Day 14 – [8th July 2025]

TOPICS COVERED

Props in React:

Props (short for "properties") are used to pass data from one component to another in React. They make components reusable and dynamic.

Example:

```
function Greeting(props) {
  return <h2>Hello, {props.name}!</h2>;
}
function App() {
  return <Greeting name="Navpreet" />;
}
```

name is a prop passed to the Greeting component, and it is accessed using props.name.

useEffect Hook:

The useEffect() hook lets us run side effects in functional components.

It is often used for fetching data, DOM manipulation, timers, etc.

Basic syntax:

```
useEffect(() => {
  // code to run after render
}, [dependencies]);
```

Runs after the first render and when dependencies change

If dependency array is empty [], it runs only once like componentDidMount

Example:

```
import { useEffect } from "react";
useEffect(() => {
  console.log("Component mounted");
}, []);
```

Fetching Data from API in React:

We often use fetch() along with useEffect() to load data when the component mounts.

Example:

```
import { useState, useEffect } from "react";
function App() {
  const [users, setUsers] = useState([]);
  useEffect(() => {
    fetch("https://jsonplaceholder.typicode.com/users")
      .then(res => res.json())
      .then(data => setUsers(data));
  }, []);
  return (
    <div>
      <h1>Users List</h1>
      <ul>
        {users.map(user => (
          <li key={user.id}>{user.name}</li>
        ))}
      </ul>
    </div>
  );
}
```

In this example:

Data is fetched once when the component mounts

The user list is stored in state using useState

The data is rendered using .map() in JSX

Rendering with .map() in JSX:

The .map() function is commonly used to render lists of elements in React.

Example:

```
const fruits = ["Apple", "Banana", "Cherry"];
return (
```

```
  <ul>
    {fruits.map((fruit, index) => (
      <li key={index}>{fruit}</li>
    ))}
  </ul>
);
```

.map() creates a list of <li> elements from the array.

TOOLS USED:

Visual Studio Code

Vite + React

Chrome Browser + DevTools

JSONPlaceholder API (for dummy user data)

TASK:

Practice API Calling using React.

14_Day14_code.jsx

```
import { useState, useEffect } from "react";
import UserCard from "./UserCard";
function App() {
  const [users, setUsers] = useState([]);
  useEffect(() => {
    // Fetch user data
    async function getUsers() {
      const response = await fetch("https://fakestoreapi.com/users");
      const data = await response.json();
      setUsers(data);
    }
    getUsers();
  }, []);
```

```
  return (
    <div style={{
      backgroundColor: "lavender",
      minHeight: "100vh",
      padding: "30px",
      fontFamily: "Arial"
    }}>
      <h1 style={{ textAlign: "center", color: "purple" }}>User List</h1>
      <div style={{
        display: "flex",
        flexWrap: "wrap",
        justifyContent: "center",
        gap: "20px",
        marginTop: "30px"
      }}>
        {users.map(user => (
          <UserCard key={user.id} name={user.name.firstname + " " + user.name.lastname} email={user.email} />
        ))}
      </div>
    </div>
  );
}
export default App;
```

14_Day14_codecard.jsx

```
function UserCard({ name, email }) {
  return (
    <div style={{
      backgroundColor: "white",
      padding: "16px",
      border: "2px solid lightgray",
      borderRadius: "10px",
```

```
      width: "220px",
      textAlign: "center",
      boxShadow: "0 4px 8px gray"
    }}>
      <h3 style={{ color: "darkblue", marginBottom: "8px" }}>{name}</h3>
      <p style={{ color: "black" }}>{email}</p>
    </div>
  );
}
export default UserCard;
```