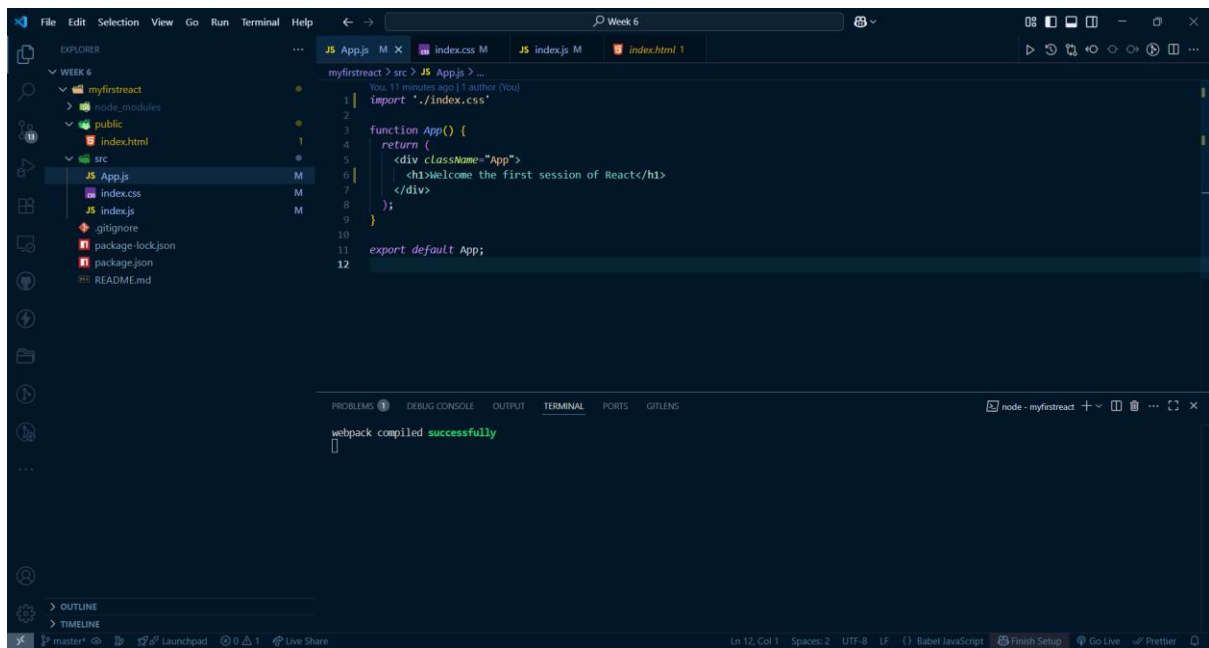


Week - 6

1. ReactJS-HOL

Create a new React Application with the name “myfirstreact”, Run the application to print “welcome to the first session of React” as heading of that page

App.js

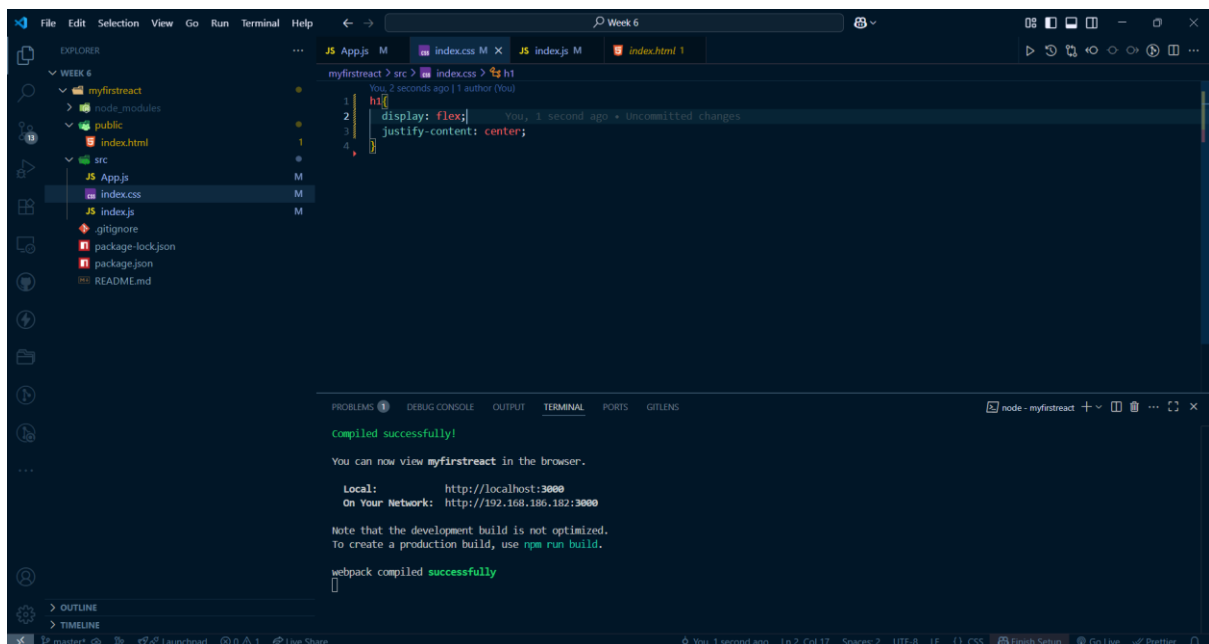


The screenshot shows the Visual Studio Code editor with the 'myfirstreact' project open. The Explorer sidebar on the left shows the file structure: 'myfirstreact' (root) contains 'node_modules', 'public' (with 'index.html'), and 'src' (with 'App.js', 'index.css', and 'index.js'). The main editor window displays the 'App.js' file with the following code:

```
1 | import './index.css'
2 |
3 | function App() {
4 |   return (
5 |     <div className="App">
6 |       <h1>Welcome the first session of React</h1>
7 |     </div>
8 |   );
9 | }
10 |
11 | export default App;
12 |
```

The bottom panel shows the 'TERMINAL' tab with the output: 'webpack compiled successfully'.

Index.css

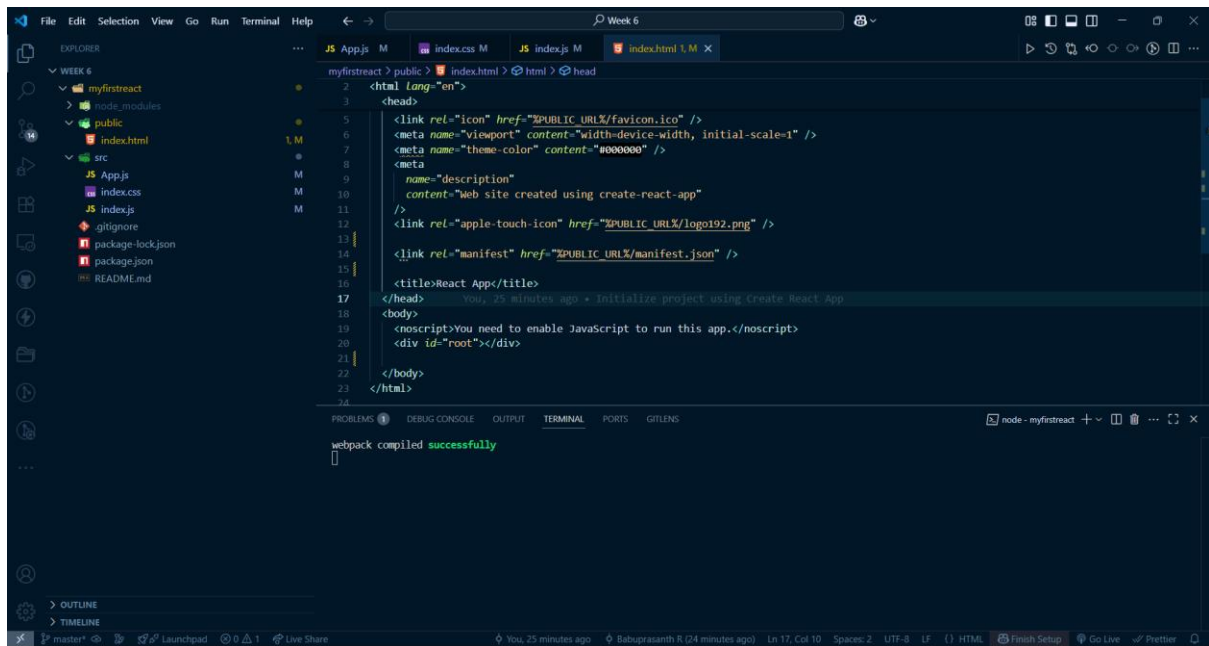


The screenshot shows the Visual Studio Code editor with the 'myfirstreact' project open. The Explorer sidebar on the left shows the file structure. The main editor window displays the 'index.css' file with the following code:

```
1 | h1 {
2 |   display: flex;
3 |   justify-content: center;
4 | }
```

The bottom panel shows the 'TERMINAL' tab with the output: 'Compiled successfully! You can now view myfirstreact in the browser. Local: http://localhost:3000 On Your Network: http://192.168.186.182:3000 Note that the development build is not optimized. To create a production build, use npm run build. webpack compiled successfully'.

Index.html



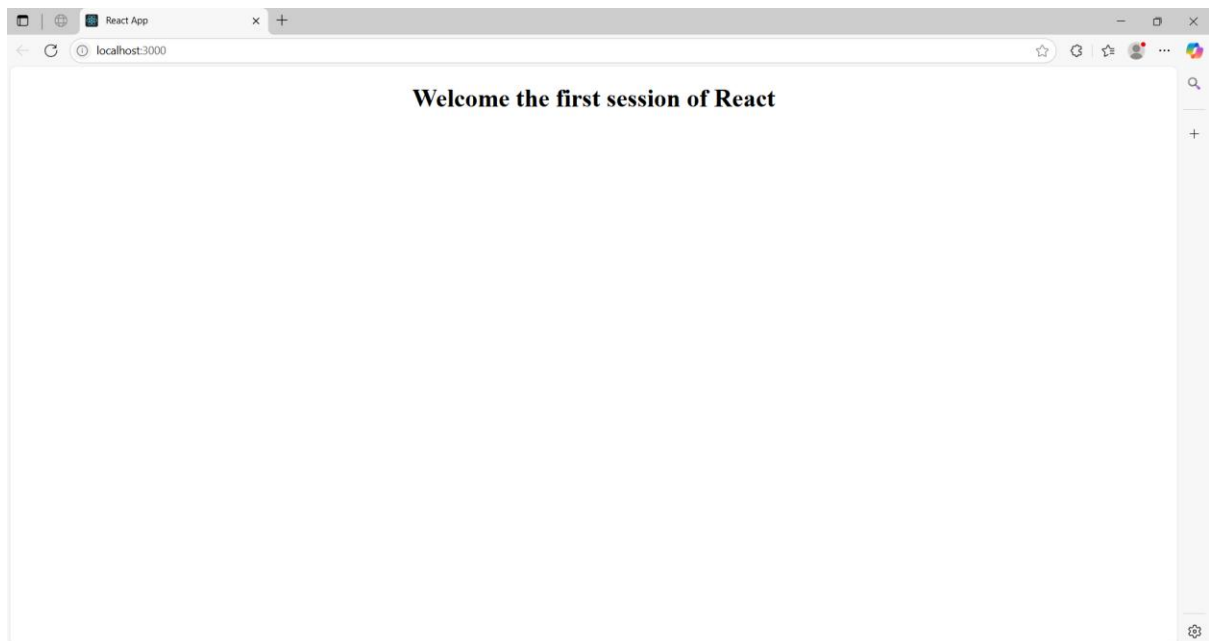
The screenshot shows the Visual Studio Code editor with the 'index.html' file open. The file content is as follows:

```
<html Lang="en">
<head>
  <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <meta name="theme-color" content="#000000" />
  <meta
    name="description"
    content="Web site created using create-react-app"
  />
  <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
  <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />

  <title>React App</title>
</head>
<body>
  <noscript>You need to enable JavaScript to run this app.</noscript>
  <div id="root"></div>
</body>
</html>
```

The terminal at the bottom shows the message: 'webpack compiled successfully'.

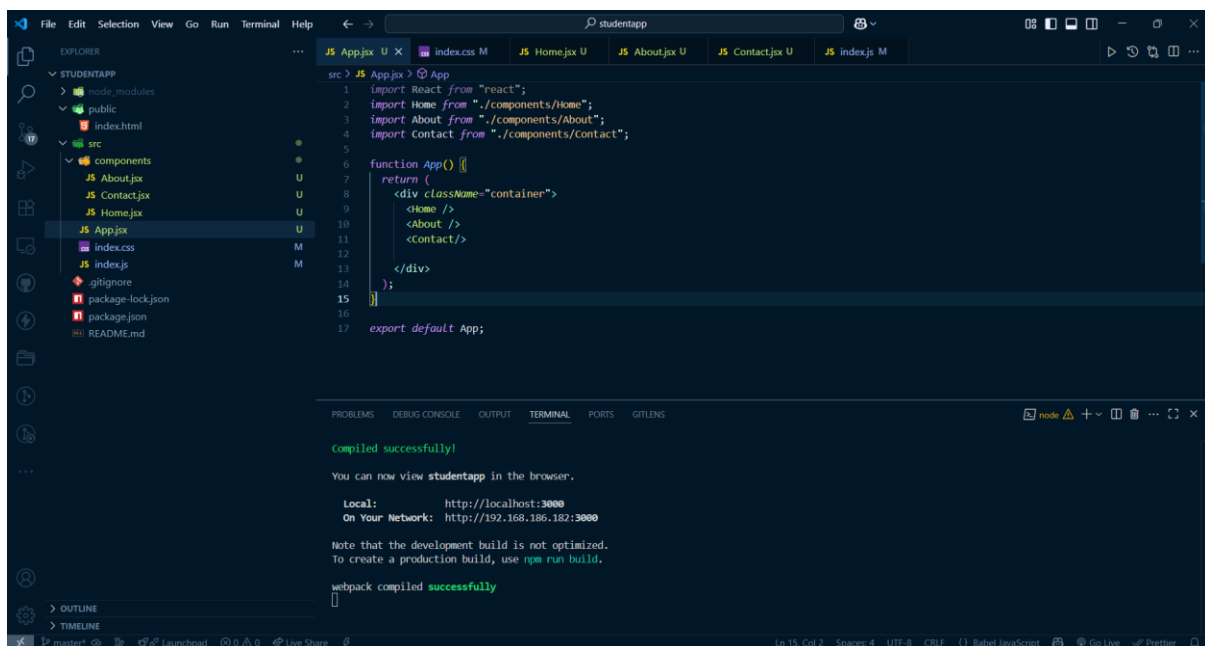
Output



2. ReactJS-HOL

Create a react app for Student Management Portal named StudentApp and create a component named Home which will display the Message “Welcome to the Home page of Student Management Portal”. Create another component named About and display the Message “Welcome to the About page of the Student Management Portal”. Create a third component named Contact and display the Message “Welcome to the Contact page of the Student Management Portal”. Call all the three components

App.jsx



The screenshot shows the VS Code editor with the `App.jsx` file open. The file contains the following code:

```
1 import React from 'react';
2 import Home from './components/Home';
3 import About from './components/About';
4 import Contact from './components/Contact';
5
6 function App() {
7   return (
8     <div className="container">
9       <Home />
10      <About />
11      <Contact />
12    </div>
13  );
14}
15
16 export default App;
```

The Explorer panel on the left shows the project structure with the following files:

- STUDENTAPP
 - node_modules
 - public
 - index.html
 - src
 - components
 - About.jsx
 - Contact.jsx
 - Home.jsx
 - App.jsx
 - index.css
 - index.js
 - .gitignore
 - package-lock.json
 - package.json
 - README.md

The terminal at the bottom shows the following output:

```
Compiled successfully!

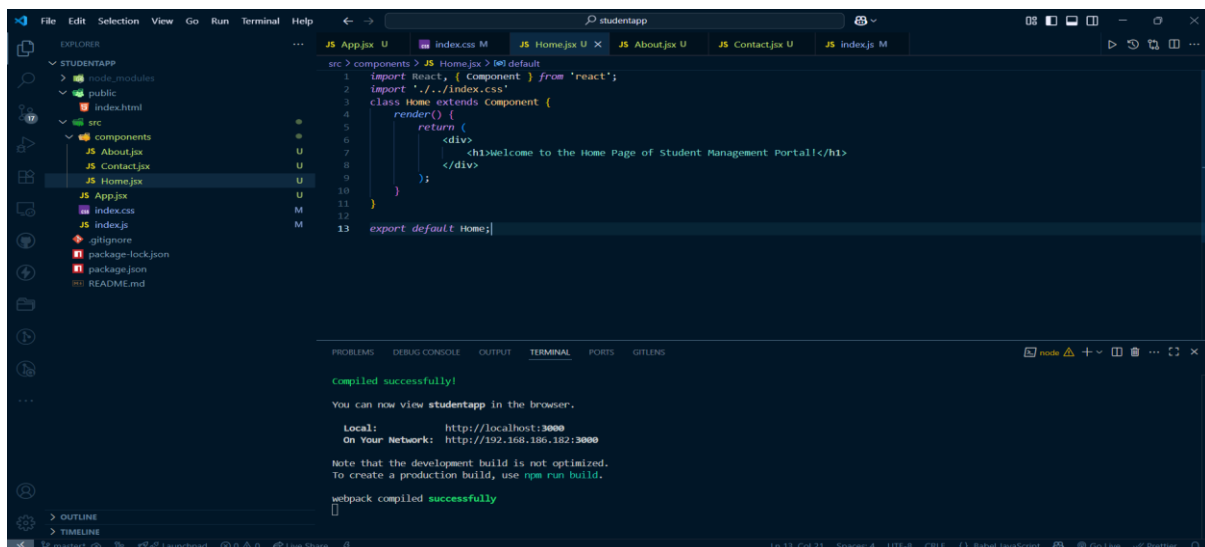
You can now view studentapp in the browser.

Local:      http://localhost:3000
On Your Network:  http://192.168.186.182:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

Component/Home.jsx



The screenshot shows the VS Code editor with the `Home.jsx` file open. The file contains the following code:

```
1 import React, { Component } from 'react';
2 import './index.css';
3 class Home extends Component {
4   render() {
5     return (
6       <div>
7         <h1>Welcome to the Home Page of Student Management Portal</h1>
8       </div>
9     );
10  }
11}
12
13 export default Home;
```

The Explorer panel on the left shows the project structure with the following files:

- STUDENTAPP
 - node_modules
 - public
 - index.html
 - src
 - components
 - About.jsx
 - Contact.jsx
 - Home.jsx
 - App.jsx
 - index.css
 - index.js
 - .gitignore
 - package-lock.json
 - package.json
 - README.md

The terminal at the bottom shows the following output:

```
Compiled successfully!

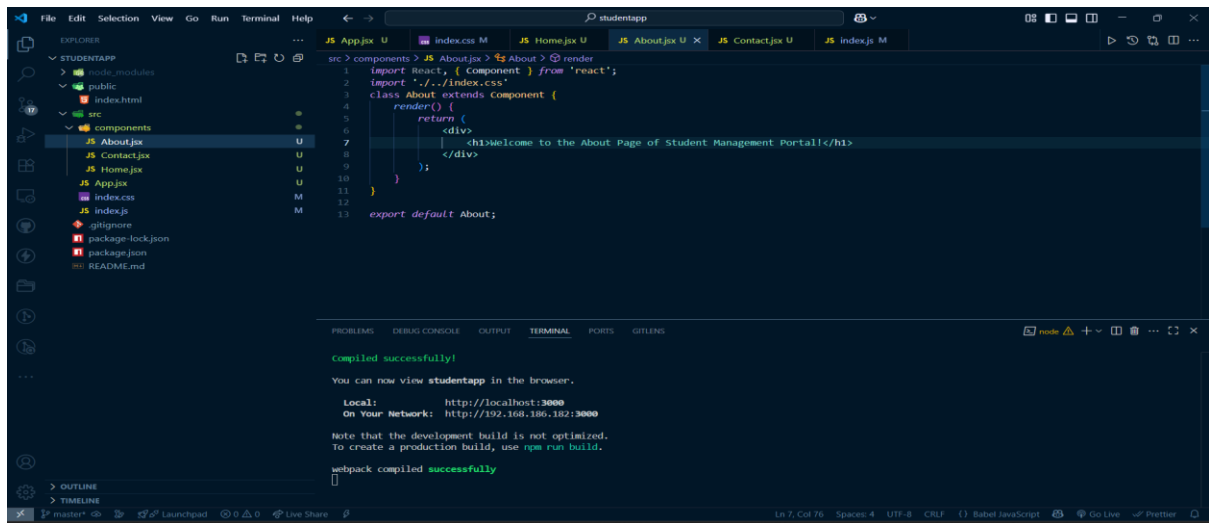
You can now view studentapp in the browser.

Local:      http://localhost:3000
On Your Network:  http://192.168.186.182:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

Component/About.jsx



The screenshot shows the VS Code editor with the `src/components/About.jsx` file open. The code defines a React component named `About` that renders a welcome message. The terminal at the bottom shows the successful compilation of the application.

```
src > components > JS About.jsx > About > render
1 import React, { Component } from 'react';
2 import './index.css'
3 class About extends Component {
4   render() {
5     return (
6       <div>
7         <h1>Welcome to the About Page of Student Management Portal</h1>
8       </div>
9     );
10   }
11 }
12
13 export default About;
```

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS GITLENS

node

Compiled successfully!

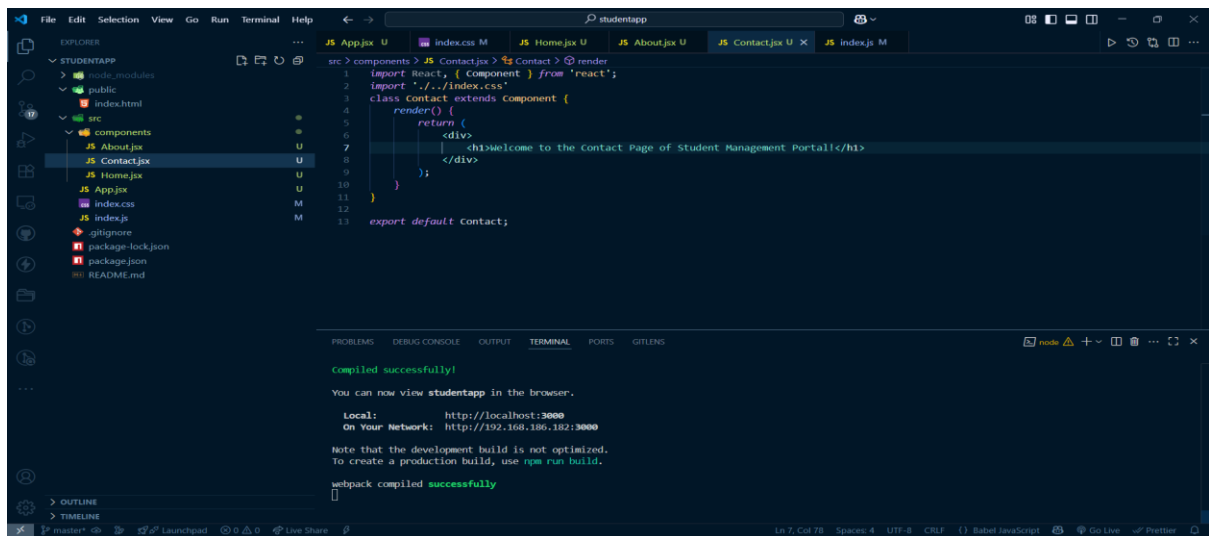
You can now view studentapp in the browser.

Local: http://localhost:3000
On Your Network: http://192.168.186.182:3000

Note that the development build is not optimized.
To create a production build, use `npm run build`.

webpack compiled successfully

Component/Contact.jsx



The screenshot shows the VS Code editor with the `src/components/Contact.jsx` file open. The code defines a React component named `Contact` that renders a welcome message. The terminal at the bottom shows the successful compilation of the application.

```
src > components > JS Contact.jsx > Contact > render
1 import React, { Component } from 'react';
2 import './index.css'
3 class Contact extends Component {
4   render() {
5     return (
6       <div>
7         <h1>Welcome to the Contact Page of Student Management Portal</h1>
8       </div>
9     );
10   }
11 }
12
13 export default Contact;
```

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS GITLENS

node

Compiled successfully!

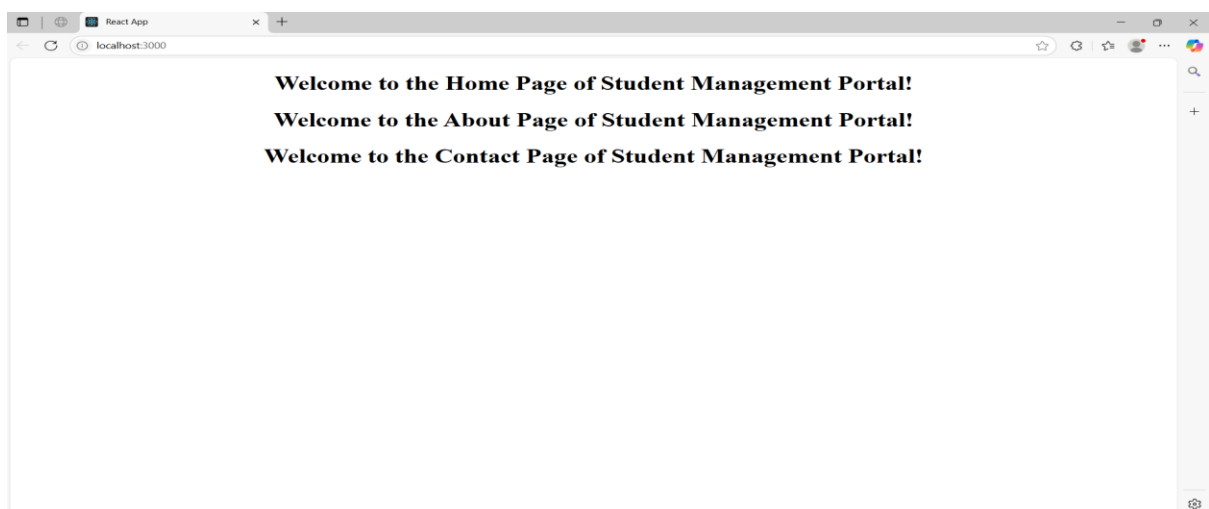
You can now view studentapp in the browser.

Local: http://localhost:3000
On Your Network: http://192.168.186.182:3000

Note that the development build is not optimized.
To create a production build, use `npm run build`.

webpack compiled successfully

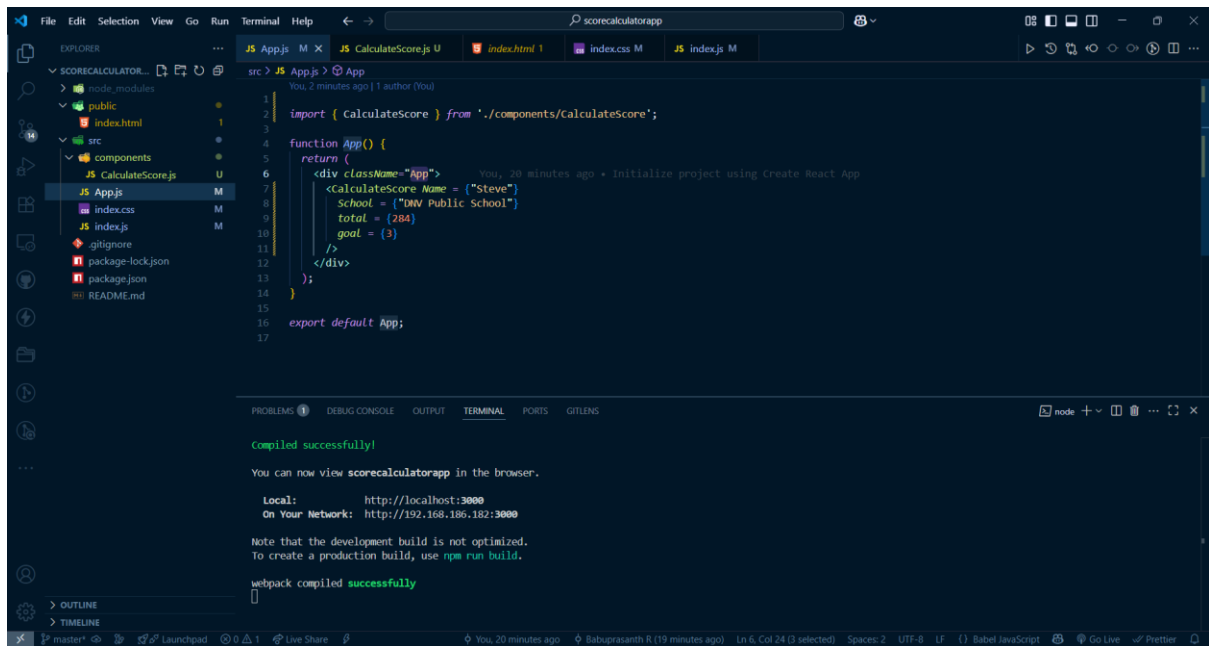
Output



3. ReactJS-HOL

Create a react app for Student Management Portal named scorecalculatorapp and create a function component named “CalculateScore” which will accept Name, School, Total and goal in order to calculate the average score of a student and display the same.

App.js



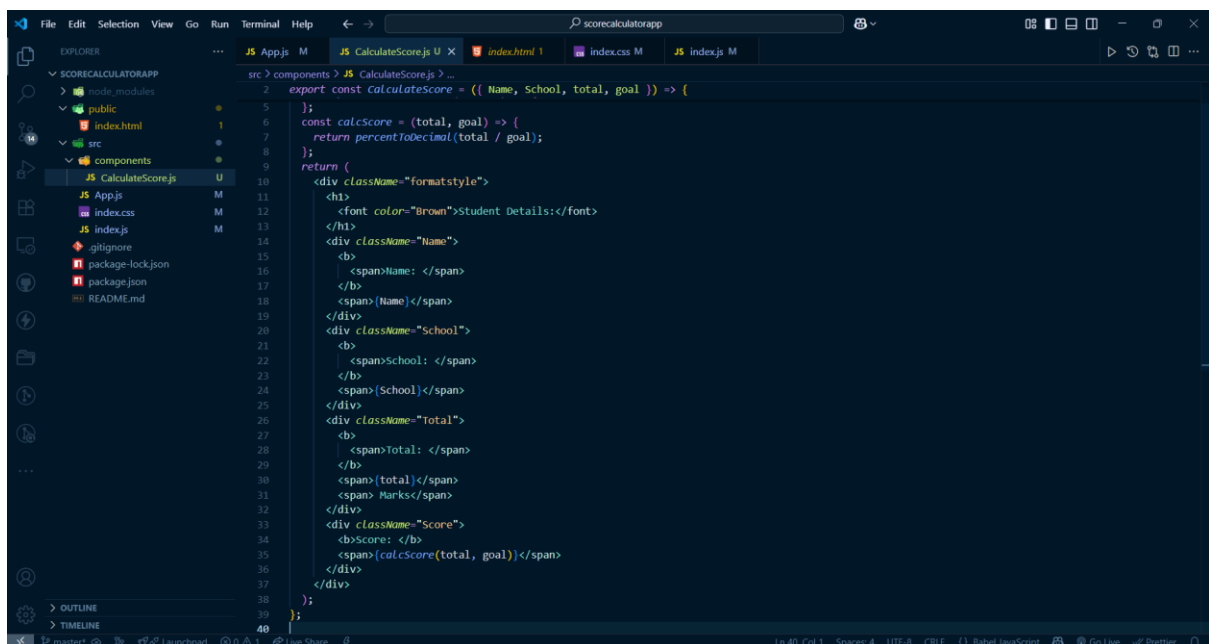
The screenshot shows the VS Code editor with the file explorer on the left displaying the project structure for 'scorecalculatorapp'. The 'App.js' file is open in the editor, showing the following code:

```
1 import { CalculateScore } from './components/CalculateScore';
2
3
4 function App() {
5   return (
6     <div className="App">
7       <CalculateScore Name = ("Steve")
8         School = ("DNV Public School")
9         total = (284)
10        goal = (3)
11      />
12    </div>
13  );
14}
15
16 export default App;
```

The terminal at the bottom shows the following output:

```
Compiled successfully!
You can now view scorecalculatorapp in the browser.
Local: http://localhost:3000
On Your Network: http://192.168.186.182:3000
Note that the development build is not optimized.
To create a production build, use npm run build.
webpack compiled successfully
```

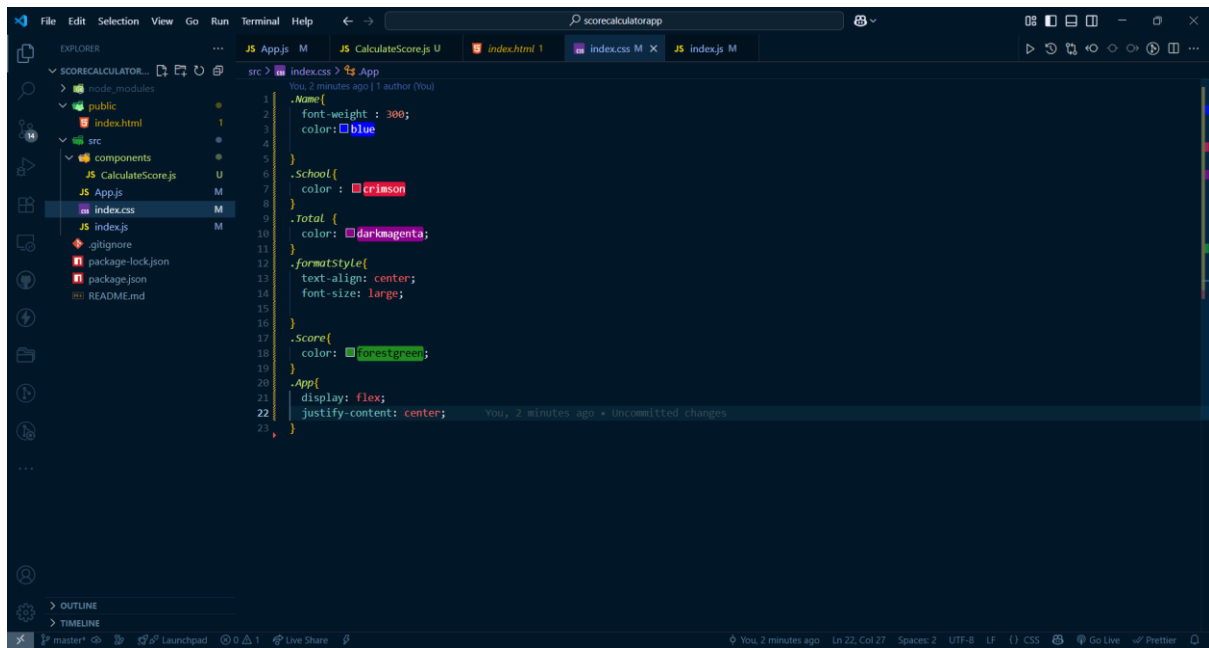
Components/CalculateScore.js



The screenshot shows the VS Code editor with the file explorer on the left displaying the project structure for 'scorecalculatorapp'. The 'CalculateScore.js' file is open in the editor, showing the following code:

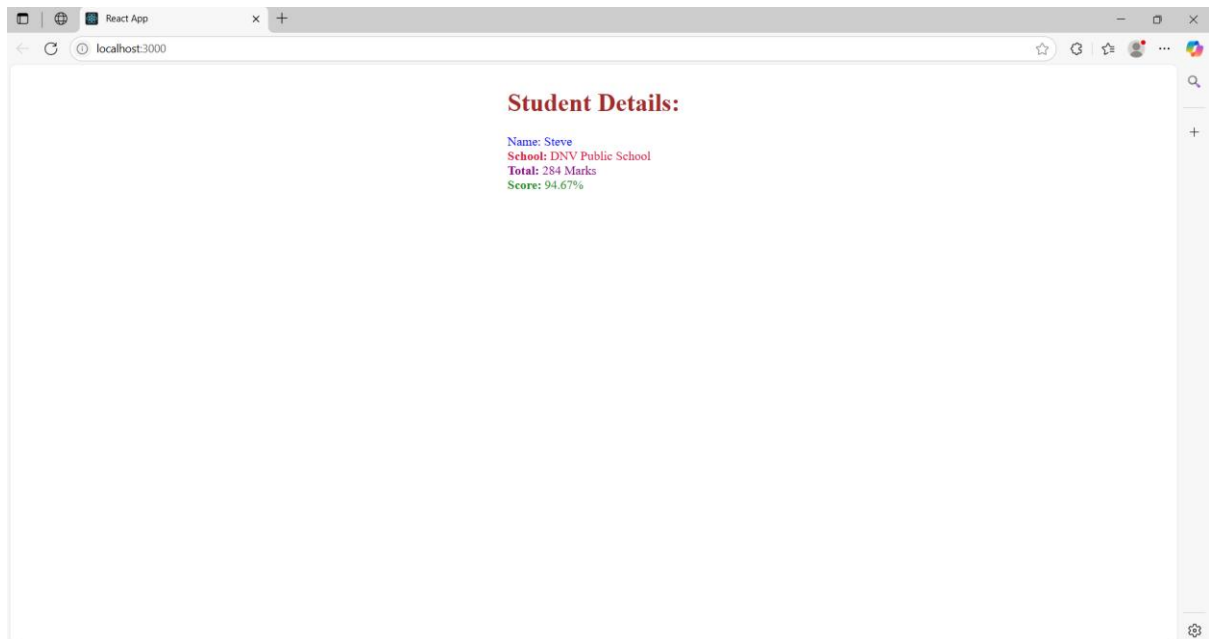
```
1 export const CalculateScore = ({ Name, School, total, goal }) => {
2
3   const calcScore = (total, goal) => {
4     return percentToDecimal(total / goal);
5   };
6   return (
7     <div className="formatstyle">
8       <h1>
9         <font color="Brown">Student Details:</font>
10      </h1>
11      <div className="Name">
12        <b>
13          <span>Name: </span>
14        </b>
15        <span>{Name}</span>
16      </div>
17      <div className="School">
18        <b>
19          <span>School: </span>
20        </b>
21        <span>{School}</span>
22      </div>
23      <div className="Total">
24        <b>
25          <span>Total: </span>
26        </b>
27        <span>{total}</span>
28      </div>
29      <div className="Score">
30        <b>Score: </b>
31        <span>{calcScore(total, goal)}</span>
32      </div>
33    </div>
34  );
35};
```

Index.css



```
1 .Name{
2   font-weight : 300;
3   color: blue
4 }
5
6 .School{
7   color : crimson
8 }
9
10 .Total {
11   color: darkmagenta;
12 }
13
14 .formatStyle{
15   text-align: center;
16   font-size: large;
17 }
18
19 .Score{
20   color: forestgreen;
21 }
22
23 .App{
24   display: flex;
25   justify-content: center;
26 }
```

Output

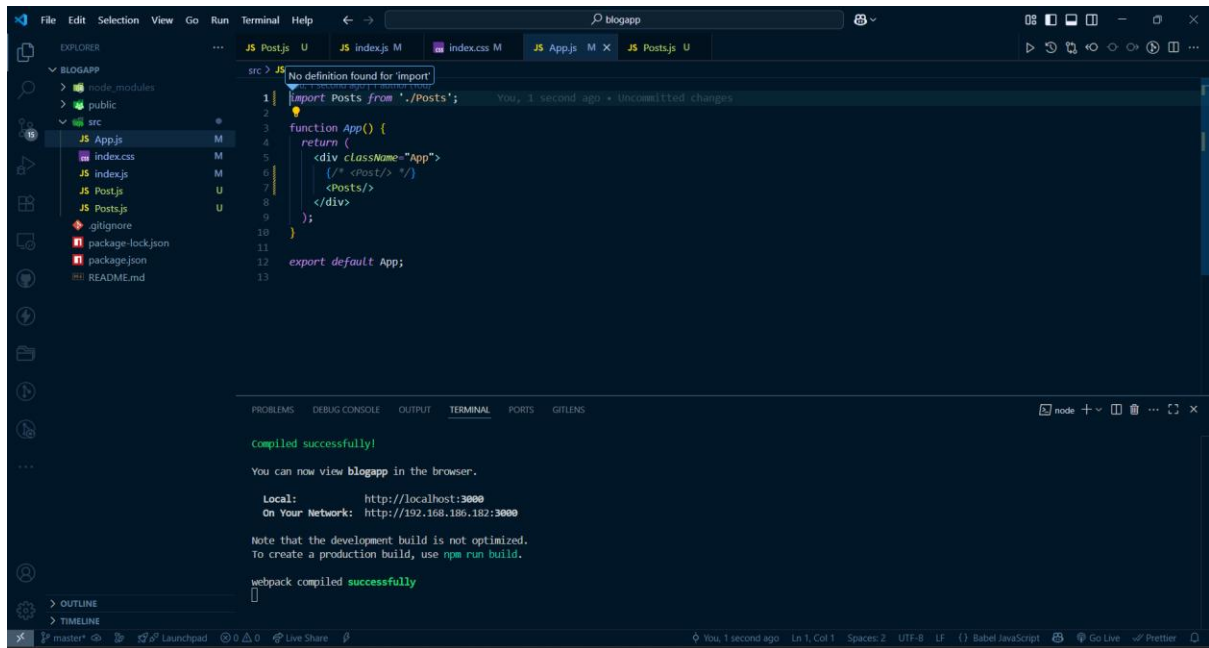


4. ReactJS-HOL

Implement componentDidMount() hook

Implementing componentDidCatch() life cycle hook.

App.js



The screenshot shows the VS Code editor with the file explorer on the left displaying the project structure. The main editor window shows the code for App.js. The code defines an App component that imports Posts from './Posts' and renders a div with className 'App' containing a Post component. The terminal at the bottom shows the output of the build process, indicating a successful compilation and providing the local and network URLs for viewing the application in the browser.

```
src > JS
1 | import Posts from './Posts';
2 |
3 | function App() {
4 |   return (
5 |     <div className="App">
6 |       <Post />
7 |     </div>
8 |   );
9 | }
10 |
11 | export default App;
```

compiled successfully!

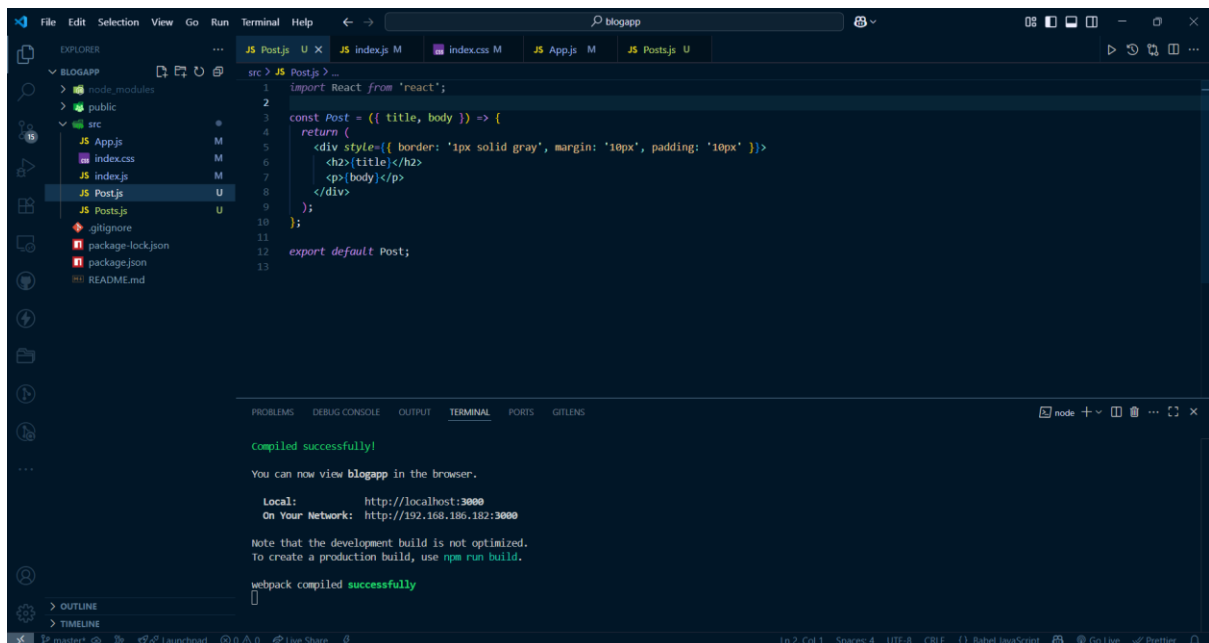
You can now view blogapp in the browser.

Local: http://localhost:3000
On Your Network: http://192.168.186.182:3000

Note that the development build is not optimized.
To create a production build, use `npm run build`.

webpack compiled successfully

Post.js



The screenshot shows the VS Code editor with the file explorer on the left displaying the project structure. The main editor window shows the code for Post.js. The code defines a Post component that takes title and body as props and renders a div with a border and padding, containing a title and body. The terminal at the bottom shows the output of the build process, indicating a successful compilation and providing the local and network URLs for viewing the application in the browser.

```
src > JS Post.js
1 | import React from 'react';
2 |
3 | const Post = ({ title, body }) => {
4 |   return (
5 |     <div style={{ border: '1px solid gray', margin: '10px', padding: '10px' }}>
6 |       <h2>{title}</h2>
7 |       <p>{body}</p>
8 |     </div>
9 |   );
10 | };
11 |
12 | export default Post;
```

compiled successfully!

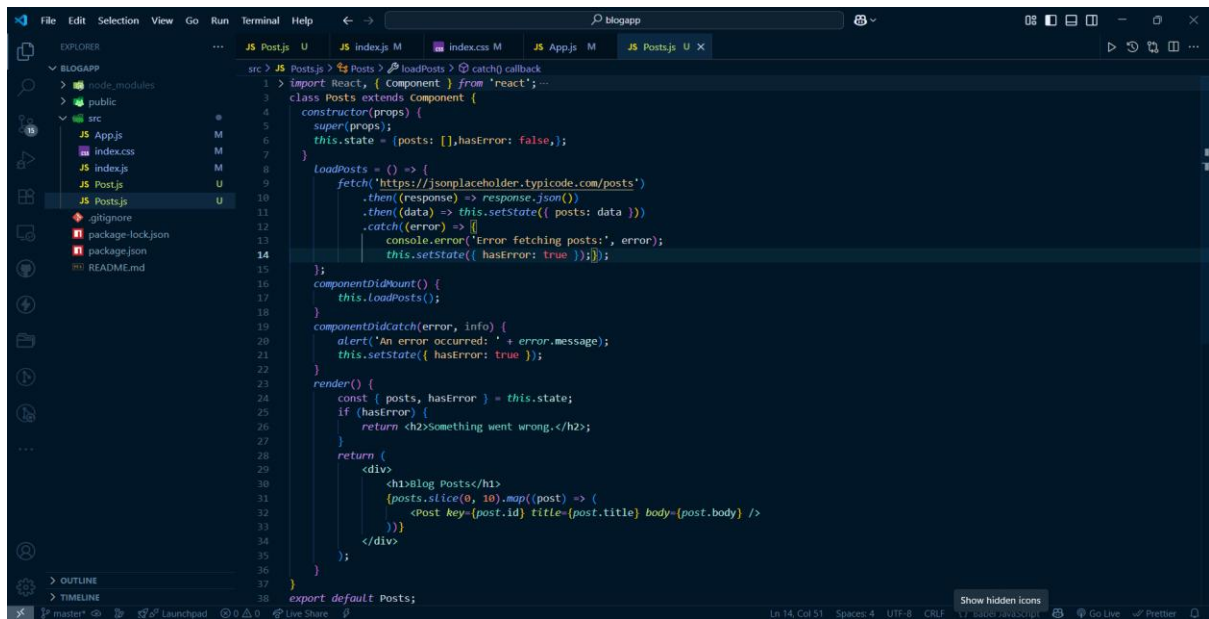
You can now view blogapp in the browser.

Local: http://localhost:3000
On Your Network: http://192.168.186.182:3000

Note that the development build is not optimized.
To create a production build, use `npm run build`.

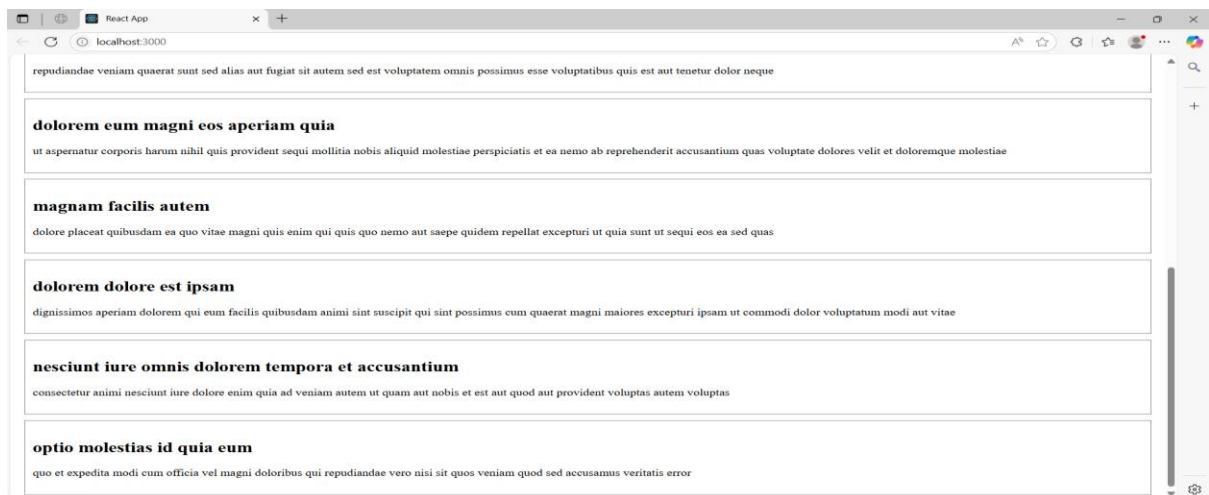
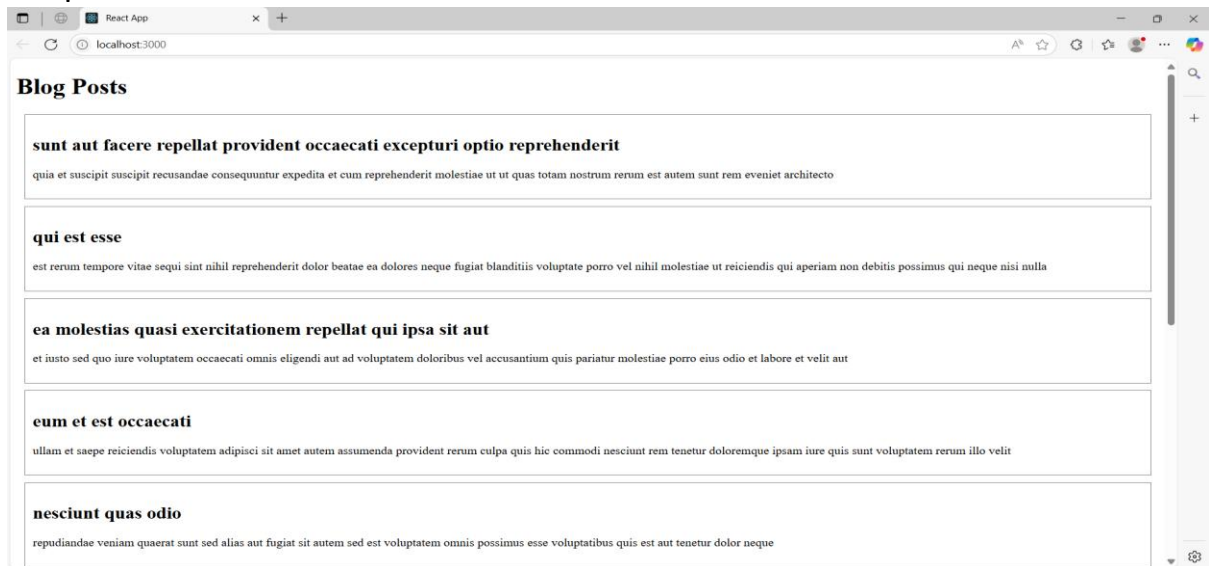
webpack compiled successfully

Posts.js



```
src > JS Posts.js > Posts > loadPosts > catch() callback
1 > Import React, { Component } from 'react'; ...
2 class Posts extends Component {
3   constructor(props) {
4     super(props);
5     this.state = { posts: [], hasError: false; };
6   }
7   loadPosts = () => {
8     fetch('https://jsonplaceholder.typicode.com/posts')
9       .then(response => response.json())
10      .then(data => this.setState({ posts: data }))
11      .catch(error => {
12        console.error('Error fetching posts:', error);
13        this.setState({ hasError: true });
14      });
15  };
16  componentDidMount() {
17    this.loadPosts();
18  }
19  componentDidCatch(error, info) {
20    alert('An error occurred: ' + error.message);
21    this.setState({ hasError: true });
22  }
23  render() {
24    const { posts, hasError } = this.state;
25    if (hasError) {
26      return <h2>Something went wrong.</h2>;
27    }
28    return (
29      <div>
30        <h1>Blog Posts</h1>
31        {posts.slice(0, 10).map((post) => (
32          <Post key={post.id} title={post.title} body={post.body} />
33        ))}
34      </div>
35    );
36  }
37 }
38 export default Posts;
```

Output



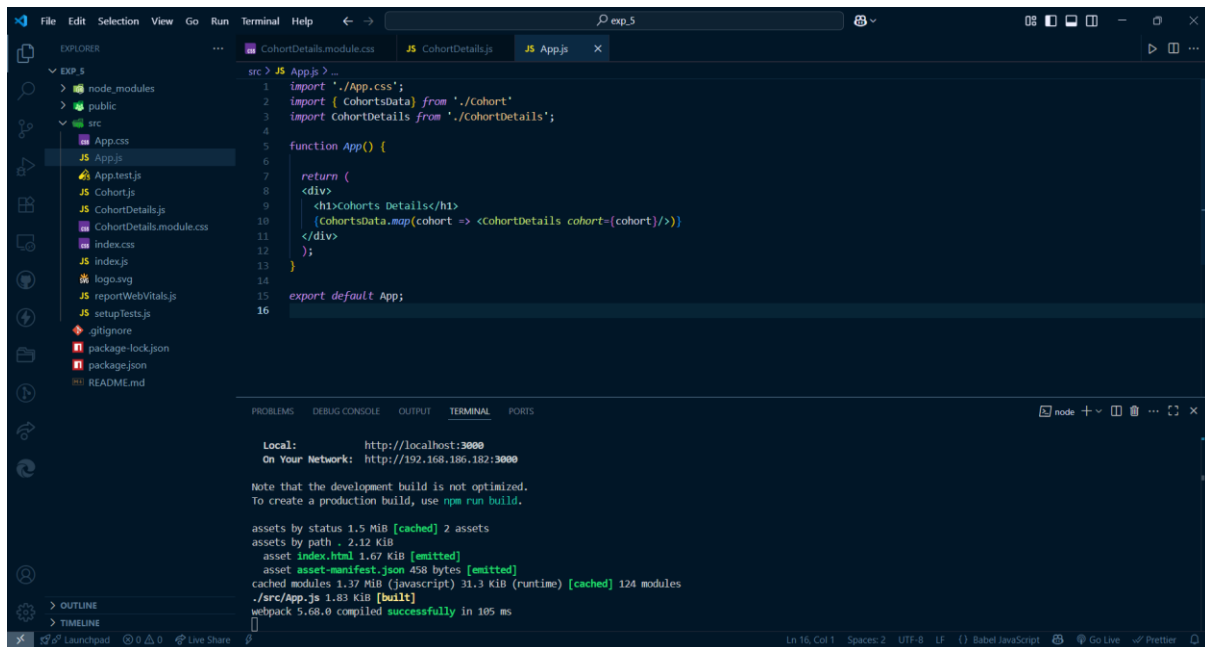
5. ReactJS-HOL

Style a react component

Define styles using the CSS Module

Apply styles to components using className and style properties

App.js



The screenshot shows the VS Code editor with the file explorer on the left displaying the project structure. The main editor window shows the `App.js` file. The code imports `App.css`, `CohortData`, and `CohortDetails`. It defines a `App` function that returns a JSX element with a `h1` and a `div` containing a map of `CohortDetails` components. The terminal at the bottom shows the development server running on `localhost:3000` and provides asset statistics.

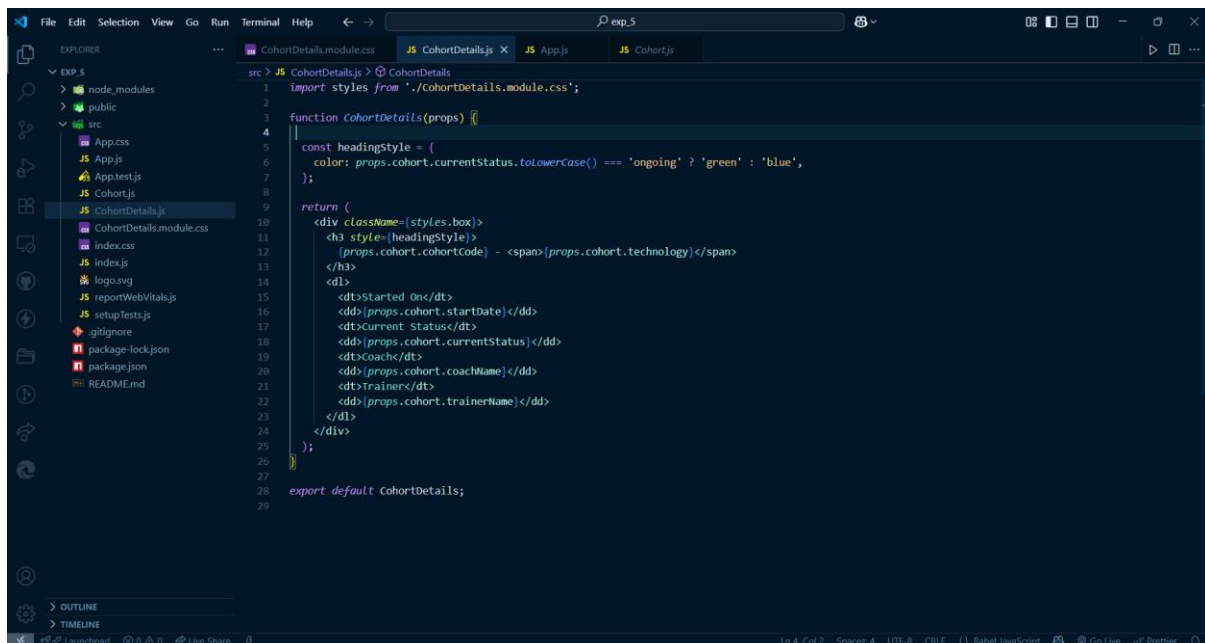
```
src > JS App.js >
1 import './App.css';
2 import { CohortData } from './cohort'
3 import CohortDetails from './CohortDetails';
4
5 function App() {
6
7   return (
8     <div>
9       <h1>Cohorts Details</h1>
10      <div>
11        {CohortData.map(cohort => <CohortDetails cohort={cohort}/>)}
12      </div>
13    </div>
14  );
15
16  export default App;
```

Local: http://localhost:3000
On Your Network: http://192.168.186.182:3000

Note that the development build is not optimized.
To create a production build, use `npm run build`.

assets by status 1.5 MiB [cached] 2 assets
assets by path 2.12 KiB
asset index.html 1.07 KiB [emitted]
asset asset-manifest.json 458 bytes [emitted]
cached modules 1.37 MiB (javascript) 31.3 KiB (runtime) [cached] 124 modules
./src/app.js 1.83 KiB [built]
webpack 5.68.0 compiled successfully in 105 ms

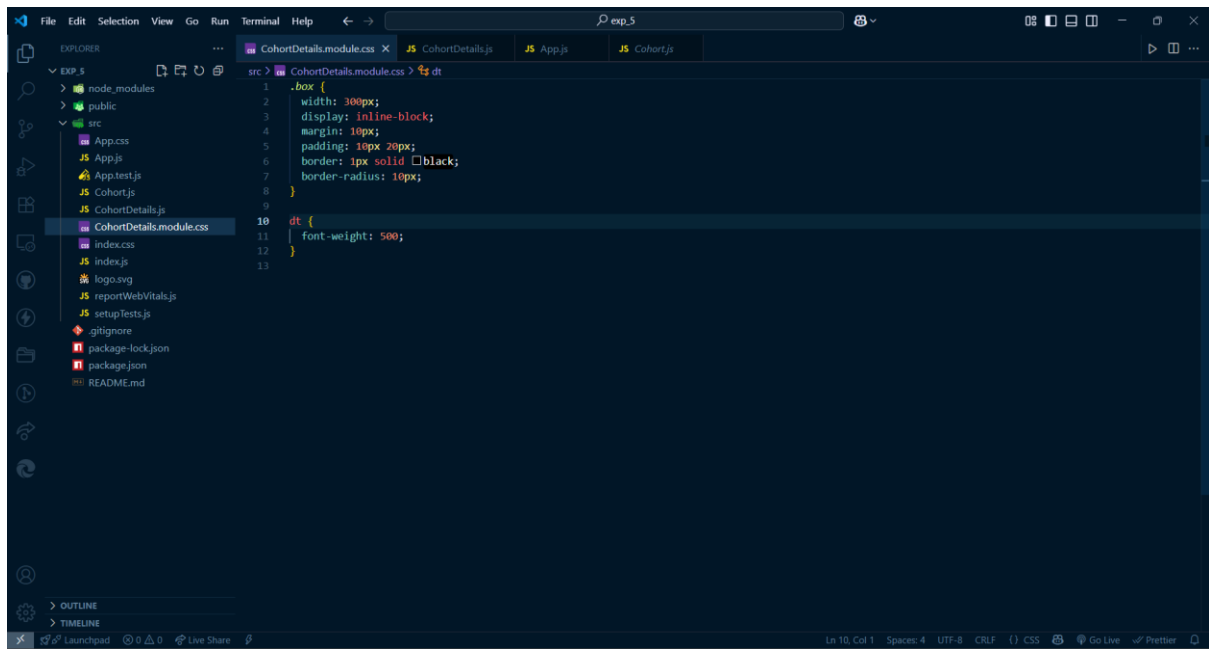
CohortDetails.js



The screenshot shows the VS Code editor with the file explorer on the left. The main editor window shows the `CohortDetails.js` file. The code imports `styles` from `./CohortDetails.module.css` and defines a `CohortDetails` function that takes `props` and returns a JSX element. The JSX element uses `className` and `style` props to apply styles to a `div` and its children.

```
src > JS CohortDetails.js > CohortDetails
1 import styles from './CohortDetails.module.css';
2
3 function CohortDetails(props) {
4
5   const headingStyle = {
6     color: props.cohort.currentStatus.toLowerCase() === 'ongoing' ? 'green' : 'blue',
7   };
8
9   return (
10     <div className={styles.box}>
11       <h3 style={headingStyle}>
12         {props.cohort.cohortCode} - <span>{props.cohort.technology}</span>
13       </h3>
14       <dl>
15         <dt>Started on</dt>
16         <dd>{props.cohort.startDate}</dd>
17         <dt>Current Status</dt>
18         <dd>{props.cohort.currentStatus}</dd>
19         <dt>Coach</dt>
20         <dd>{props.cohort.coachName}</dd>
21         <dt>Trainer</dt>
22         <dd>{props.cohort.trainerName}</dd>
23       </dl>
24     </div>
25   );
26
27   export default CohortDetails;
```

CohortDetails.module.css



The screenshot shows the VS Code editor with the file `CohortDetails.module.css` open. The file contains the following CSS code:

```
1 .box {
2   width: 300px;
3   display: inline-block;
4   margin: 10px;
5   padding: 10px 20px;
6   border: 1px solid black;
7   border-radius: 10px;
8 }
9
10 dt {
11   font-weight: 500;
12 }
13
```

The Explorer sidebar on the left shows the project structure with files like `App.css`, `App.js`, `App.test.js`, `Cohort.js`, `CohortDetails.js`, `CohortDetails.module.css`, `index.css`, `index.js`, `logo.svg`, `reportWebVitals.js`, `setupTests.js`, `.gitignore`, `package-lock.json`, `package.json`, and `README.md`.

Output

