1. **ReactJS-HOL**

**1. SPA (Single-Page Application) & Benefits**

* Loads a single HTML page and updates dynamically.
* **Benefits**: Fast, smooth UX, less server load, good for dynamic content.

**2. React & Its Working**

* A JavaScript library for building UI.
* **Works by**: Using components, JSX, virtual DOM, and one-way data flow.

**3. SPA vs MPA**

| **SPA** | **MPA** |
| --- | --- |
| One HTML page | Multiple pages |
| Fast after load | Slower due to reloads |
| Client-side routing | Server-side routing |
| Hard SEO | Better SEO |

**4. Pros & Cons of SPA**

* **Pros**: Fast, smooth, less reload.
* **Cons**: Poor SEO, slow initial load, harder browser navigation.

**5. About React**

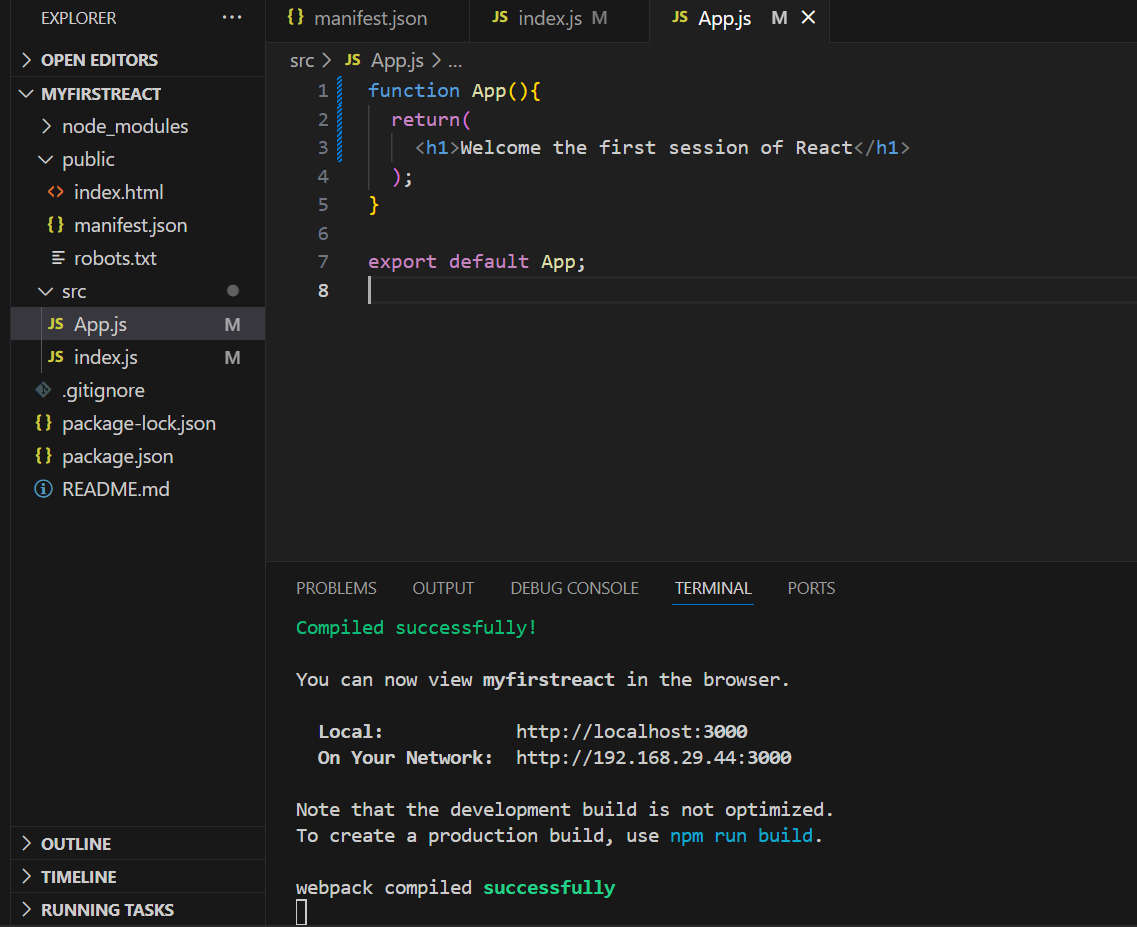
* JavaScript library to build UIs using reusable components.
* Used by Facebook, Netflix, etc.

**6. Virtual DOM**

* A lightweight copy of real DOM.
* React updates only the changed parts → faster performance.

**7. Features of React**

* JSX syntax
* Component-based
* Virtual DOM
* One-way data flow
* Lifecycle methods
* Hooks support



**## Output**

****

**2. ReactJS-HOL**

**1. React Components**

React components are **reusable building blocks** of a React UI. Each component returns JSX (UI elements).

2. **Component vs JavaScript Function**

|  |  |
| --- | --- |
| **Component** | **JavaScript Function** |
|  |  |
| Returns JSX | Returns data/values |
| Follows React rules | No special structure needed |
| Can have state/hooks | No built-in React features |

**3. Types of Components**

* **Class Component**
* **Function Component**

**4. Class Component**

* A class-based component extends React.Component and uses a render() method to return JSX.

**5. Function Component**

* A simple JS function that returns JSX. Can use Hooks for state and lifecycle.

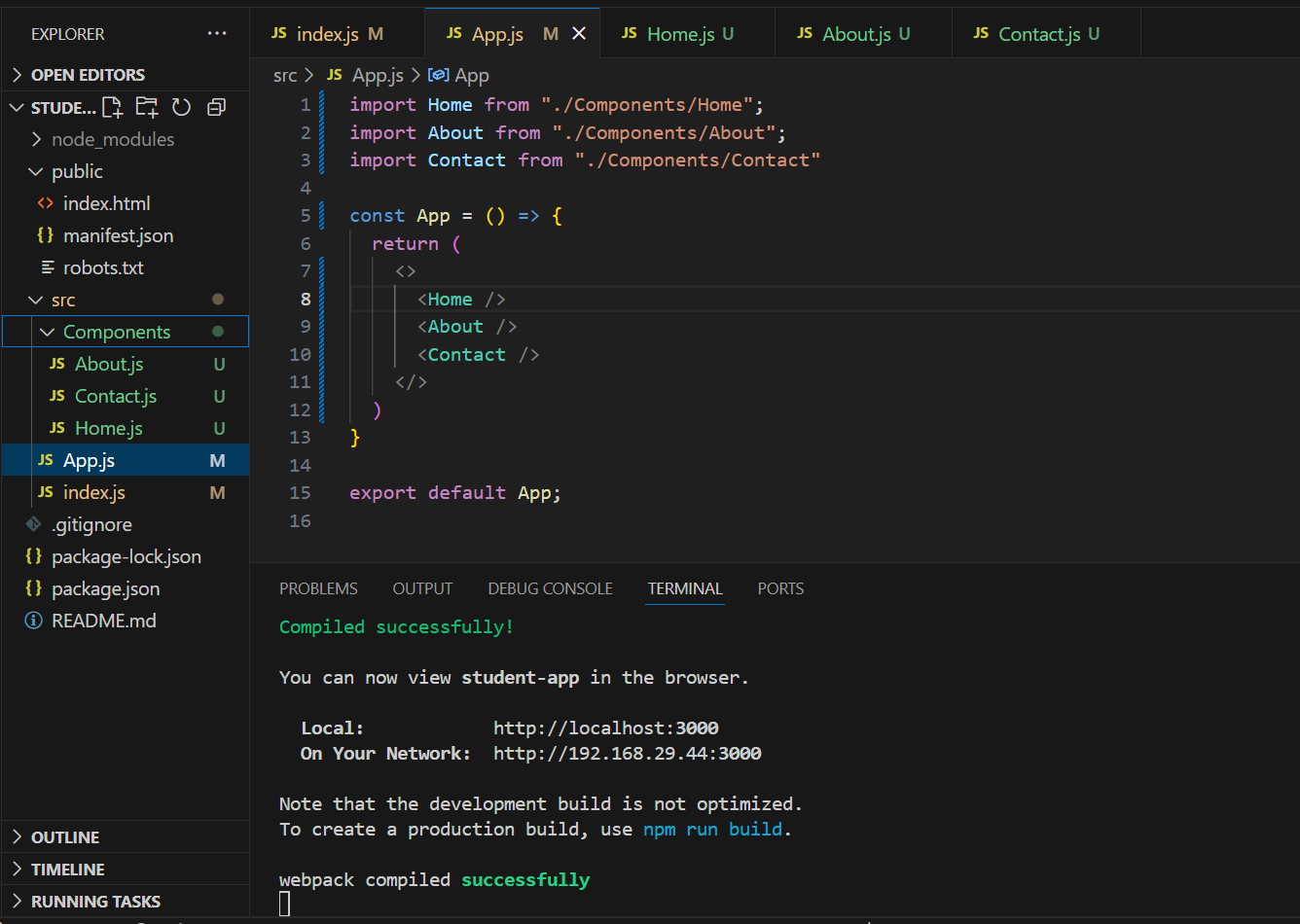
**6. Component Constructor**

* constructor(props) is used in class components to initialize state and bind methods.

**7. render() Function**

* render() is a method in class components that returns the JSX to be displayed on the screen.

**## App.js**

****

**# src/Components/Home.js**

const Home=() =>{

    return(

        <h2>Welcome to the Home page of Student Management Portal</h2>

    )

}

export default Home;

**# src/Components/About.js**

const About=() =>{

    return(

        <h2>Welcome to the About page of Student Management Portal</h2>

    )

}

export default About;

**# src/Components/Contact.js**

const Contact=() =>{

    return(

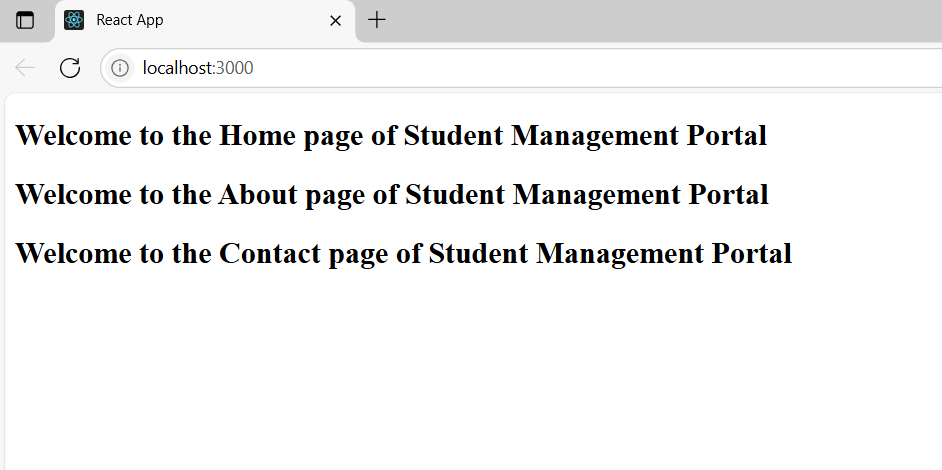
        <h2>Welcome to the Contact page of Student Management Portal</h2>

    )

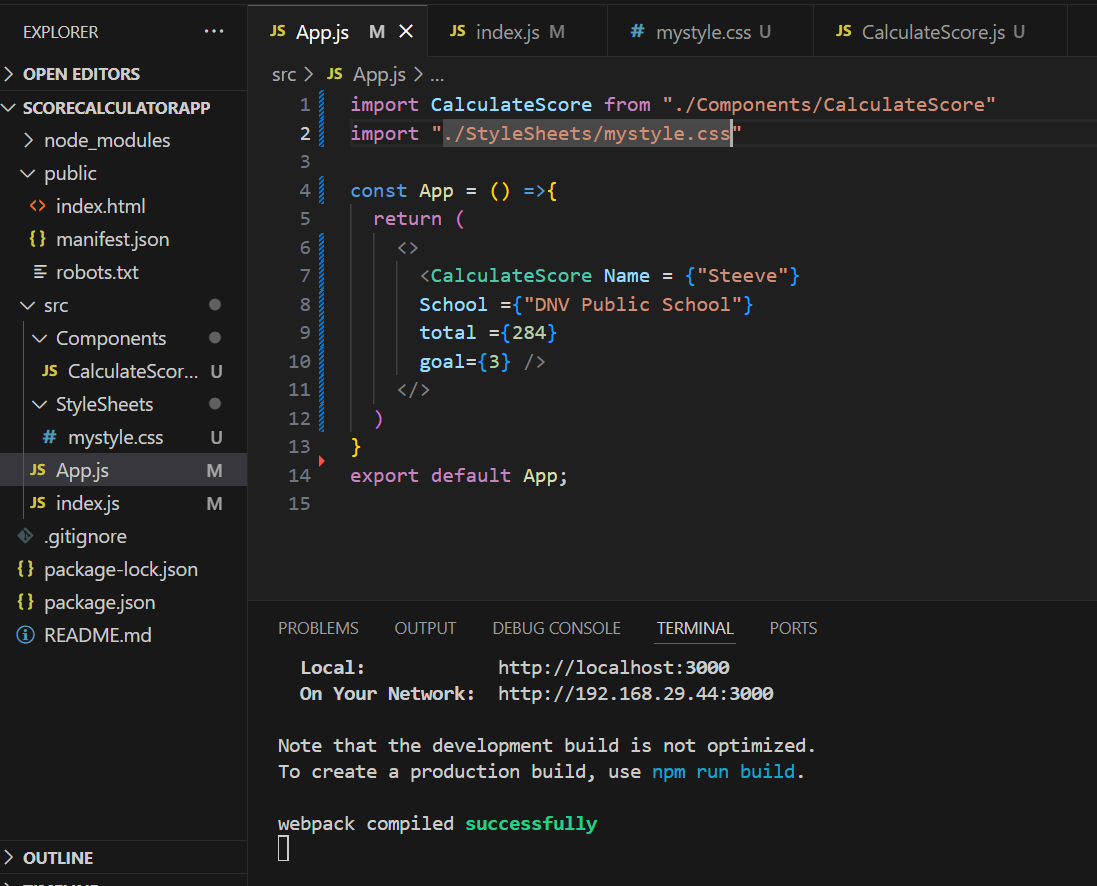
}

export default Contact;

**## Output**

****

**3. ReactJS-HOL**

**# App.js**

**# src/StyleSheets/mystyle.css**

.Name{

    font-weight: 300;

    color: blue;

}

.School{

    color:crimson;

}

.Total{

    color: darkmagenta;

}

.formatstyle{

    text-align: center;

    font-size: large;

}

.Score{

    color: forestgreen;

}

**# src/Components/CalculateScore.js**

const percentToDecimal= (decimal) => {

        return (decimal.toFixed(2) + "%")

    }

    const calcScore= (total,goal) => {

        return percentToDecimal(total/goal)

    }

const CalculateScore = ({Name, School, total, goal}) => {

    return (

        <div className="formalstyle">

            <h1><font color="Brown"> Student Details : </font></h1>

            <div className="Name">

                <b><span> Name : </span></b>

                <span>{Name}</span>

            </div>

            <div className="School">

                <b><span> School : </span></b>

                <span>{School}</span>

            </div>

            <div className="Total">

                <b><span> Total : </span></b>

                <span>{total}</span>

                <span>Marks</span>

            </div>

            <div className="Score">

                <b><span> Score : </span></b>

                <span>{

                calcScore(

                    total,goal

                )}

                </span>

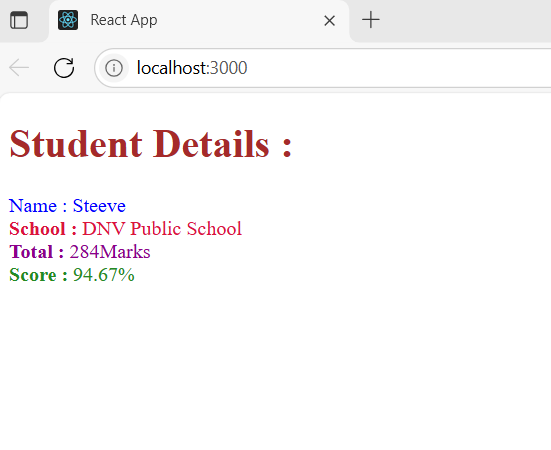
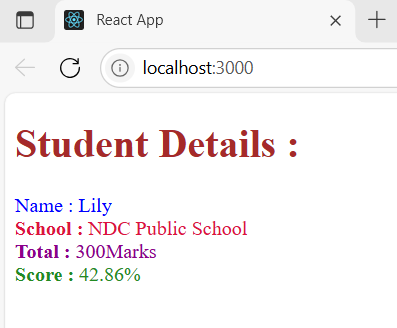
            </div>

        </div>

    )

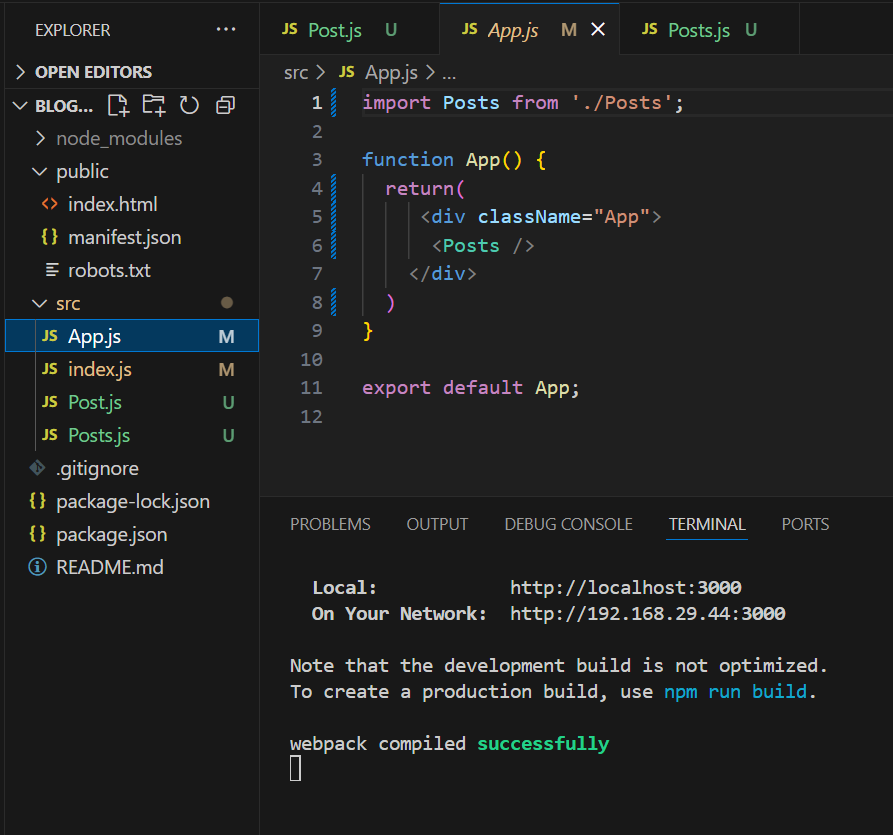
}

export default CalculateScore;

**# Output**

**4. ReactJS-HOL**

**# App.js**

****

**# src/Post.js**

function Post(props) {

  return (

    <div style={{ border: "1px solid #ccc", padding: "10px", marginBottom: "10px" }}>

      <h2>{props.id}</h2>

      <h3>{props.title}</h3>

      <p>{props.body}</p>

    </div>

  );

}

export default Post;

**# src/Posts.js**

import { Component } from 'react';

import Post from './Post';

class Posts extends Component {

  constructor(props) {

    super(props);

    this.state = {

      posts: [],

      hasError: false

    };

  }

  loadPosts = () => {

    fetch('https://jsonplaceholder.typicode.com/posts')

      .then(res => res.json())

      .then(data => this.setState({ posts: data }))

      .catch(err => {

        this.setState({ hasError: true });

        console.error("Fetch error:", err);

      });

  }

  componentDidMount() {

    this.loadPosts();

  }

  render() {

    if (this.state.hasError) {

      return <h2>Something went wrong!</h2>;

    }

    return (

      <div>

        <h2 style={{color:'darkblue'}}>Blog Posts</h2>

        {this.state.posts.map(post => (

          <Post id={post.id} title={post.title} body={post.body} />

        ))}

      </div>

    );

  }

  componentDidCatch(error, info) {

    alert("An error occurred while rendering the posts.");

    console.error("Error:", error, info);

  }

}

export default Posts;

**# src/index.js**

import React from 'react';

import ReactDOM from 'react-dom/client';

import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(

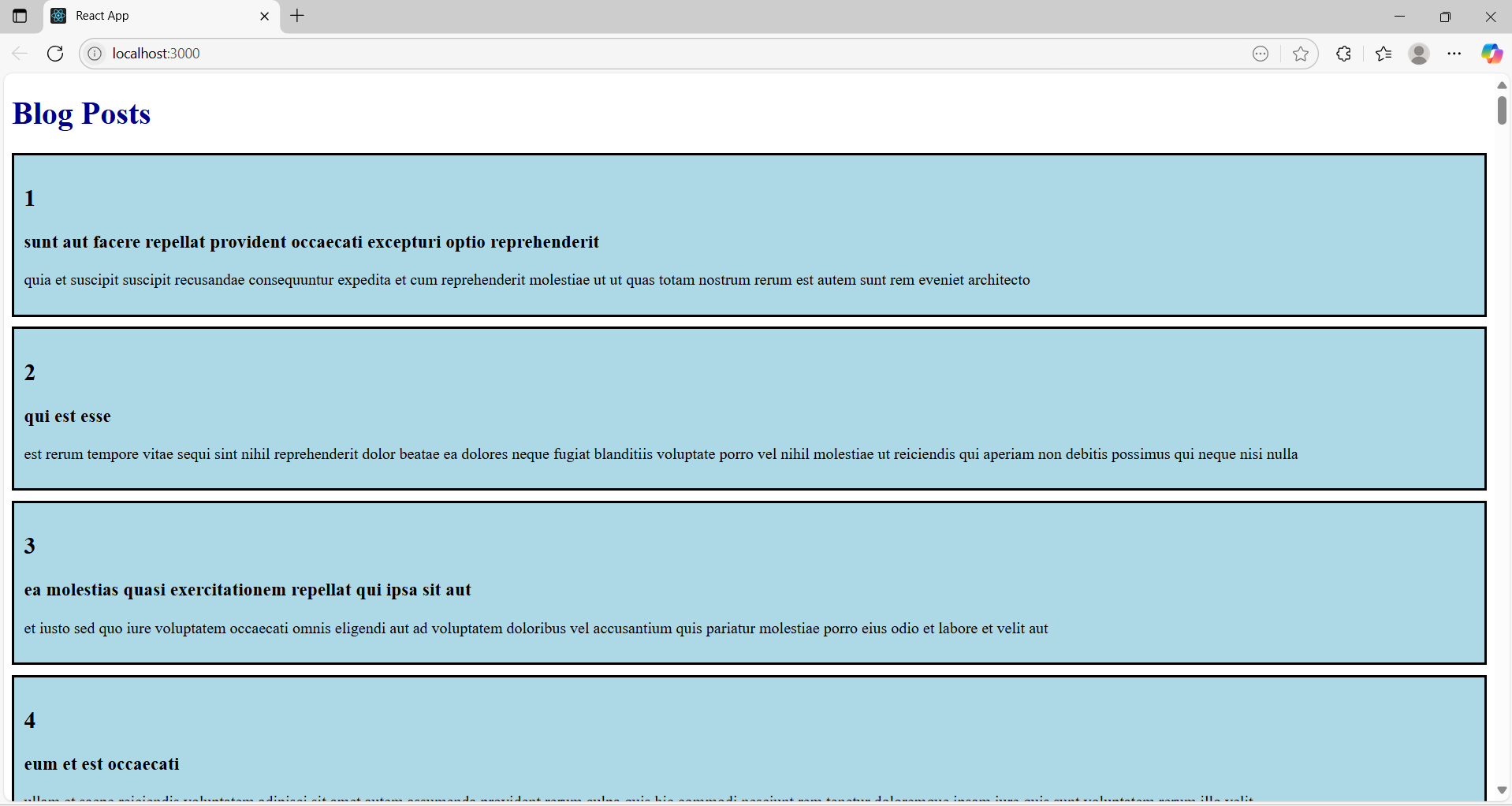
  <React.StrictMode>

    <App />

  </React.StrictMode>

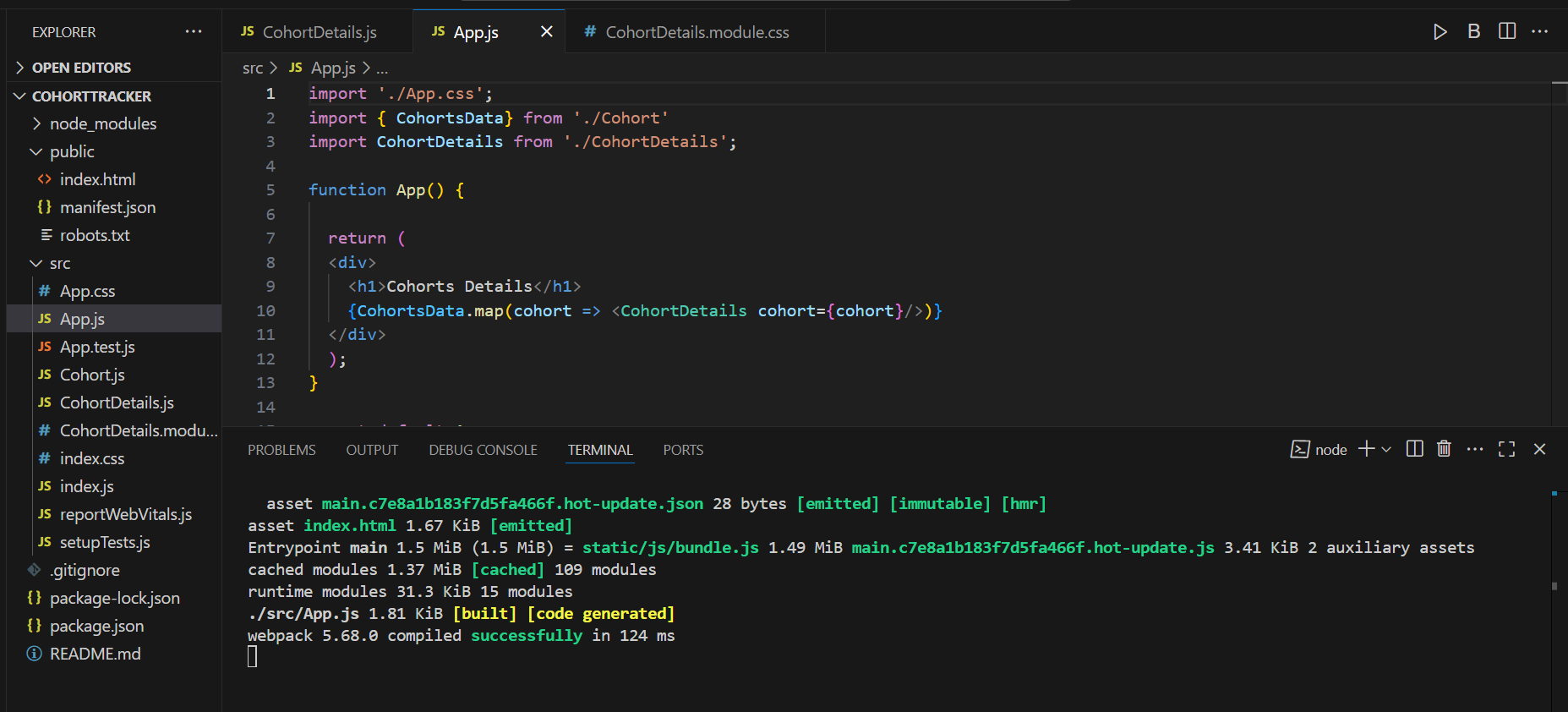
);

**# Output**



****

**5. ReactJS-HOL**

**# App.js**

**# src/ CohortDetails.js**

import styles from './CohortDetails.module.css';

function CohortDetails(props) {

    return (

        <div className={styles.box}>

            <h3 style={{ color: props.status === 'ongoing' ? 'green' : 'blue' }}>

                {props.cohort.cohortCode} -

                <span>{props.cohort.technology}</span>

            </h3>

            <dl>

                <dt>Started On</dt>

                <dd>{props.cohort.startDate}</dd>

                <dt>Current Status</dt>

                <dd>{props.cohort.currentStatus}</dd>

                <dt>Coach</dt>

                <dd>{props.cohort.coachName}</dd>

                <dt>Trainer</dt>

                <dd>{props.cohort.trainerName}</dd>

</dl>

        </div>

    );

}

export default CohortDetails;

**# src/ CohortDetails.module.css**

.box {

  width: 300px;

  display: inline-block;

  margin: 10px;

  padding: 10px 20px;

  border: 1px solid black;

  border-radius: 10px;

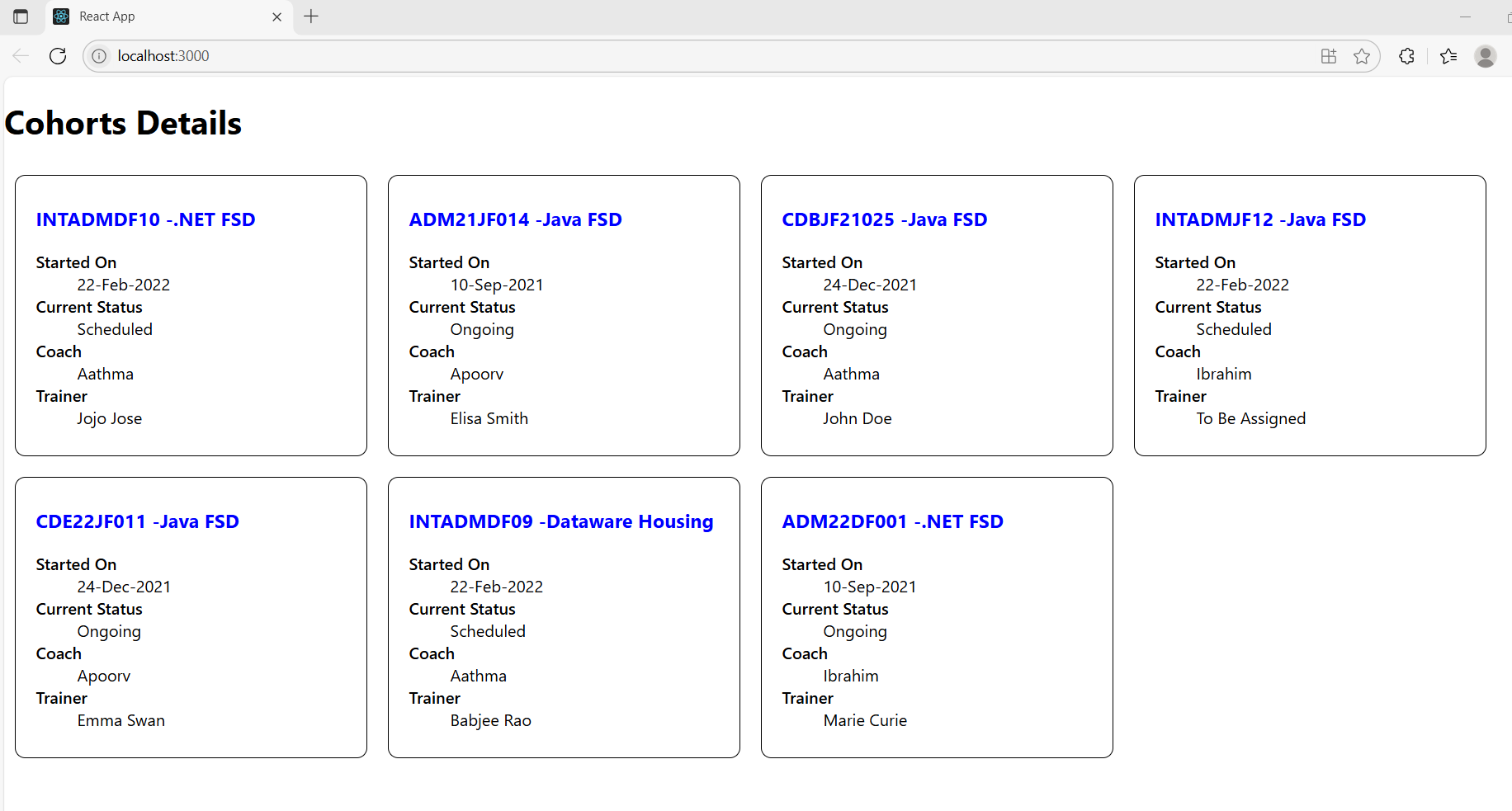
}

dt {

  font-weight: 500;

}

**# Output**

****