**Exercise 13: Create a React App named “bloggerapp”**

**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**

**data.js**

// components/data.js

export const books = [

{ id: 101, bname: 'Master React', price: 670 },

{ id: 102, bname: 'Deep Dive into Angular 11', price: 800 },

{ id: 103, bname: 'Mongo Essentials', price: 450 },

];

export const blogs = [

{ id: 1, title: 'React Learning', author: 'Stephen Biz', content: 'Welcome to learning React!' },

{ id: 2, title: 'Installation', author: 'Schewzdenier', content: 'You can install React from npm.' },

];

export const courses = [

{ id: 1, name: 'Angular', date: '4/5/2021' },

{ id: 2, name: 'React', date: '6/3/2021' },

];

**BookDetails.js**

import React from 'react';

function BookDetails({ books }) {

return (

<div className="st2">

<h1>Book Details</h1>

{books.length ? (

books.map((book) => (

<div key={book.id}>

<h3>{book.bname}</h3>

<h4>{book.price}</h4>

</div>

))

) : (

<p>No books found.</p>

)}

</div>

);

}

export default BookDetails;

**BlogDetails.js**

import React from 'react';

function BlogDetails({ blogs }) {

return (

<div className="v1">

<h1>Blog Details</h1>

{blogs.length > 0 && (

<ul>

{blogs.map((blog) => (

<li key={blog.id}>

<h3>{blog.title}</h3>

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

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

</li>

))}

</ul>

)}

</div>

);

}

export default BlogDetails;

**CourseDetails.js**

import React from 'react';

function CourseDetails({ courses, showCourses }) {

if (!showCourses) return null;

return (

<div className="mystyle1">

<h1>Course Details</h1>

{courses.map((course) => (

<div key={course.id}>

<p>{course.name}</p>

<p>{course.date}</p>

</div>

))}

</div>

);

}

export default CourseDetails;

**Step 3: Main Component App.js**

import React, { useState } from 'react';

import BookDetails from './components/BookDetails';

import BlogDetails from './components/BlogDetails';

import CourseDetails from './components/CourseDetails';

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

function App() {

const [showCourses, setShowCourses] = useState(true);

return (

<div>

<h1>Blogger App</h1>

<BookDetails books={books} />

<BlogDetails blogs={blogs} />

<button onClick={() => setShowCourses(!showCourses)}>

{showCourses ? 'Hide Courses' : 'Show Courses'}

</button>

<CourseDetails courses={courses} showCourses={showCourses} />

</div>

);

}

export default App;

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.



