**OBJECTIVES & CONCEPTS**

**1. Define SPA and Its Benefits**

**SPA (Single Page Application)**:  
A web application that loads a single HTML page and dynamically updates the content without refreshing the whole page.

**Benefits of SPA:**

* Faster user experience (less server round-trips)
* Smooth navigation
* Backend only sends data, not entire pages
* Better caching and offline support

**2. Define React and Identify Its Working**

**React**:  
A JavaScript library created by Facebook for building **user interfaces**—especially SPAs.

**How it works:**

* Uses **components** (reusable UI blocks)
* Uses a **virtual DOM** to detect changes and update the real DOM efficiently

**3. Difference Between SPA and MPA**

| **Aspect** | **SPA** | **MPA (Multi Page Application)** |
| --- | --- | --- |
| Page reload | No (updates dynamically) | Yes (each interaction reloads page) |
| Speed | Faster, after initial load | Slower due to full reloads |
| Technologies | React, Angular, Vue | JSP, PHP, traditional server-rendered |
| URL management | Done using client-side routing | Server-based routing |
| SEO | More difficult | Easier (static content) |

**4. Pros & Cons of Single-Page Application**

**Pros:**

* Fast and responsive
* Smooth user experience
* Less server load

**Cons:**

* Poor SEO (unless using SSR)
* Initial load may be slower
* Can be complex to manage state & routing

**5. Explain About React**

* Component-based UI library
* Developed and maintained by **Meta (Facebook)**
* Follows **declarative programming**
* Uses **JSX** (JavaScript + XML)

**6. Define Virtual DOM**

The **Virtual DOM** is a lightweight in-memory copy of the actual DOM.  
React compares (diffs) the new virtual DOM with the previous one, and **efficiently updates** only the parts of the actual DOM that changed.

**7. Features of React**

* Declarative UI
* Component-based architecture
* JSX Syntax
* Virtual DOM
* Unidirectional data flow
* React Hooks
* Strong community and ecosystem

**1. Hands-On Lab: React Setup**

**Prerequisites:**

* Install [Node.js and npm](https://nodejs.org/en/download)
* Install **Visual Studio Code**

**STEP-BY-STEP SETUP**

**1. Install Create-React-App Tool**

npm install -g create-react-app

**2. Create React App**

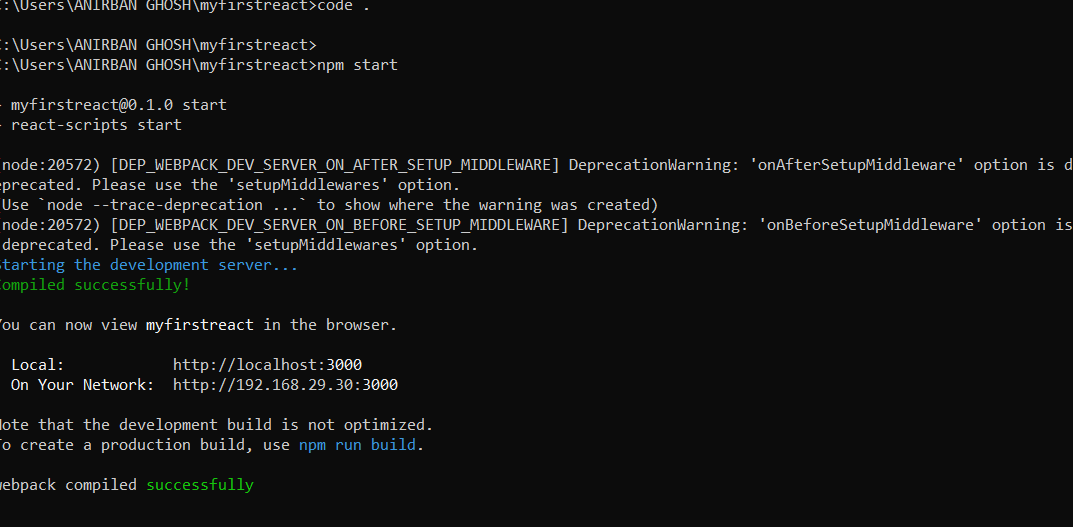
npx create-react-app myfirstreact

**3. Navigate to the App Directory**

cd myfirstreact

**4. Open in Visual Studio Code**

code .



**5. Edit App.js**

* Go to src/App.js
* Delete the existing code
* Paste the following:

import React from 'react';

function App() {

return (

<div>

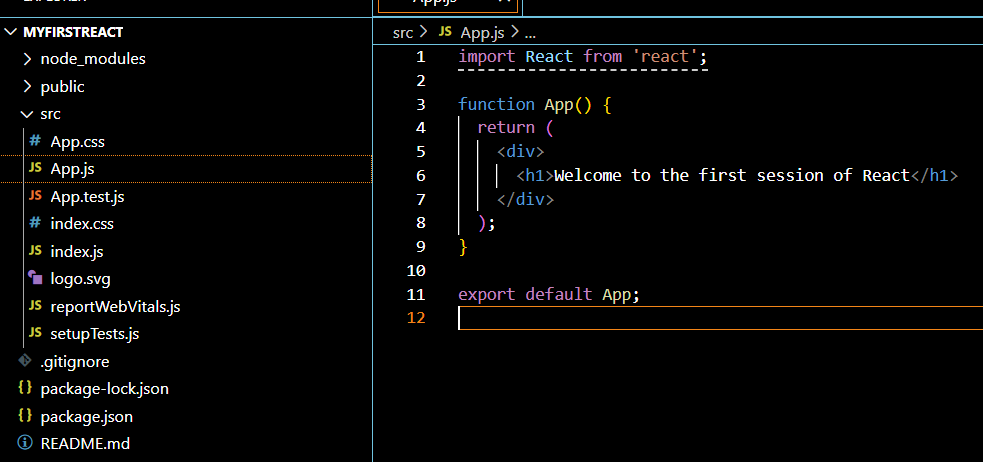
<h1>Welcome to the first session of React</h1>

</div>

);

}

export default App;

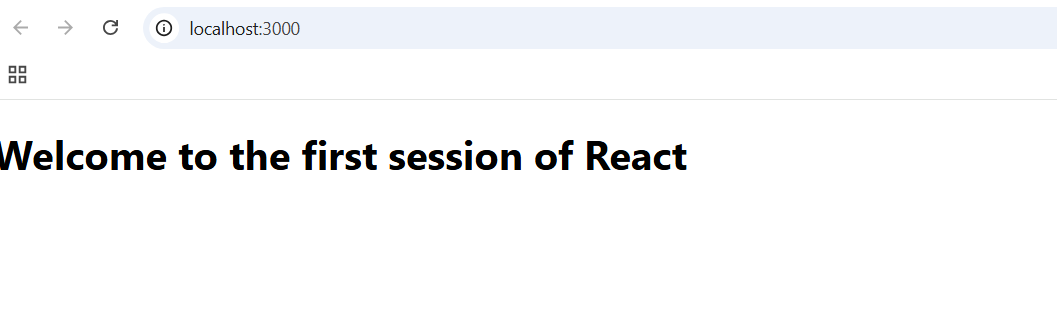


**6. Run the App**

npm start

**7. Open in Browser**

Go to:  
🔗 http://localhost:3000



**2.Hands-On Lab: Build “StudentApp” with Class Components**

**Create a React App**

Open terminal in **VS Code** and run:

npx create-react-app StudentApp

cd StudentApp

**Create "Components" Folder**

Inside the src/ folder, create a new folder:

src/

└── Components/

└── Home.js

└── About.js

└── Contact.js



**Home Component (Class Component)**

src/Components/Home.js

import React, { Component } from 'react';

class Home extends Component {

constructor(props) {

super(props); // calling parent constructor

}

render() {

return (

<div>

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

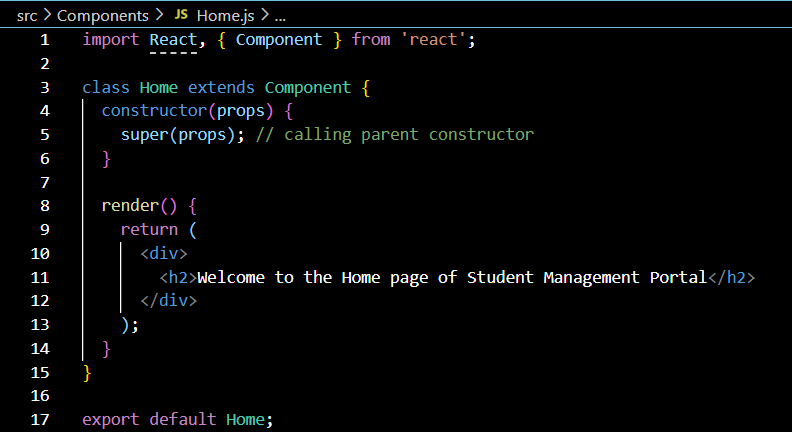
</div>

);

}

}

export default Home;



**About Component (Class Component)**

src/Components/About.js

import React, { Component } from 'react';

class About extends Component {

constructor(props) {

super(props);

}

render() {

return (

<div>

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

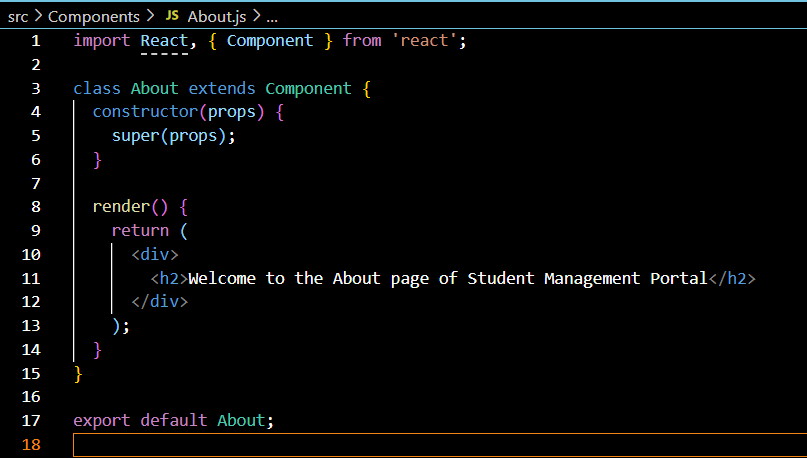
</div>

);

}

}

export default About;



**Contact Component (Class Component)**

src/Components/Contact.js

import React, { Component } from 'react';

class Contact extends Component {

constructor(props) {

super(props);

}

render() {

return (

<div>

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

</div>

);

}

}

export default Contact;



**Edit App.js to Call All Components**

src/App.js

import React from 'react';

import './App.css';

import Home from './Components/Home';

import About from './Components/About';

import Contact from './Components/Contact';

function App() {

return (

<div className="App">

<h1>Student Management Portal</h1>

<Home />

<About />

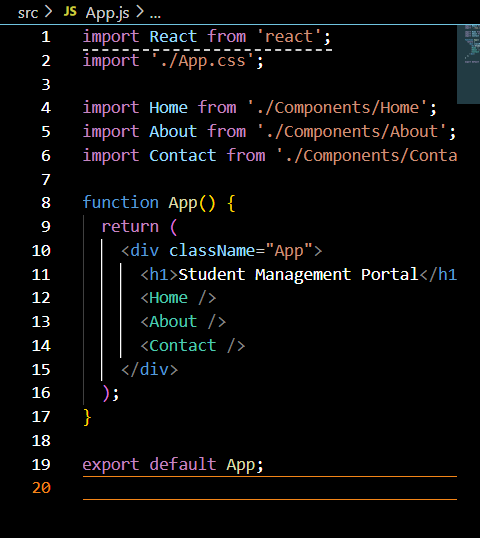
<Contact />

</div>

);

}

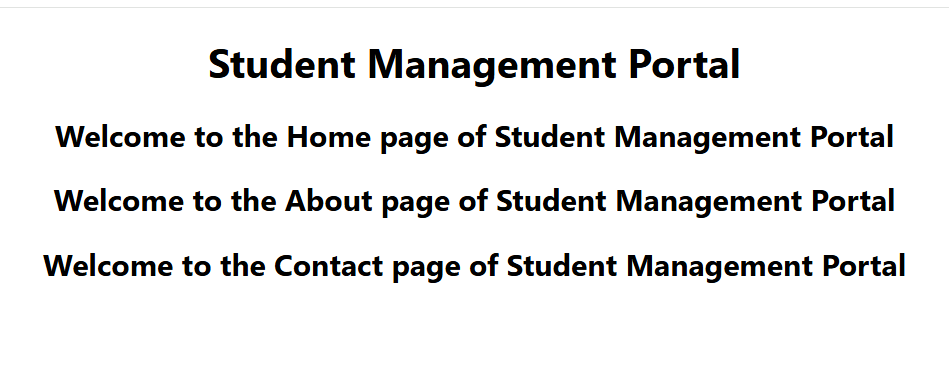
export default App;



**Run the React App**

In the terminal:

npm start



**3.Hands-On Lab: Build “ScoreCalculatorApp” with Functional Component**

**1. Create a React App**

npx create-react-app scorecalculatorapp

cd scorecalculatorapp

**2. Create "Components" Folder**

Inside the src/ folder, create a new folder:

src/

└── Components/

└── CalculateScore.js

**3. Create the CalculateScore Functional Component**

**src/Components/CalculateScore.js**

import React from 'react';

import '../Stylesheets/mystyle.css';

function CalculateScore({ name, school, total, goal }) {

const average = total / goal;

return (

<div className="score-card">

<h2>Score Calculator</h2>

<p><strong>Name:</strong> {name}</p>

<p><strong>School:</strong> {school}</p>

<p><strong>Total Marks:</strong> {total}</p>

<p><strong>Number of Subjects:</strong> {goal}</p>

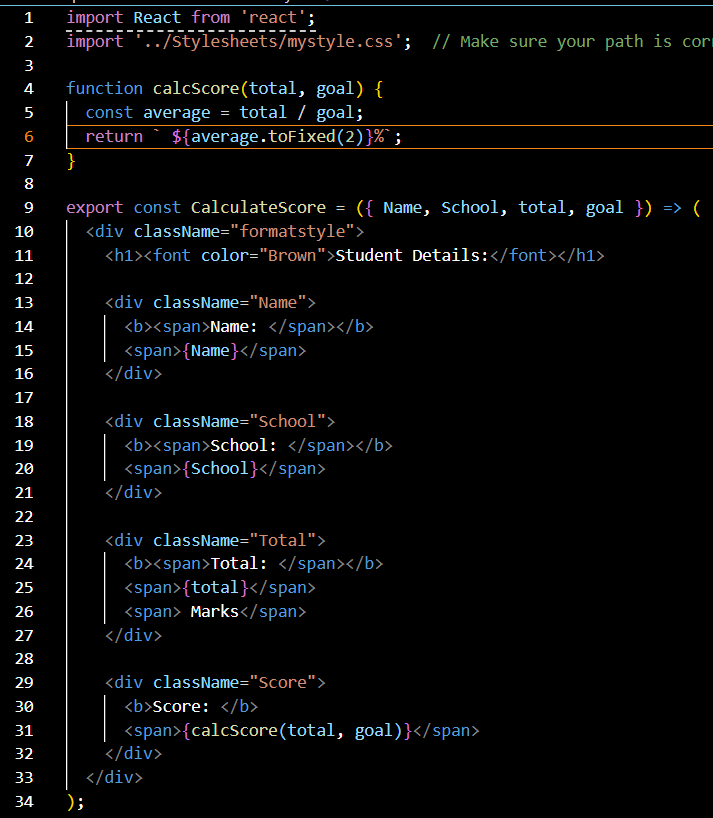
<p><strong>Average Score:</strong> {average.toFixed(2)}</p>

</div>

);

}

export default CalculateScore;



**4. Add Styling**

Create a folder Stylesheets/ and a file inside it named mystyle.css:

📄 **src/Stylesheets/mystyle.css**

.score-card {

background-color: #f2f2f2;

padding: 20px;

margin: 40px auto;

width: 50%;

border-radius: 10px;

box-shadow: 0 4px 8px rgba(0,0,0,0.1);

font-family: Arial, sans-serif;

}

.score-card h2 {

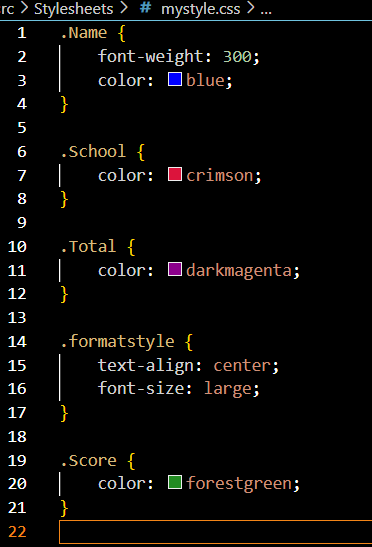
color: #333;

}

.score-card p {

font-size: 16px;

}



**5. Edit App.js to Use Component**

**src/App.js**

import React from 'react';

import './App.css';

import CalculateScore from './Components/CalculateScore';

function App() {

return (

<div className="App">

<h1>Student Score Calculator</h1>

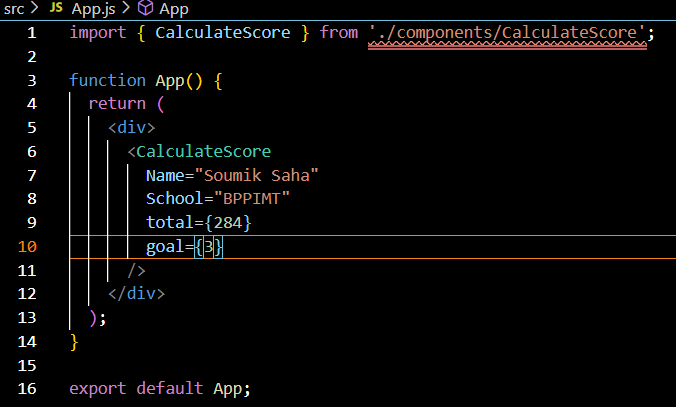
<CalculateScore name="Soumik Saha" school="ABC High School" total={450} goal={5} />

</div>

);

}

export default App;

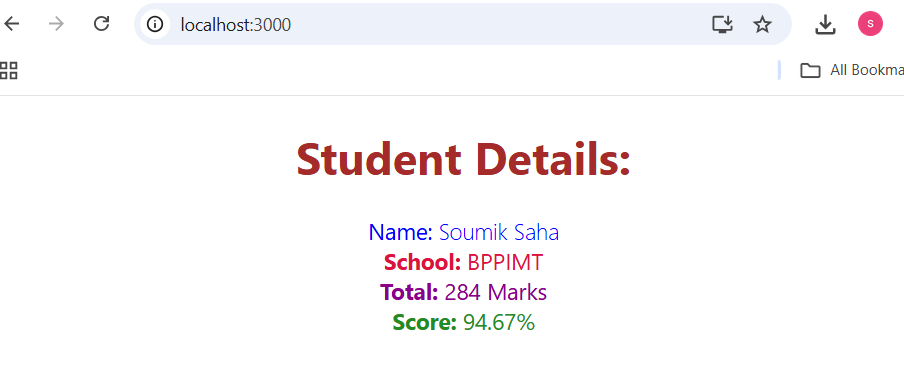


**6. Run the React App**

npm start

**7. View in Browser**

Open browser and type:  
<http://localhost:3000>



**4.Hands-On Lab: Understanding Component Lifecycle in React (Class-Based)**

**Step 1: Create React App**

npx create-react-app blogapp

cd blogapp

code .

**Step 2: Create Post.js File**

Inside src/, create a file called Post.js and add:

import React from 'react';

class Post extends React.Component {

render() {

return (

<div>

<h3>{this.props.title}</h3>

<p>{this.props.body}</p>

<hr />

</div>

);

}

}

export default Post;

**Step 3: Create Posts.js Component**

Create a new file Posts.js inside src/ folder and define a **class component**:

import React, { Component } from 'react';

import Post from './Post';

class Posts extends Component {

constructor(props) {

super(props);

this.state = {

posts: [],

error: null,

};

}

**Step 4: Create loadPosts() Method**

loadPosts = () => {

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

.then(response => {

if (!response.ok) {

throw new Error("Network response was not ok");

}

return response.json();

})

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

.catch(error => this.setState({ error }));

};

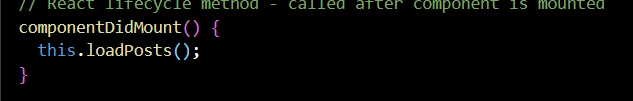


**Step 5: Use componentDidMount()**

componentDidMount() {

this.loadPosts();

}

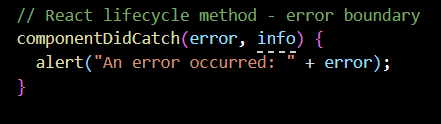


**Step 6: Implement componentDidCatch()**

componentDidCatch(error, info) {

alert("An error occurred: " + error.message);

}



**Step 7: Implement render() Method**

render() {

const { posts, error } = this.state;

if (error) {

return <p>Error loading posts!</p>;

}

return (

<div>

<h2>Blog Posts</h2>

{posts.slice(0, 10).map(post => (

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

))}

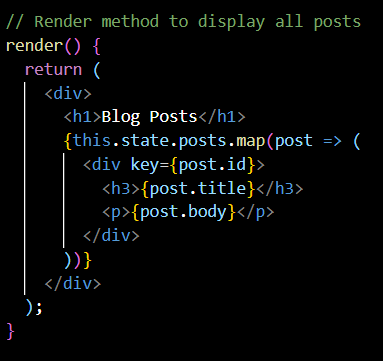
</div>

);

}

}

export default Posts;



**Step 8: Update App.js**

Edit src/App.js to include Posts:

import React from 'react';

import './App.css';

import Posts from './Posts';

function App() {

return (

<div className="App">

<h1>Welcome to BlogApp</h1>

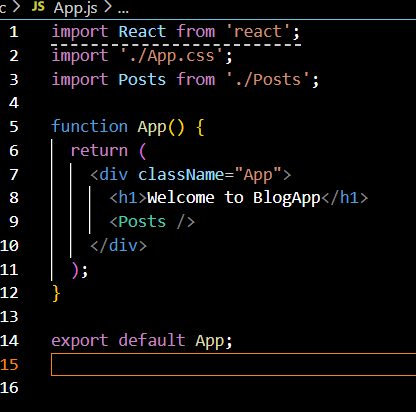
<Posts />

</div>

);

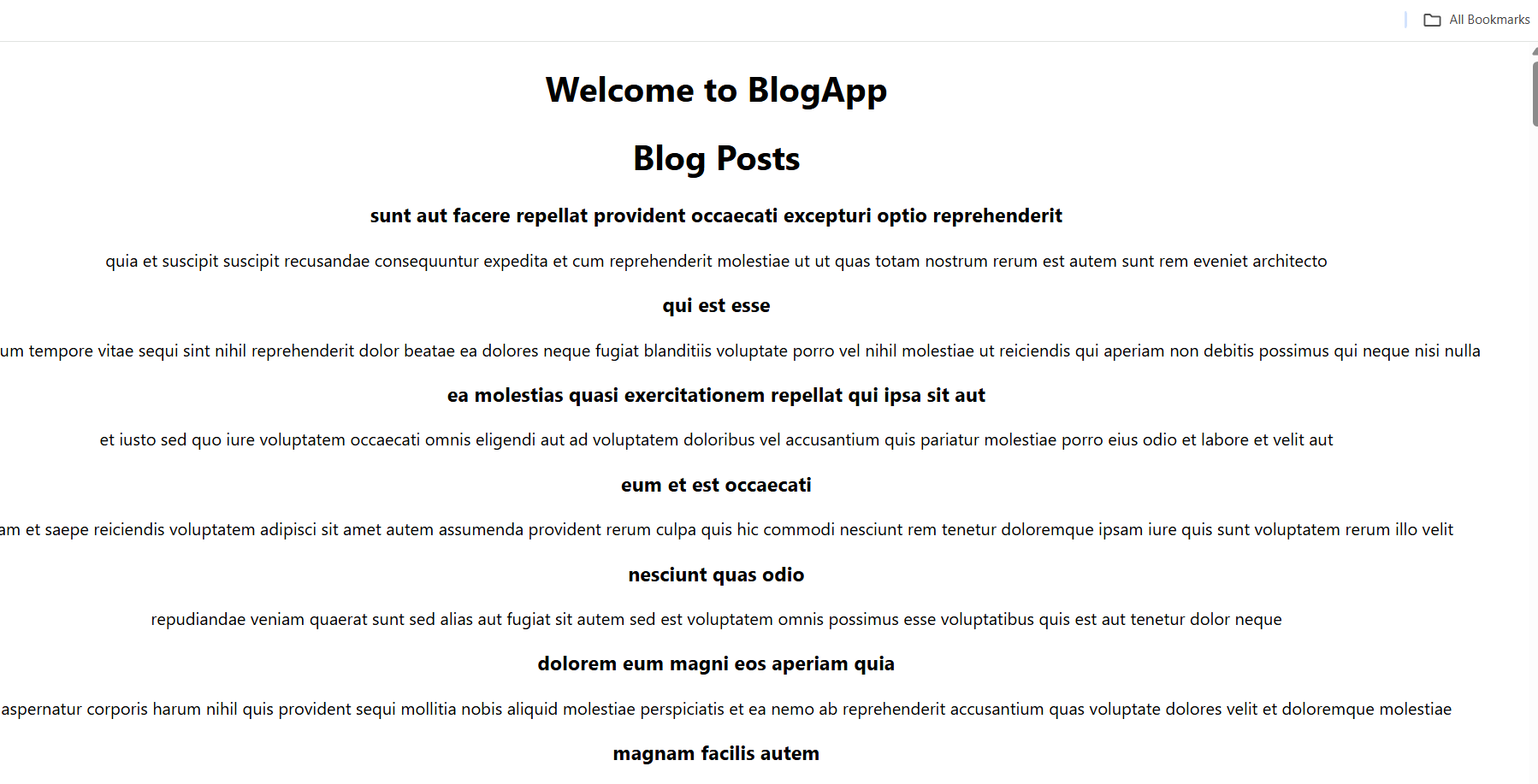
}

export default App;



**Step 9: Run the App**

npm start



**5.Hands-On Lab**

**Setup the React App**

1. **Unzip** the given React app zip file.
2. Open **Command Prompt/Terminal** and navigate to the app folder.
3. Run:

npm install

this restores all node packages.

1. Open the folder in **VS Code**:

code .

**Step 1: Create CohortDetails.module.css**

css

CopyEdit

.container {

padding: 20px;

}

.title {

font-size: 28px;

font-weight: bold;

margin-bottom: 20px;

}

.cardContainer {

display: flex;

gap: 20px;

flex-wrap: wrap;

}

.card {

width: 300px;

border: 1px solid #ccc;

padding: 15px;

border-radius: 10px;

box-shadow: 0 0 4px rgba(0, 0, 0, 0.1);

}

dt {

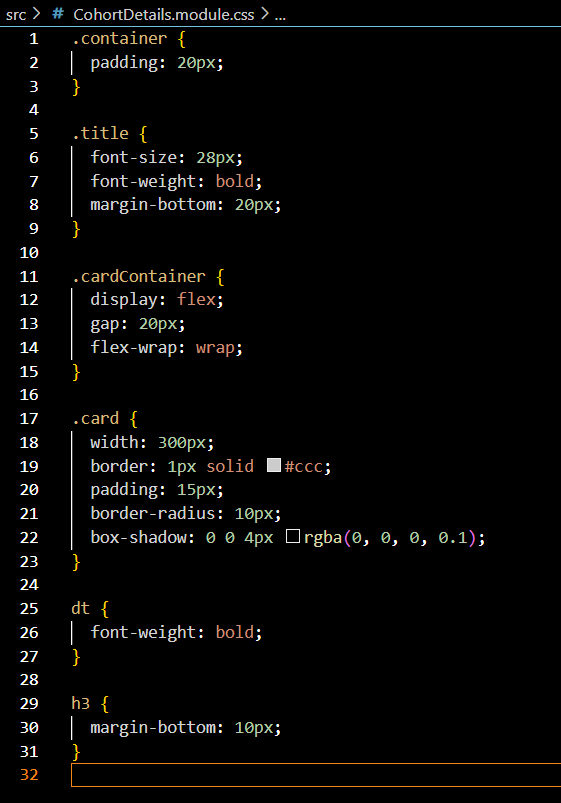
font-weight: bold;

}

h3 {

margin-bottom: 10px;

}



**Step 2: Create CohortDetails.js**

import React from 'react';

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

const CohortDetails = ({ cohort }) => {

const titleStyle = {

color:

cohort.status === 'Scheduled' ? 'blue' :

cohort.status === 'Ongoing' ? 'green' : 'black'

};

return (

<div className={styles.card}>

<h3 style={titleStyle}>{cohort.name}</h3>

<dl>

<dt>Started On</dt>

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

<dt>Current Status</dt>

<dd>{cohort.status}</dd>

<dt>Coach</dt>

<dd>{cohort.coach}</dd>

<dt>Trainer</dt>

<dd>{cohort.trainer}</dd>

</dl>

</div>

);

};

export default CohortDetails;



**Step 3: Update App.js**

import React from 'react';

import CohortDetails from './CohortDetails';

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

const cohorts = [

{

name: 'INTADMDF10 - .NET FSD',

startDate: '22-Feb-2022',

status: 'Scheduled',

coach: 'Soumik',

trainer: 'Jojo Jose'

},

{

name: 'ADM21JF014 - Java FSD',

startDate: '10-Sep-2021',

status: 'Ongoing',

coach: 'Apoorp',

trainer: 'Elisa Smith'

},

{

name: 'CDBJF21025 - Java FSD',

startDate: '24-Dec-2021',

status: 'Ongoing',

coach: 'Soumik',

trainer: 'John Doe'

}

];

function App() {

return (

<div className={styles.container}>

<div className={styles.title}>Cohorts Details</div>

<div className={styles.cardContainer}>

{cohorts.map((cohort, index) => (

<CohortDetails key={index} cohort={cohort} />

))}

</div>

</div>

);}

export default App;

