## **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

**src/components/BookDetails.js**

import React from "react";

export default function BookDetails() {

const books = [

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

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

];

return (

<div>

<h2>Book Details</h2>

<ul>

{books.map(book => (

<li key={book.id}>{book.title} — {book.author}</li>

))}

</ul>

</div>

);

}

**src/components/BlogDetails.js**

import React from "react";

export default function BlogDetails() {

const blogs = [

{ id: 1, title: "React Conditional Rendering", date: "2025-08-10" },

{ id: 2, title: "Understanding Hooks", date: "2025-08-12" },

];

return (

<div>

<h2>Blog Details</h2>

<ul>

{blogs.map(blog => (

<li key={blog.id}>{blog.title} — {blog.date}</li>

))}

</ul>

</div>

);

}

**src/components/CourseDetails.js**

import React from "react";

export default function CourseDetails() {

const courses = [

{ id: 1, name: "React for Beginners", duration: "6 weeks" },

{ id: 2, name: "Advanced JavaScript", duration: "8 weeks" },

];

return (

<div>

<h2>Course Details</h2>

<ul>

{courses.map(course => (

<li key={course.id}>{course.name} — {course.duration}</li>

))}

</ul>

</div>

);

}

**src/App.js:**

import React, { useState } from "react";

import BookDetails from "./components/BookDetails";

import BlogDetails from "./components/BlogDetails";

import CourseDetails from "./components/CourseDetails";

export default function App() {

const [view, setView] = useState("book");

// Method 1: Using if-else

const renderIfElse = () => {

if (view === "book") return <BookDetails />;

else if (view === "blog") return <BlogDetails />;

else if (view === "course") return <CourseDetails />;

else return <p>Select a valid view</p>;

};

// Method 2: Using ternary operator

const renderTernary = () =>

view === "book" ? <BookDetails /> :

view === "blog" ? <BlogDetails /> :

view === "course" ? <CourseDetails /> :

<p>No matching content</p>;

// Method 3: Using element variables

let content;

if (view === "book") content = <BookDetails />;

else if (view === "blog") content = <BlogDetails />;

else content = <CourseDetails />;

return (

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

<h1>Blogger App</h1>

<div style={{ marginBottom: 15 }}>

<button onClick={() => setView("book")}>Show Books</button>

<button onClick={() => setView("blog")}>Show Blogs</button>

<button onClick={() => setView("course")}>Show Courses</button>

</div>

<h3>Rendered with If-Else</h3>

{renderIfElse()}

<h3>Rendered with Ternary</h3>

{renderTernary()}

<h3>Rendered with Element Variable</h3>

{content}

</div>

);

}

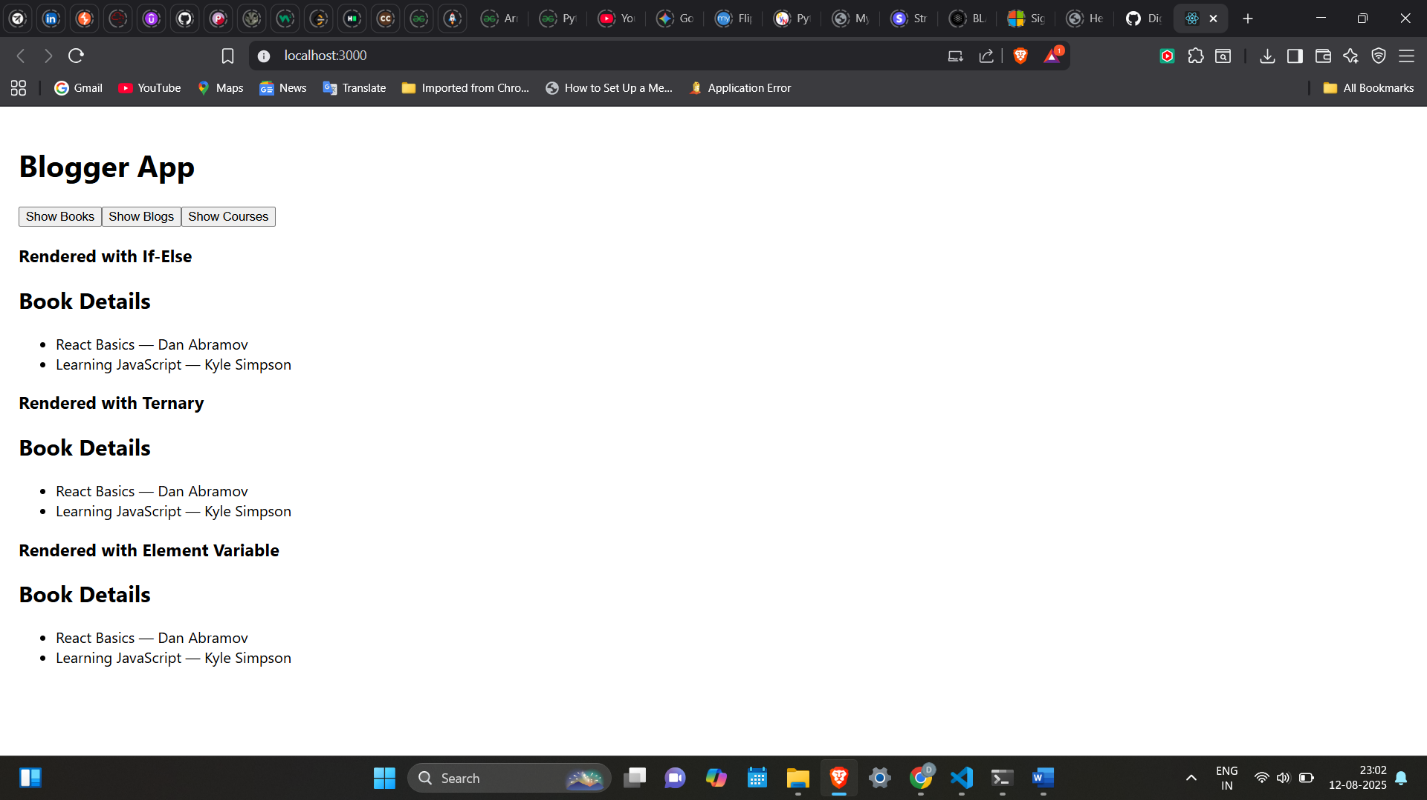
## **Notes**

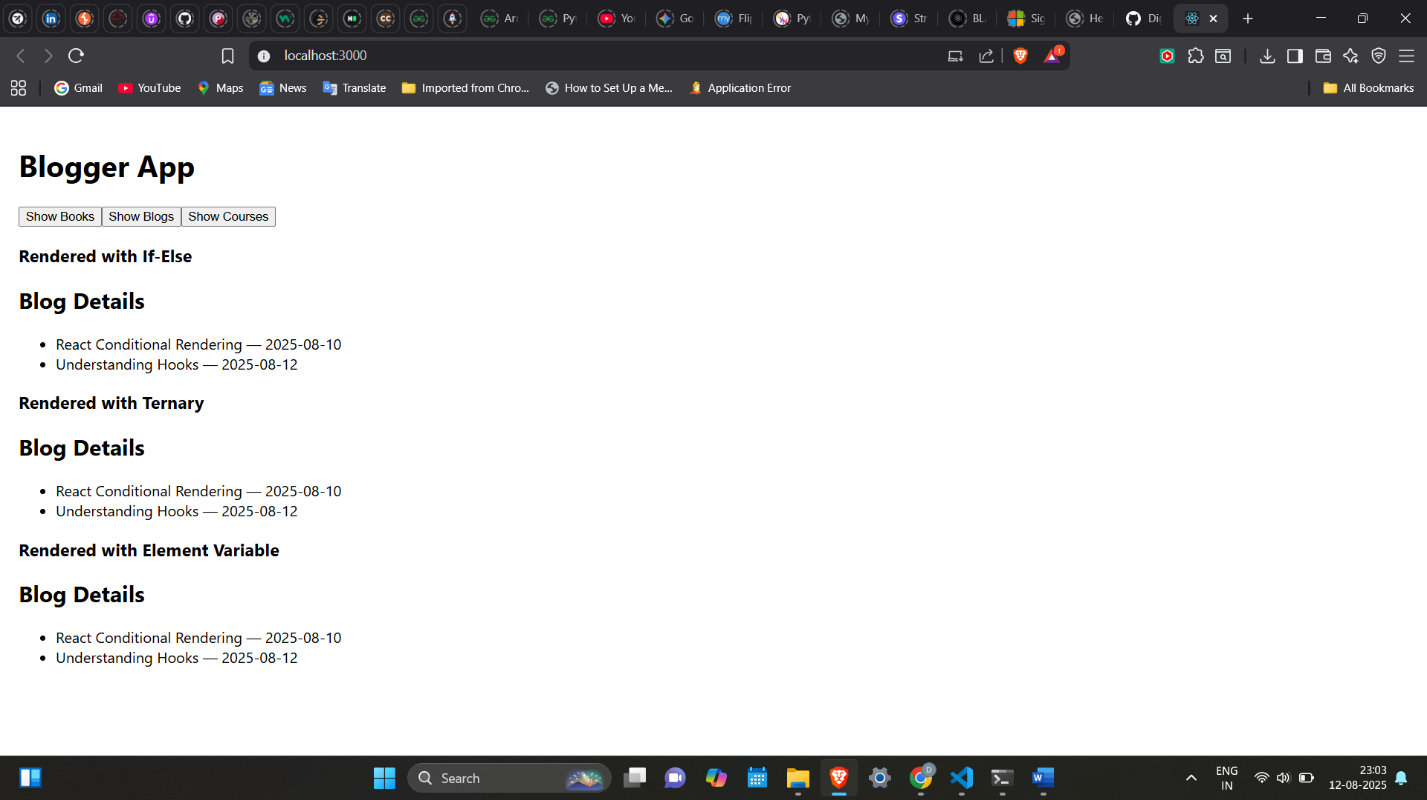
Estimated time to complete this lab: **60 minutes.**

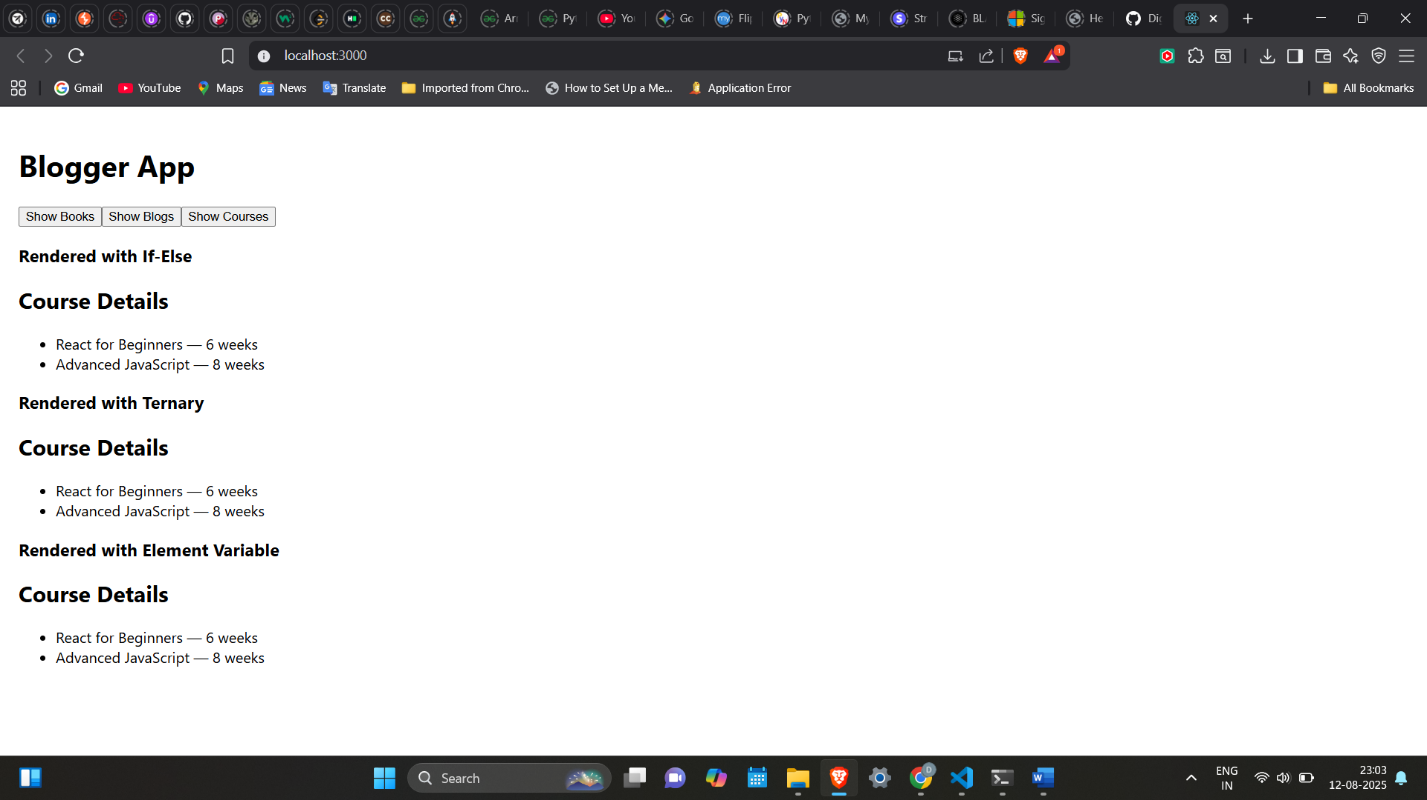
Create a React App named “bloggerapp” in with 3 components.

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

Implement this with as many ways possible of Conditional Rendering.







**Hint:**





