**WEEK-7**

**React JS HOL-13**

* Explain various ways of conditional rendering?  
  In React, we can render content conditionally using if/else statements, ternary operators, logical AND, or by storing JSX in a variable and returning it based on conditions.
* Explain how to render multiple components?  
  You can render multiple components by placing them one after the other inside a parent element like a <div>. You can also return an array of components or use fragments (<> </>) to avoid extra DOM nodes.
* Define list component?  
  A list component is a component that displays a collection of similar items (like users, products, etc.). It usually maps over an array and renders each item inside JSX elements like <li> or <div>.
* Explain about keys in React applications?  
  Keys are unique identifiers used when rendering lists. They help React identify which items have changed, added, or removed, improving performance during re-renders.
* Explain how to extract components with keys?  
  When mapping a list, it's good practice to move each list item into a separate child component and pass the unique key prop to it. This keeps the code clean and lets React track the individual components efficiently.
* Explain React Map, map() function?  
  In React, map() is a JavaScript method used to loop through arrays and return a new array of elements. We use it in JSX to create dynamic lists, such as rendering a list of users or posts from an array.

**App.js**

import { useState } from "react";

import BookDetails from "./components/BookDetails";

import BlogDetails from "./components/BlogDetails";

import CourseDetails from "./components/CourseDetails";

import "./App.css";

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", description: "Welcome to learning React!" },

  { id: 2, title: "Installation", author: "Schwezdnier", description: "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" },

];

function App() {

  const [showBooks, setShowBooks] = useState(true);

  const [showBlogs, setShowBlogs] = useState(true);

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

  return (

    <div className="container">

      {showBooks && <BookDetails books={books} />}

      {showBlogs ? <BlogDetails blogs={blogs} /> : <h1>No Blogs Available</h1>}

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

    </div>

  );

}

export default App;

**App.css**

body {

  background-color: white;

  color: black;

  margin: 0;

  font-family: Arial, sans-serif;

}

.container {

  display: flex;

  justify-content: space-around;

  padding: 50px;

}

.st2, .v1, .mystyle1 {

  border-right: 5px solid green;

  padding: 20px;

  width: 30%;

}

.st2 h1, .v1 h1, .mystyle1 h1 {

  font-size: 24px;

  margin-bottom: 20px;

  font-weight: bold;

}

.st2 h3, .v1 h3, .mystyle1 h3 {

  font-size: 18px;

  font-weight: bold;

}

.st2 h4, .v1 h4, .mystyle1 h4 {

  font-size: 16px;

  color: #333;

}

.v1 p {

  font-size: 14px;

  color: #555;

}

**CourseDetails.js**

function CourseDetails(props) {

  const { show, courses } = props;

  if (!show) {

    return <h1>No Courses Available</h1>;

  }

  const coursedet = (

    <ul>

      {courses.map((course) => (

        <div key={course.id}>

          <h3>{course.name}</h3>

          <h4>{course.date}</h4>

        </div>

      ))}

    </ul>

  );

  return (

    <div className="mystyle1">

      <h1>Course Details</h1>

      {coursedet}

    </div>

  );

}

export default CourseDetails;

**BlogDetails.js**

function BlogDetails(props) {

  const content = props.blogs.map((blog) => (

    <div key={blog.id}>

      <h3>{blog.title}</h3>

      <h4>{blog.author}</h4>

      <p>{blog.description}</p>

    </div>

  ));

  return (

    <div className="v1">

      <h1>Blog Details</h1>

      {content}

    </div>

  );

}

export default BlogDetails;

**BookDetails.js**

function BookDetails(props) {

  const bookdet = (

    <ul>

      {props.books.map((book) => (

        <div key={book.id}>

          <h3>{book.bname}</h3>

          <h4>{book.price}</h4>

        </div>

      ))}

    </ul>

  );

  return (

    <div className="st2">

      <h1>Book Details</h1>

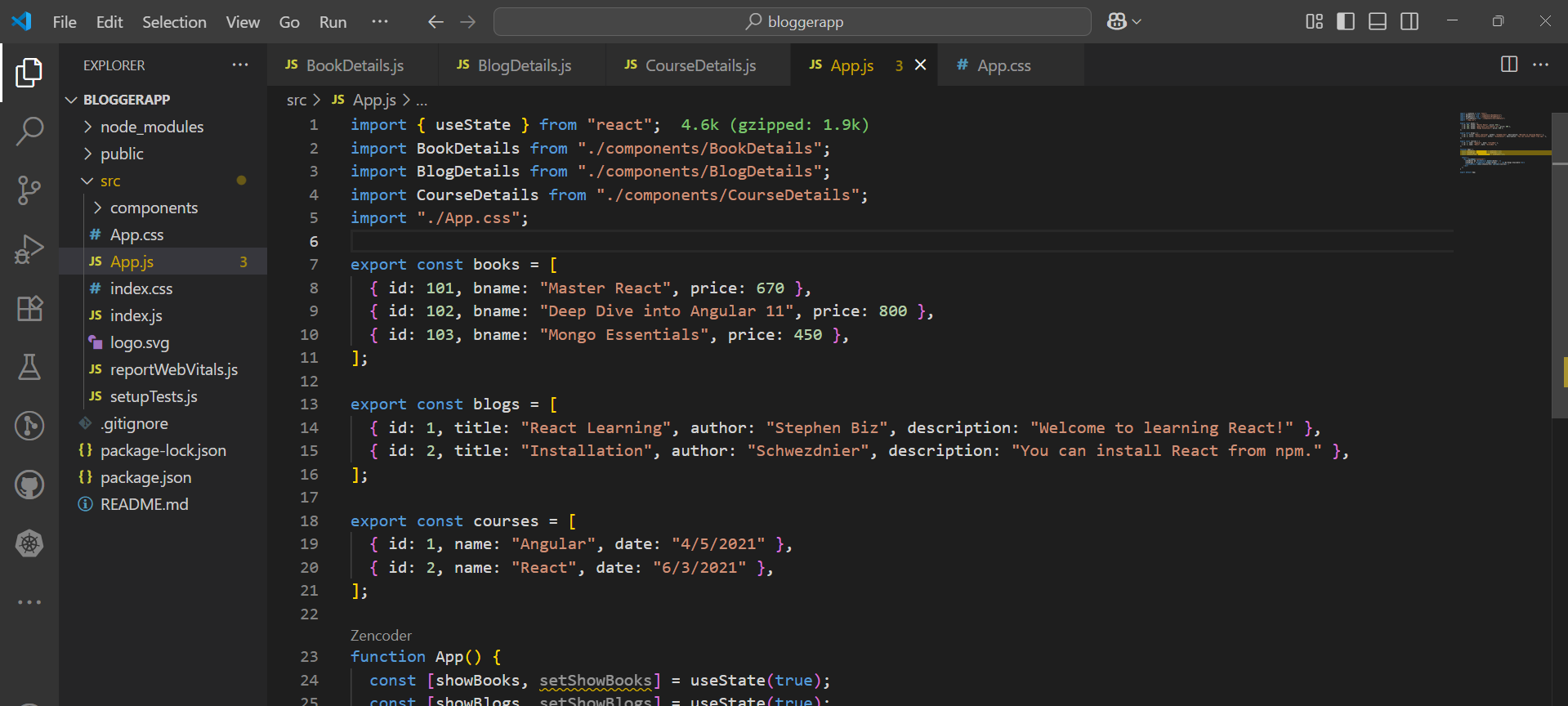
      {bookdet}

    </div>

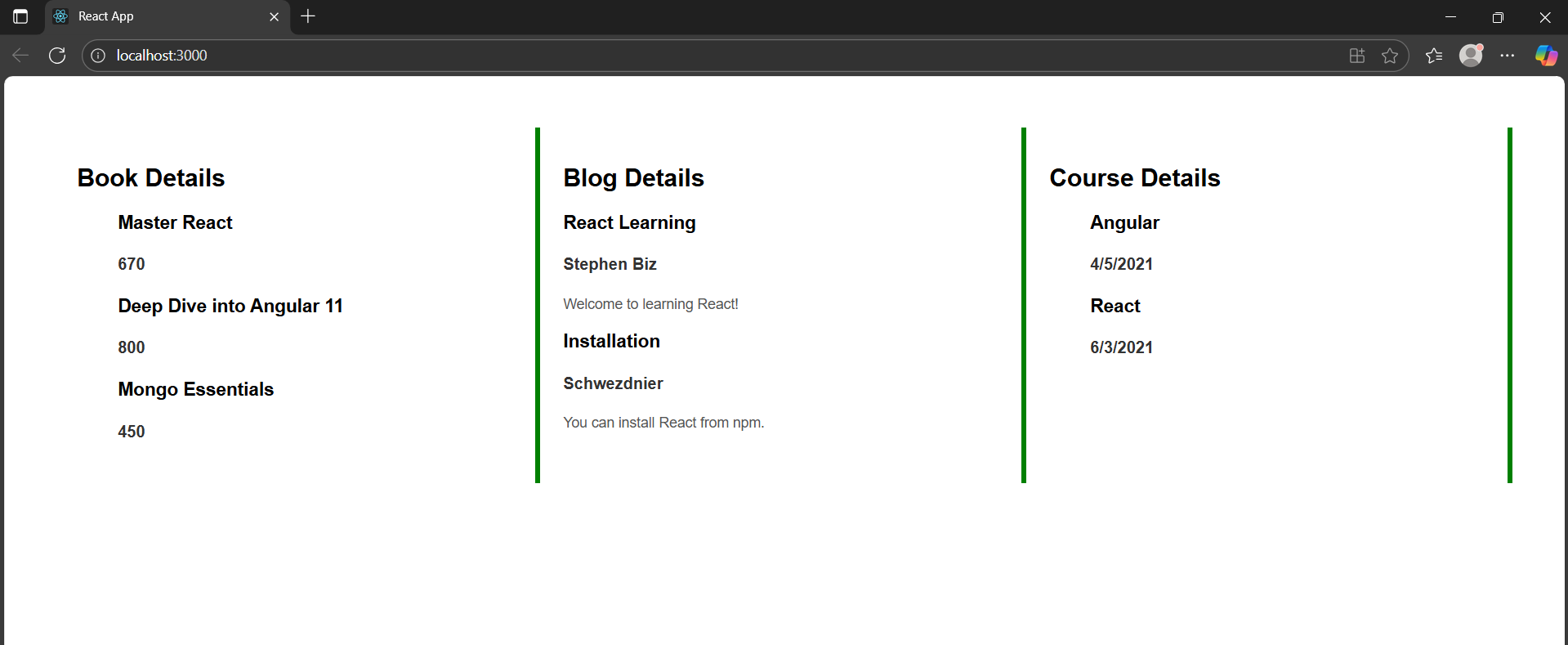
  );

}

export default BookDetails;



**Output:**



**Explanation:**

* The app is split into three separate components: BookDetails, BlogDetails, and CourseDetails. Each handles displaying a specific type of data — books, blogs, and courses — making the code modular and easy to maintain.
* The useState hook is used in the App.js file to control whether books, blogs, or courses are shown on the page. For example, if showBlogs is false, a fallback message is shown instead of the blog list.
* Each component receives data as props (books, blogs, courses) from the App.js file. The components use .map() to loop over the data and display each item dynamically.
* The App.css file contains neatly written class-based CSS styles. Each component section (.st2, .v1, .mystyle1) has its own formatting with padding, borders, and font styling for a clean layout.
* The entire app is wrapped inside a div with the class container styled using Flexbox. This allows the three sections to appear side by side. Also, conditions like if (!show) or ternary operators are used to control what gets displayed.