## **Objectives**

* Explain various ways of conditional rendering
* Explain how to render multiple components
* Define list component
* Explain about keys in React applications
* Explain how to extract components with keys
* Explain React Map, map() function

In this hands-on lab, you will learn how to:

* Implement conditional rendering in React applications

## **Prerequisites**

The following is required to complete this hands-on lab:

* Node.js
* NPM
* Visual Studio Code

## **Notes**

Estimated time to complete this lab: **60 minutes.**

Create a React App named “bloggerapp” in with 3 components.

1. Book Details
2. Blog Details
3. Course Details

Implement this with as many ways possible of Conditional Rendering.

**Program:**

**Data.js:**

export const books = [

{ id: 1, title: "React Essentials", author: "Dan Abramov" },

{ id: 2, title: "Learning JS", author: "Kyle Simpson" }

];

export const blogs = [

{ id: 1, title: "Intro to React", date: "2025-01-01" },

{ id: 2, title: "Hooks Explained", date: "2025-02-14" }

];

export const courses = [

{ id: 1, name: "React for Beginners", duration: "4 weeks" },

{ id: 2, name: "Advanced React", duration: "6 weeks" }

];

**BookDetails.js:**

import React from 'react';

function BookDetails({ books }) {

return (

<div>

<h2>Book Details</h2>

<ul>

{books.map(book => (

<li key={book.id}>

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

</li>

))}

</ul>

</div>

);

}

export default BookDetails;

**BlogDetails.js:**

import React from 'react';

function BlogDetails({ blogs }) {

return (

<div>

<h2>Blog Details</h2>

<ul>

{blogs.map(blog => (

<li key={blog.id}>

{blog.title} – {blog.date}

</li>

))}

</ul>

</div>

);

}

export default BlogDetails;

**CourseDetails.js:**

import React from 'react';

function CourseDetails({ courses }) {

return (

<div>

<h2>Course Details</h2>

<ul>

{courses.map(course => (

<li key={course.id}>

{course.name} – Duration: {course.duration}

</li>

))}

</ul>

</div>

);

}

export default CourseDetails;

**ContentToggle.js:**

import React, { useState } from 'react';

import BookDetails from './BookDetails';

import BlogDetails from './BlogDetails';

import CourseDetails from './CourseDetails';

import { books, blogs, courses } from './data';

function ContentToggle() {

const [type, setType] = useState('books');

let content;

if (type === 'books') {

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

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

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

} else {

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

}

return (

<div style={{ textAlign: 'center', marginTop: 40 }}>

<h1>BloggerApp</h1>

{/\* Button Controls \*/}

<div style={{ marginBottom: 20 }}>

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

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

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

</div>

{/\* Conditional rendering using variable \*/}

{content}

{/\* Bonus: conditional using ternary \*/}

<p>{type === 'courses' ? "You’re viewing Courses." : "Select Courses to view course details."}</p>

{/\* Conditional using && operator \*/}

{type === 'blogs' && <p>These are blog entries!</p>}

</div>

);

}

export default ContentToggle;

**App.js:**

import React from 'react';

import ContentToggle from './ContentToggle';

function App() {

return (

<div>

<ContentToggle />

</div>

);

}

export default App;

**Output:**



