**13. ReactJS – HOL**

**Objectives**

* **Explain various ways of conditional rendering**

Conditional rendering means show something only if a condition is true.  
If a user is logged in - show “Welcome”  
If not logged in - show “Please login”

Example: 1 Using if else

*function App() {  
 const isLoggedIn = true;  
 if (isLoggedIn) {  
 return <h1>Welcome!</h1>;  
 } else {  
 return <h1>Please Log In</h1>;  
 }}*

Example: 2 Using ternary operator

*function App() {  
 const isMorning = false;  
 return (  
 <h1>{isMorning ? "Good Morning" : "Good Evening"}</h1>  
 );}*

Example: 3 Using && operator

function App() {  
 const showMsg = true;  
 return (  
 <div> <h1>Hello!</h1>  
 {showMsg && <p>This is a secret message.</p>}  
 </div> );  
}

* **Explain how to render multiple components**

It means showing more than one component on the screen at the same time.

Example:  
function App() {  
 return (  
 <div>  
 <Header />  
 <Content />  
 <Footer />  
 </div> );

}

* **Define list component**

A List Component is a React component that displays a list of items.

1. A list of names
2. Products
3. Messages

Use the .map() method to loop through the list and render each item.  
  
Example:  
*function NameList() {  
 const names = ["A", "B", "C"];  
 return (  
 <ul> {names.map((name, index) => (  
 <li key={index}>{name}</li> ))}  
 </ul> );  
}*

* **Explain about keys in React applications**

Keys are special IDs that React uses to track each item in a list. When React updates the UI (re-rendering), it needs to:

* Identify each list item correctly
* Avoid bugs or unnecessary changes
* **Explain how to extract components with keys**  
  When you're rendering a list, it's good to extract each list item into its own component and you still need to pass a unique key.

Example:  
*const users = [  
 { id: 1, name: "A" },*

*{ id: 2, name: "B" },*

*{ id: 3, name: "C" }*

*];  
function UserItem(props) {*

*return <li>{props.name}</li>;*

*}  
function UserList() {*

*return (*

*<ul>*

*{users.map(user => (*

*<UserItem key={user.id} name={user.name} />*

*))}*

*</ul>*

*);*

*}*

* **Explain React Map, map() function**Javascript expressions can be used in JSX

In React, we use the JavaScript .map() function to loop through an array and render elements (like <li>, <div>, or components) for each item.

* + Loops through each item in an array
  + Returns a new array
  + You use it to return JSX for each item

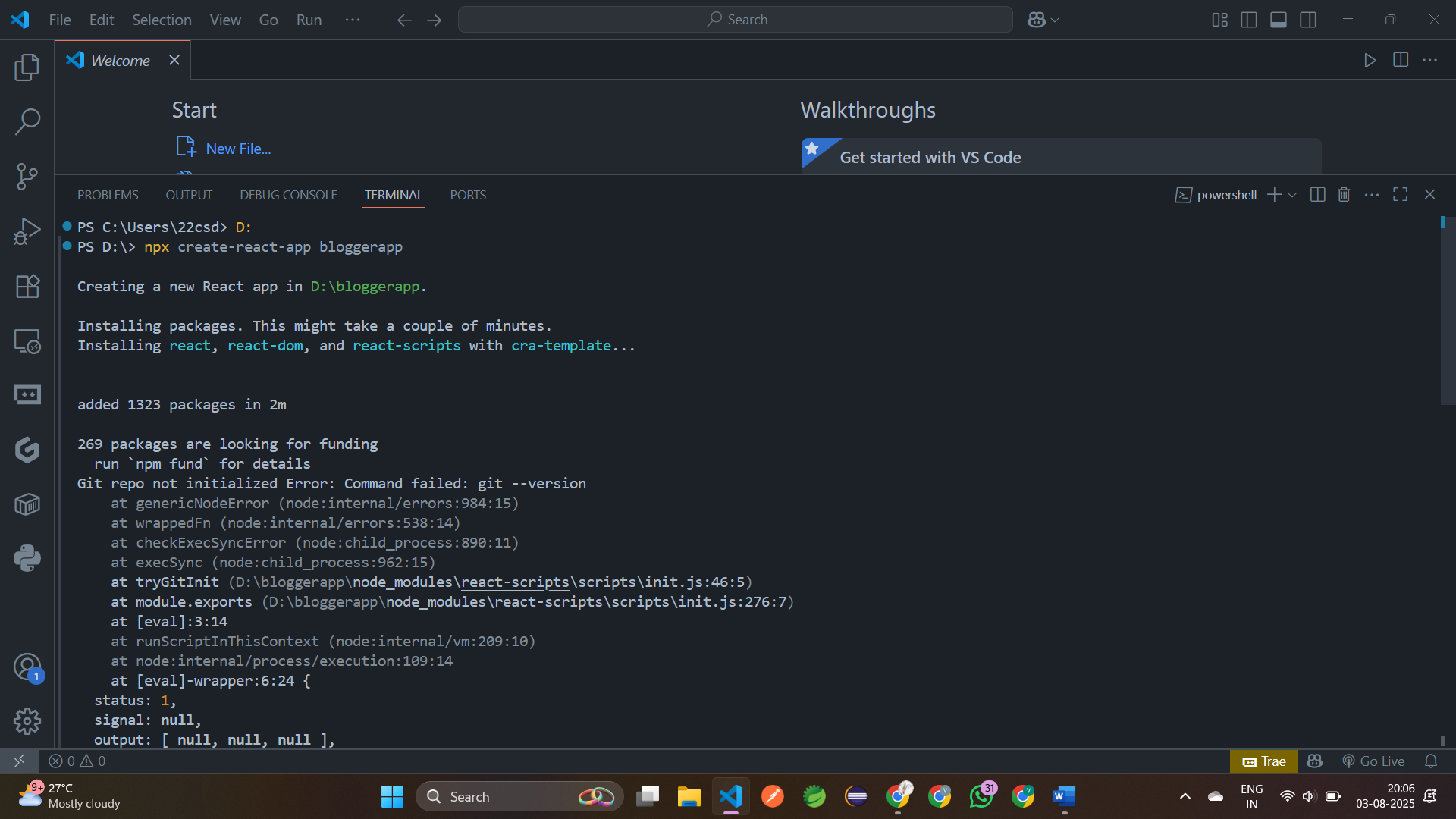
**EXERCISE – 2:** Creation of a Blogger App

Step 1: Open **VS Code**

Step 2: Create a New React App

**npx create-react-app bloggerapp**  
Step 3: Navigate into the project folder

**cd bloggerapp**



Step 4: Create a new file **books.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 },  
 ];

Step 5: Create component **BookDetails.js** inside **components** folder

import React from "react";  
 export default function BookDetails({ books }) {  
   return (  
     <div>  
       <h1 style={{color:'red'}}>Book Details</h1>  
       <ul>        {books.map((book) => (  
           <li key={book.id}>  
             <h3>{book.bname}</h3>  
             <h4>{book.price}</h4>  
           </li>        ))}  
       </ul>  
     </div>  );  
 }

Step 6: Create component **BlogDetails.js** inside **components** folder

import React from "react";  
 export default function BlogDetails({ blogs }) {  
   return (  
     <div>  
       <h1 style={{color:'blue'}}>Blog Details</h1>  
       {blogs.map((blog, idx) => (  
         <div key={idx}>  
           <h2>{blog.title}</h2>  
           <h4>{blog.author}</h4>  
           <p>{blog.desc}</p>  
         </div>  
       ))}  
     </div>  
   );  
 }

Step 6: Create component **CourseDetails.js** inside **components** folder

import React from "react";  
 export default function CourseDetails({ courses }) {  
   return (  
     <div>       <h1 style={{color:'green'}}>Course Details</h1>  
       {courses.map((c, idx) => (         <div key={idx}>  
           <h2>{c.name}</h2>           <small>{c.date}</small>  
         </div>       ))}     </div>   );}

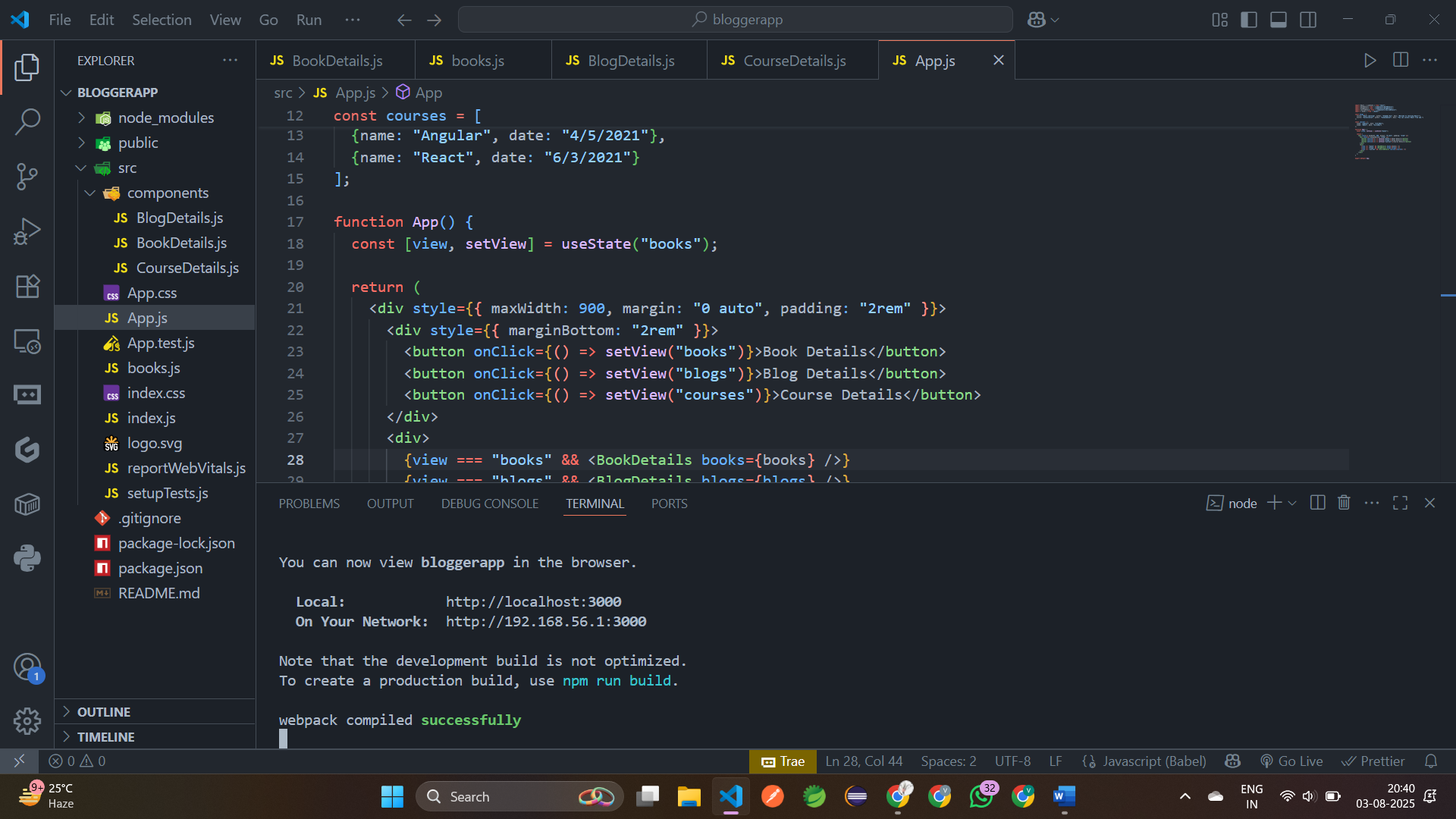
step 7: Open **src/App.js** and modify it

import './App.css';  
import React, { useState } from "react";  
import BookDetails from "./components/BookDetails";  
import BlogDetails from "./components/BlogDetails";  
import CourseDetails from "./components/CourseDetails";  
import { books } from "./books";  
const blogs = [  
  {title: "React Learning", author: "Stephen Biz", desc: "Welcome to learning React!"},  
  {title: "Installation", author: "Schwezdenier", desc: "You can install React from npm."}  
 ];const courses = [  
  {name: "Angular", date: "4/5/2021"},  
  {name: "React", date: "6/3/2021"}  
 ]; function App() {  
   const [view, setView] = useState("books");  return (  
    <div style={{ maxWidth: 900, margin: "0 auto", padding: "2rem" }}>  
      <div style={{ marginBottom: "2rem" }}>  
        <button onClick={() => setView("books")}>Book Details</button>  
        <button onClick={() => setView("blogs")}>Blog Details</button>  
        <button onClick={() => setView("courses")}>Course Details</button>      </div>  
      <div>        {view === "books" && <BookDetails books={books} />}  
        {view === "blogs" && <BlogDetails blogs={blogs} />}  
        {view === "courses" && <CourseDetails courses={courses} />}      </div>  
    </div>  );}

export default App;

step 8: Run the App using **npm start**

Url: <http://localhost:3000>



**Output:**

