React Teaching Material: Lists and map()

Session 1: Rendering Lists using map()

Goal: Teach students how to render lists in React using the map() method.

PART 1: Introduction & Theory (20 min)

Why use .map() in React?

- Rendering dynamic content from arrays
- Cleaner and more maintainable code
- Separates logic from UI rendering

Basic Syntax:

```
array.map((item, index) => {
  return <li key={index}>{item}</li>;
});
```

Example:

```
const fruits = ['Banana', 'Apple', 'Orange'];
function FruitList() {
  return (
    <ul>
      {fruits.map((fruit, index) => (
        <li key={index}>{fruit}</li>
      ))}
    </ul>
  );
}
```

Important Notes:

- Each element must have a unique key prop.
- key helps React identify which items have changed.
- Avoid using index as key in real-world apps if the list is editable.

PART 2: Practice Exercises (25 min)

React Teaching Material: Lists and map()

Activity 1: Render Static List

```
const animals = ['Dog', 'Cat', 'Elephant'];
function AnimalList() {
  return (
    <ul>
      {animals.map((animal, index) => (
        <li key={index}>{animal}</li>
      ))}
    </ul>
  );
}
```

Activity 2: Render Objects

```
const students = [
  { id: 1, name: 'Arta' },
  { id: 2, name: 'Blerim' },
  { id: 3, name: 'Dona' }
];
function StudentList() {
  return (
    <ul>
      {students.map(student => (
        <li key={student.id}>{student.name}</li>
      ))}
    </ul>
  );
}
```

Session 2: Hands-on: Displaying a List Dynamically

Goal: Build a functional React component that lets users add and view a list of items.

PART 1: Live Coding Walkthrough (20 min)

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

React Teaching Material: Lists and map()

```
function TodoList() {
  const [tasks, setTasks] = useState([]);
  const [newTask, setNewTask] = useState('');

  const handleAddTask = () => {
    if (newTask.trim() === '') return;
    setTasks([...tasks, newTask]);
    setNewTask('');
  };

  return (
    <div>
      <h2>To-Do List</h2>
      <input
        type="text"
        placeholder="Enter a task"
        value={newTask}
        onChange={(e) => setNewTask(e.target.value)}
      />
      <button onClick={handleAddTask}>Add</button>

      <ul>
        {tasks.map((task, index) => (
          <li key={index}>{task}</li>
        ))}
      </ul>
    </div>
  );
}
```

PART 2: Student Tasks (25 min)

Task 1 - Create a shopping list component

Input field + Add button

React Teaching Material: Lists and map()

Render list below

Reset input field after adding

Task 2 - Extend with delete button

Add a delete button next to each item

On click, remove that item from the list