**Objectives Covered**

| **Objective** | **How It’s Implemented** |
| --- | --- |
| Various ways of **conditional rendering** | if-else, ternary, logical &&, switch rendering |
| **Render multiple components** | Conditional inclusion of multiple components |
| Define **List Component** | Create lists with map() |
| Use of **keys** | Provide unique keys for list items |
| Extract components with keys | Render child components from lists using keys |
| Use of **map()** | To iterate and render items in list-based components |

**Step 1: Create App**

npx create-react-app bloggerapp

cd bloggerapp

npm start

**Step 2: Create Components**

**BookDetails.js**

import React from 'react';

const BookDetails = ({ books }) => {

if (!books || books.length === 0) return <p>No books available</p>;

return (

<div>

<h2>Book Details</h2>

<ul>

{books.map((book, index) => (

<li key={book.id}>

<strong>{book.title}</strong> - {book.author}

</li>

))}

</ul>

</div>

);

};

export default BookDetails;

**BlogDetails.js**

import React from 'react';

const BlogDetails = ({ blogs }) => {

return blogs.length ? (

<div>

<h2>Blog Details</h2>

{blogs.map((blog) => (

<div key={blog.id}>

<h4>{blog.title}</h4>

<p>{blog.content}</p>

</div>

))}

</div>

) : null; // Conditional rendering using return null

};

export default BlogDetails;

**CourseDetails.js**

jsx

CopyEdit

import React from 'react';

const CourseDetails = ({ courses }) => (

<div>

<h2>Course Details</h2>

{courses.map(course => (

<div key={course.code}>

<p><strong>{course.name}</strong> - {course.code}</p>

</div>

))}

</div>

);

export default CourseDetails;

**Step 4: Main Component App.js**

import React, { useState } from 'react';

import BookDetails from './components/BookDetails';

import BlogDetails from './components/BlogDetails';

import CourseDetails from './components/CourseDetails';

function App() {

const [view, setView] = useState('books'); // current selected view

const books = [

{ id: 1, title: 'React Basics', author: 'Dan Abramov' },

{ id: 2, title: 'Advanced React', author: 'Kent Dodds' },

];

const blogs = [

{ id: 1, title: 'React Hooks Guide', content: 'Understanding useState and useEffect' },

{ id: 2, title: 'JSX Deep Dive', content: 'JSX is a syntax extension for JavaScript.' },

];

const courses = [

{ code: 'RE101', name: 'React Essentials' },

{ code: 'JS202', name: 'JavaScript Mastery' },

];

// Conditional rendering using if-else

let content;

if (view === 'books') {

content = <BookDetails books={books} />;

} else if (view === 'blogs') {

content = <BlogDetails blogs={blogs} />;

} else if (view === 'courses') {

content = <CourseDetails courses={courses} />;

} else {

content = <p>Select a section above.</p>;

}

return (

<div style={{ padding: '20px' }}>

<h1>Blogger App</h1>

{/\* Buttons to toggle views \*/}

<button onClick={() => setView('books')}>Books</button>{' '}

<button onClick={() => setView('blogs')}>Blogs</button>{' '}

<button onClick={() => setView('courses')}>Courses</button>

<hr />

{/\* Render selected component \*/}

{content}

{/\* Additional conditional rendering using ternary operator \*/}

<p style={{ marginTop: '20px', color: view === 'blogs' ? 'blue' : 'gray' }}>

{view === 'blogs' ? 'You are viewing Blogs' : 'Not viewing Blogs'}

</p>

{/\* Logical && operator \*/}

{view === 'courses' && <p>Courses are in demand!</p>}

</div>

);

}

export default App;

**✅ Conditional Rendering Techniques Used**

| **Technique** | **Where Used** |
| --- | --- |
| if-else | To determine which main component to render |
| ternary | view === 'blogs' ? ... : ... |
| && operator | view === 'courses' && <p>Courses are in demand</p> |
| return null | In BlogDetails when no blogs available |

**Key Concepts Covered**

| **Concept** | **Implementation** |
| --- | --- |
| **List with keys** | map() in all components with unique key |
| **Extracting components with keys** | BookDetails, BlogDetails, CourseDetails |
| **Rendering multiple components** | Conditional display using view state |
| **React map() function** | Used in all list-rendering components |

**Output Summary**

* Dynamically switches between Book, Blog, and Course views.
* Clean use of conditional rendering techniques.
* Lists rendered using map() and key.
* null returned to skip rendering empty blog section.