---

1. List in React

- List = Array of elements displayed dynamically
- Example:

```
const users = ["Sunil", "Aman", "Rakesh"];
<ul>
  {users.map((user, index) => (
    <li key={index}>{user}</li>
  ))}
</ul>
```

---

2. Map Function in React

- `map()` iterates over array and returns JSX elements
- Syntax: `array.map((element, index) => <JSX_Element />)`
- Example:

```
const numbers = [1,2,3];
const listItems = numbers.map(num => <li>{num*2}</li>);
```

---

3. Keys in React

- Special prop to uniquely identify each element in a list
- Improves **performance** and **efficient re-rendering**
- Example:

```
<ul>
  {users.map(user => <li key={user.id}>{user.name}</li>)}
</ul>
```

- Rule: Keys should be **unique and stable**

---

4. When and Why Use List, Map, and Keys

Feature	When to Use	Why
List	Display multiple similar elements	Dynamic rendering of array
Map	Convert array to JSX	Iterate array efficiently
Keys	Render list items	Efficient re-rendering & unique identification

---

5. Real-Life Example – Dynamic Users Table

```
const users = [
  {id:1, name:'Sunil', age:25},
  {id:2, name:'Aman', age:30},
  {id:3, name:'Rakesh', age:28}
];

<table border='1'>
  <thead>
    <tr><th>ID</th><th>Name</th><th>Age</th></tr>
  </thead>
  <tbody>
    {users.map(user => (
      <tr key={user.id}>
        <td>{user.id}</td>
        <td>{user.name}</td>
        <td>{user.age}</td>
      </tr>
    ))}
  </tbody>
</table>
```

---

6. Advanced CRUD Example – Dynamic Users List

App.js

```
import React, { useState } from 'react';
import UserCard from './UserCard';
import AddUserForm from './AddUserForm';

function App() {
  const [users, setUsers] = useState([
    {id:1, name:'Sunil', age:25},
```

```
    {id:2, name:'Aman', age:30},
    {id:3, name:'Rakesh', age:28},
  ]);

  const addUser = (user) => setUsers([...users, {...user, id: Date.now()}]);
  const removeUser = (id) => setUsers(users.filter(user=>user.id!==id));
  const editUser = (id, newName, newAge) =>
setUsers(users.map(user=>user.id===id ? {...user,name:newName,age:newAge} :
user));

  return (
    <div style={{padding:'20px'}}>
      <h1>Dynamic Users List</h1>
      <AddUserForm addUser={addUser} />
      {users.map(user=>(
        <UserCard key={user.id} user={user} removeUser={removeUser}
editUser={editUser} />
      ))}
    </div>
  );
}

export default App;
```

UserCard.js

```
import React,{useState} from 'react';

function UserCard({user, removeUser, editUser}) {
  const [isEditing,setIsEditing]=useState(false);
  const [name,setName]=useState(user.name);
  const [age,setAge]=useState(user.age);

  const handleSave = () => {
    editUser(user.id,name,parseInt(age));
    setIsEditing(false);
  };

  return (
    <div style={{border:'1px solid
gray',borderRadius:'5px',padding:'10px',marginBottom:'10px',width:'300px'}}>
      {isEditing ? (
        <div>
          <input type='text' value={name}
onChange={e=>setName(e.target.value)} style={{marginRight:'10px'}}/>
          <input type='number' value={age}
onChange={e=>setAge(e.target.value)} style={{marginRight:'10px'}}/>
          <button onClick={handleSave}>Save</button>
        </div>
      ) : (
```

```
        <div>
          <h2>{user.name}</h2>
          <p>Age: {user.age}</p>
          <button onClick={()=>setIsEditing(true)}>Edit</button>
          <button onClick={()=>removeUser(user.id)}
style={{marginLeft:'10px'}}>Remove</button>
        </div>
      )}
    </div>
  );
}

export default UserCard;
```

AddUserForm.js

```
import React,{useState} from 'react';

function AddUserForm({addUser}) {
  const [name,setName]=useState('');
  const [age,setAge]=useState('');

  const handleSubmit = (e) => {
    e.preventDefault();
    if(!name || !age) return alert('Please fill all fields');
    addUser({name, age:parseInt(age)});
    setName('');
    setAge('');
  }

  return (
    <form onSubmit={handleSubmit} style={{marginBottom:'20px'}}>
      <input type='text' placeholder='Name' value={name}
onChange={e=>setName(e.target.value)} style={{marginRight:'10px'}}/>
      <input type='number' placeholder='Age' value={age}
onChange={e=>setAge(e.target.value)} style={{marginRight:'10px'}}/>
      <button type='submit'>Add User</button>
    </form>
  );
}

export default AddUserForm;
```

---

7. Features Demonstrated

1. Dynamic List rendering using `map()`
2. Unique `key` prop for each item

3. Add, Remove, Edit Users – full CRUD
4. Props – parent → child function passing
5. State Management – parent state for users, child state for editing

---

8. Conclusion

- **List + Map** = Render multiple elements dynamically
- **Keys** = Efficient update & identification
- **Props + Functions** = Parent-child communication
- **State** = Manage dynamic content

Perfect practical guide for React List, Map, Keys & CRUD.