📋 Lists in React

In React, rendering a list of elements (like users, cards, posts, etc.) is done using **JavaScript array methods**, most commonly `.map()`.

✅ Basic Example

jsx   Copy   Edit

```jsx
function NameList() { const names = ['Vivek', 'Anita', 'Ravi']; return ( <ul> {names.map((name,
index) => ( <li key={index}>{name}</li> ))} </ul> ); }
```

🔑 Why `key` is important?

Each child in a list needs a **unique** `key` **prop** so React can efficiently re-render only what changes.

jsx   Copy   Edit

```jsx
{items.map(item => ( <li key={item.id}>{item.name}</li> ))}
```

🛑 Don't use array index as key if items can change or be reordered.

🧠 List of Objects Example

jsx   Copy   Edit

```jsx
const users = [ { id: 1, name: 'Vivek', age: 22 }, { id: 2, name: 'Priya', age: 25 }, { id: 3,
name: 'Amit', age: 24 }, ]; function UserList() { return ( <div> {users.map((user) => ( <div key=
{user.id}> <h4>{user.name}</h4> <p>Age: {user.age}</p> </div> ))} </div> ); }
```

🧪 Rendering Components from a List

jsx

```jsx
function UserCard({ name, age }) { return ( <div className="card"> <h3>{name}</h3> <p>Age: {age}
</p> </div> ); } function UserList() { const users = [ { id: 1, name: 'Vivek', age: 22 }, { id: 2,
name: 'Priya', age: 25 }, ]; return ( <div> {users.map((user) => ( <UserCard key={user.id} name=
{user.name} age={user.age} /> ))} </div> ); }
```

```jsx
function UserCard({ name, age }) { return ( <div className="card"> <h3>{name}</h3> <p>Age: {age}
</p> </div> ); } function UserList() { const users = [ { id: 1, name: 'Vivek', age: 22 }, { id: 2,
name: 'Priya', age: 25 }, ]; return ( <div> {users.map((user) => ( <UserCard key={user.id} name=
{user.name} age={user.age} /> ))} </div> ); }
```