**1. Explain the Need and Benefits of React Context API**

**Need for Context API:**

In React, props are used to pass data from parent to child. But when deeply nested components need the same data, **prop drilling** (passing props through many levels) becomes inefficient.

**Example Problem:**

<App>

<Parent>

<Child>

<GrandChild> ← Needs user data

Instead of passing props manually through each level, **Context API** allows sharing data globally across components.

**Benefits of Context API:**

* Avoids prop drilling.
* Centralized and consistent data management.
* Simple alternative to Redux for small-medium apps.
* Useful for themes, user authentication, language settings, etc.

**2. Working with createContext()**

The createContext() function is used to create a Context object in React.

**Basic Steps:**

**Step 1: Create the context**

import React, { createContext } from 'react';

const UserContext = createContext(); // Creates context

**Step 2: Provide context value**

<UserContext.Provider value={{ name: "Gobika" }}>

<MyComponent />

</UserContext.Provider>

**Step 3: Consume context**

import { useContext } from 'react';

const { name } = useContext(UserContext);

You can also use <UserContext.Consumer> in class components.

**3. List the Types of Router Components (React Router v6)**

React Router provides several components for building routing in single-page applications:

| **Component** | **Description** |
| --- | --- |
| <BrowserRouter> | Wraps the app and enables routing using HTML5 history API |
| <HashRouter> | Uses URL hash for routing (#/home) – useful for static apps |
| <Routes> | A container for all <Route> components |
| <Route> | Defines the path and component to render |
| <Link> | Used for navigation without reloading the page |
| <NavLink> | Like <Link>, but applies active styles when matched |
| <Navigate> | Redirects to a different route |
| <Outlet> | Placeholder for nested routes |
| <useParams> | Hook to get route parameters |
| <useNavigate> | Hook to navigate programmatically |
| <useLocation> | Hook to access the current location object |