React Advanced (Lecture - 2)

[Context API]

Definition -

The **Context API** in React is used to **share data globally** between components **without passing props manually at every level** (avoids *prop drilling*).

Steps to Use Context API

1 Create Context

```js
// context/AppContext.js

import { createContext } from "react";

export const AppContext = createContext();
```

2 Create a Provider

The Provider wraps components that need access to the data.

```jsx
// context/AppContextProvider.jsx

import { useState } from "react";
import { AppContext } from "./AppContext";

export function AppContextProvider({ children }) {
  const [user, setUser] = useState("Vikas");

  return (
    <AppContext.Provider value={{ user, setUser }}>
      {children}
    </AppContext.Provider>
  );
}
```

3 Wrap App with Provider

```jsx
// main.jsx or App.jsx

import React from "react";
import ReactDOM from "react-dom/client";
import App from "./App";
import { AppContextProvider } from "./context/AppContextProvider";

ReactDOM.createRoot(document.getElementById("root")).render(
  <AppContextProvider>
    <App />
  </AppContextProvider>
);
```

4 Consume Context in Any Component

```
// components/UserProfile.jsx

import { useContext } from "react";
import { AppContext } from "../context/AppContext";

function UserProfile() {
  const { user, setUser } = useContext(AppContext);

  return (
    <div>
      <h2>Welcome, {user}</h2>
      <button onClick={() => setUser("Rahul")}>
        Change User
      </button>
    </div>
  );
}

export default UserProfile;
```

How Context Works (In Simple Words)

- `createContext()` → creates a global container
- `Provider` → supplies data
- `useContext()` → consumes data
- When data changes → all subscribed components re-render

When to Use Context API

✅ Theme (dark/light mode)

✅ Authentication (user, token)

✅ Cart data (small to medium apps)

❌ Not ideal for very complex state logic (Redux/Zustand better)

When to Use Context API

Props	Context
Passed component to component	Direct access
Hard to manage deep trees	Cleaner
Better for small data	Best for global data

When to Use Context API

🛒 **E-commerce App**

- Context for `Cart` , `User` , `Theme`
- Redux only if logic becomes complex