5) to useNavigate (v6)

```
// React Router v5 (Old) const history = useHistory(); history.push('/dashboard'); // React Router v6 (New) const navigate = useNavigate();
navigate('/dashboard');
```

When to Use useNavigate?

- Always use useNavigate for new React Router v6+ projects.
- Use it for **navigating to new pages, passing state, or handling back/forward navigation**.

When to Use useHistory?

- Only if working with **React Router v5** (but upgrading is recommended).

---

3. useParams Hook in React Router

▯ What is useParams?

useParams is a hook that allows you to **access route parameters** in a component.

▯ Example Scenario: Extracting Route Parameters

Consider a **dynamic route** where a user profile page depends on a userId in the URL:

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

▯ Using useParams to Extract userId

```
import { useParams } from 'react-router-dom'; const UserProfile = () => { const { userId } = useParams(); // Extract userId from
URL return <h1>Welcome, User {userId}!</h1>; };
```

If the URL is /user/123, useParams() returns { userId: "123" }.

□ Real-World Use Cases of useParams

- Fetching user details from an API:

  ```
  useEffect(() => { fetch(`https://api.example.com/users/${userId}`) .then(response => response.json()) .then(data => setUser(data)); },
  [userId]);
  ```

- Displaying product details (/products/:productId).
- Navigating to blog posts (/blog/:slug).
