**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.

1. To create a new React app, Install Nodejs and Npm from the following link:

<https://nodejs.org/en/download/>

1. Install Create-react-app by running the following command in the command prompt:



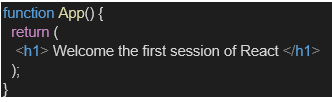
1. To create a React Application with the name of “myfirstreact”, type the following command:



1. Once the App is created, navigate into the folder of myfirstreact by typing the following command:



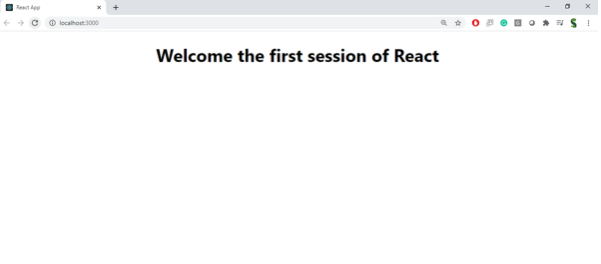
1. Open the folder of myfirstreact in Visual Studio Code
2. Open the App.js file in Src Folder of myfirstreact
3. Remove the current content of “App.js”
4. Replace it with the following:



1. Run the following command to execute the React application:



1. Open a new browser window and type “localhost:3000” in the address bar



**Source Code:**

**App.js**

import "./App.css";

function App() {

return (

<div className="App">

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

</div>

);

}

export default App;

**Command Line:**

PS C:\Users\ramgo\Documents\cognizant\_company\deepskilling\solutions\ReactJS> cd myfirstreact PS C:\Users\ramgo\Documents\cognizant\_company\deepskilling\solutions\ReactJS\myfirstreact> npm start

[myfirstreact@0.1.0](mailto:myfirstreact@0.1.0) start react-scripts start

(node:19012) [DEP\_WEBPACK\_DEV\_SERVER\_ON\_AFTER\_SETUP\_MIDDLEWARE] DeprecationWarning: 'onAfterSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option. (Use node --trace-deprecation ... to show where the warning was created) (node:19012) [DEP\_WEBPACK\_DEV\_SERVER\_ON\_BEFORE\_SETUP\_MIDDLEWARE] DeprecationWarning: 'onBeforeSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option. Starting the development server...

Compiled successfully!

You can now view myfirstreact in the browser.

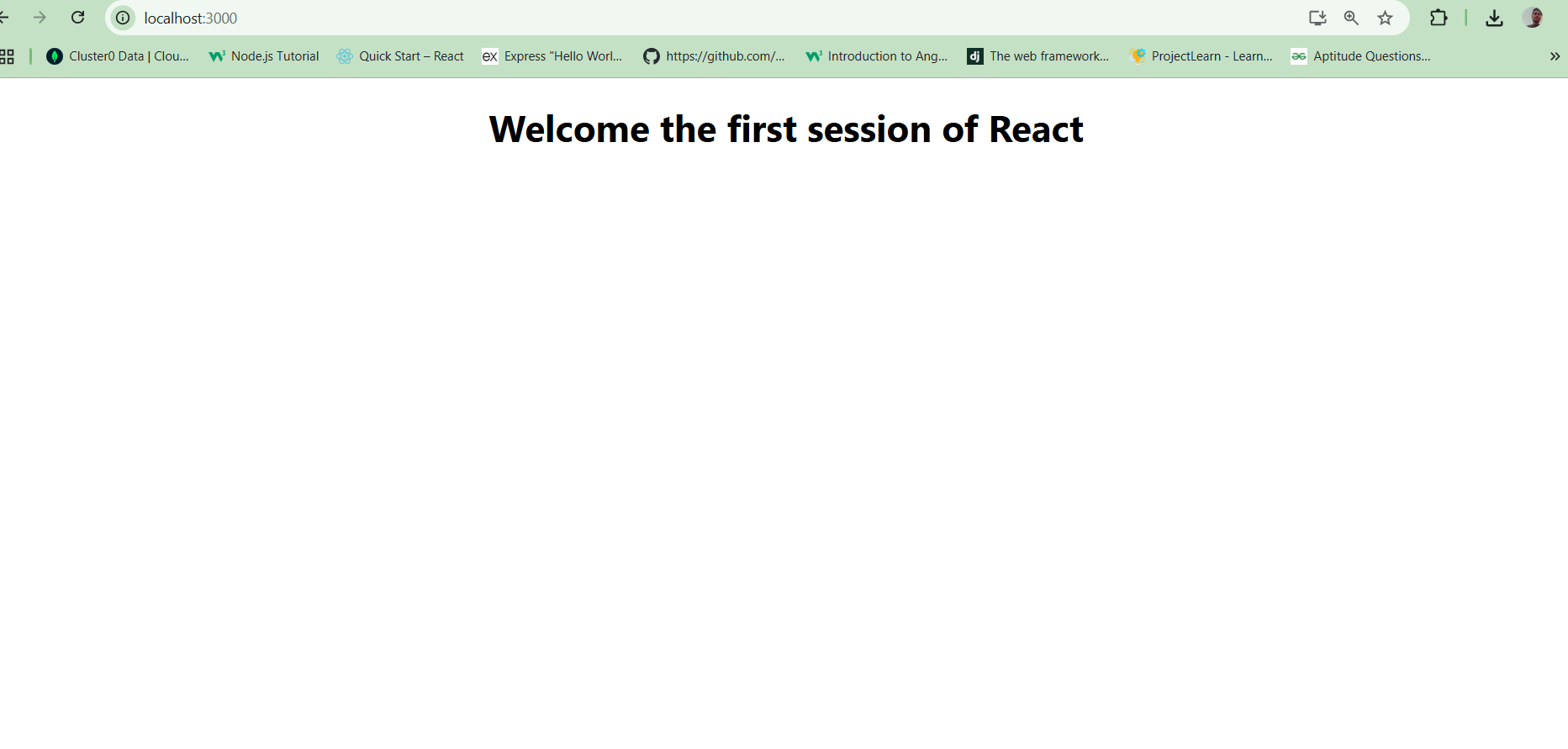
Local: [http://localhost:3000](http://localhost:3000/) On Your Network: [http://192.168.1.23:3000](http://192.168.1.23:3000/)

Note that the development build is not optimized.

To create a production build, use npm run build.

webpack compiled successfully

**Output:**



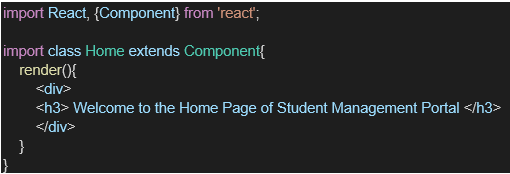
**2.ReactJS - HOL**

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.

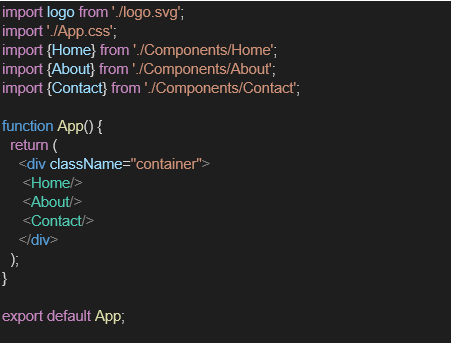
1. Create a React project named “StudentApp” type the following command in terminal of Visual studio:



1. Create a new folder under Src folder with the name “Components”. Add a new file named “Home.js”
2. Type the following code in Home.js



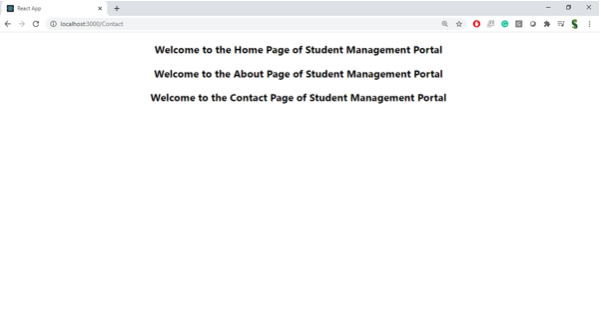
1. Under Src folder add another file named “About.js”
2. Repeat the same steps for Creating “About” and “Contact” component by adding a new file as ”About.js”, “Contact.js” under “Src” folder and edit the code as mentioned for “Home” Component.
3. Edit the App.js to invoke the Home, About and Contact component as follows:



1. In command Prompt, navigate into StudentApp and execute the code by typing the following command:



1. Open browser and type “localhost:3000” in the address bar:



**Source Code:**

**App.js**

import "./App.css";

import About from "./Components/About";

import Contact from "./Components/Contact";

import Home from "./Components/Home";

function App() {

return (

<div className="App">

<Home />

<About />

<Contact />

</div>

);

}

export default App;

**./Components/Home.js**

import { Component } from "react";

class Home extends Component {

render() {

return (

<div>

<h1>Welcome to the Home Page of Student Management Portal</h1>

</div>

);

}

}

export default Home;

**./Components/Contact.js**

import { Component } from "react";

class Contact extends Component {

render() {

return (

<div>

<h1>Welcome to the Contact Page of Student Management Portal</h1>

</div>

);

}

}

export default Contact;

**./Components/About.js**

import { Component } from "react";

class About extends Component {

render() {

return (

<div>

<h1>Welcome to the About Page of Student Management Portal</h1>

</div>

);

}

}

export default About;

**Command Line:**

PS C:\Users\ramgo\Documents\cognizant\_company\deepskilling\solutions\ReactJS\studentapp> npm start

[studentapp@0.1.0](mailto:studentapp@0.1.0) start react-scripts start

(node:14264) [DEP\_WEBPACK\_DEV\_SERVER\_ON\_AFTER\_SETUP\_MIDDLEWARE] DeprecationWarning: 'onAfterSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option. (Use node --trace-deprecation ... to show where the warning was created) (node:14264) [DEP\_WEBPACK\_DEV\_SERVER\_ON\_BEFORE\_SETUP\_MIDDLEWARE] DeprecationWarning: 'onBeforeSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option. Starting the development server... Compiled successfully!

You can now view studentapp in the browser.

webpack compiled successfully Compiling... Compiled successfully!

You can now view studentapp in the browser.

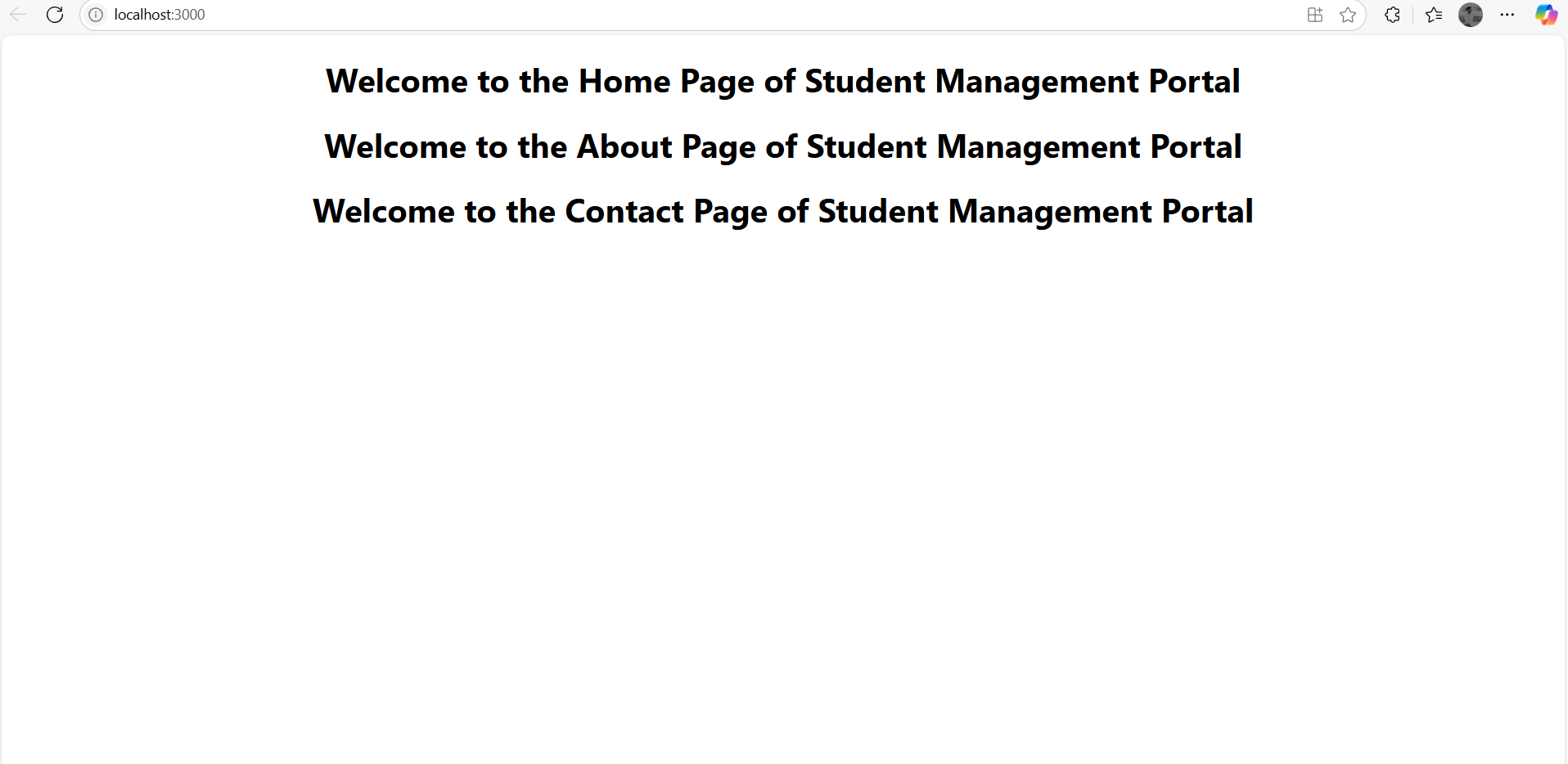
Local: [http://localhost:3000](http://localhost:3000/) On Your Network: [http://192.168.1.23:3000](http://192.168.1.23:3000/)

Note that the development build is not optimized.

To create a production build, use npm run build.

webpack compiled successfully

**Output:**



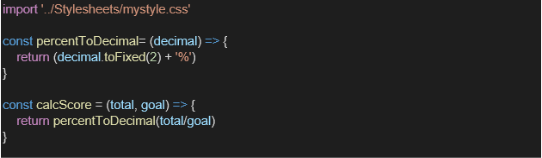
**3.ReactJS - HOL**

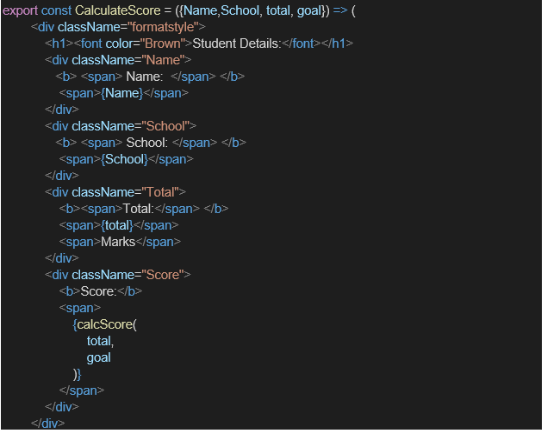
Create a react app for Student Management Portal named scorecalculatorapp and create a function component named “CalculateScore” which will accept Name, School, Total and goal in order to calculate the average score of a student and display the same.

1. Create a React project named “scorecalculatorapp” type the following command in terminal of Visual studio:

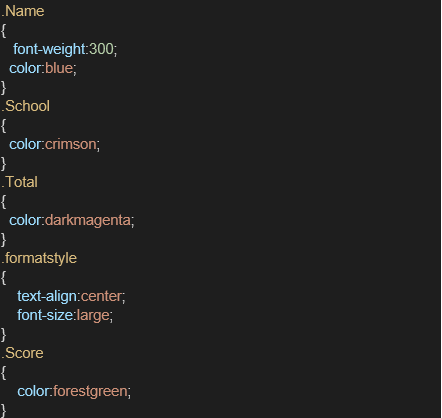
Inserting image...

1. Create a new folder under Src folder with the name “Components”. Add a new file named “CalculateScore.js”
2. Type the following code in CalculateScore.js

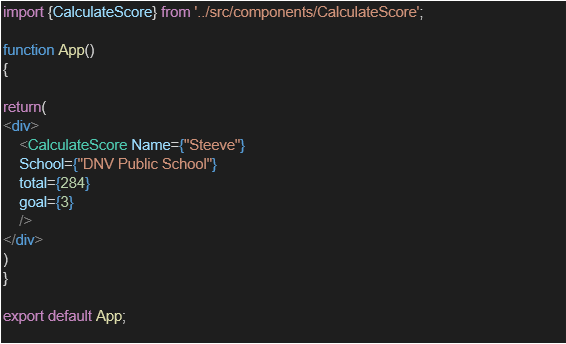




1. Create a Folder named Stylesheets and add a file named “mystyle.css” in order to add some styles to the components:



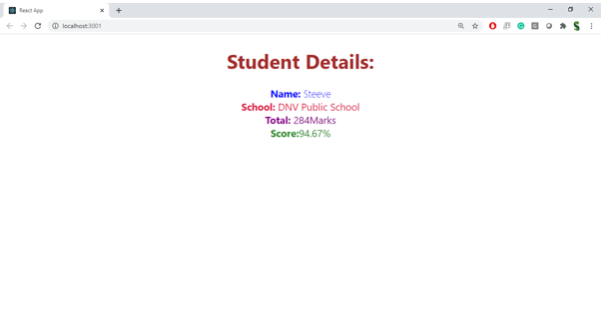
1. Edit the App.js to invoke the CalculateScore functional component as follows:



1. In command Prompt, navigate into scorecalculatorapp and execute the code by typing the following command:

Inserting image...

1. Open browser and type “localhost:3000” in the address bar:



**Source Code:**

**App.js**

import "./App.css";

import { CalculateScore } from "./Components/CalculateScore";

function App() {

return (

<div className="App">

<CalculateScore

Name="Steeve"

School="DNV Public School"

total={284}

goal={3}

/>

</div>

);

}

export default App;

**./Components/CalculateScore.js**

import "./CalculateScore.css";

const percentToDecimal = (percent) => {

return percent.toFixed(2) + "%";

};

const CalcScore = (total, goal) => {

return percentToDecimal(total / goal);

};

function CalculateScore({ Name, School, total, goal }) {

return (

<div className="formatstyle">

<font color="brown">Student Details:</font>

<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 { CalcScore, CalculateScore, percentToDecimal };

**./Components/CalculateScore.css**

.Name {

font-weight: 300;

color: blue;

}

.School {

color: crimson;

}

.Total {

color: darkmagenta;

}

.formatstyle {

text-align: center;

font-size: large;

}

.Score {

color: forestgreen;

}

**Command Line:**

Note that the development build is not optimized.

To create a production build, use npm run build.

webpack compiled successfully

Compiling...

Compiled successfully!

You can now view scorecalculatorapp in the browser.

Local: http://localhost:3000

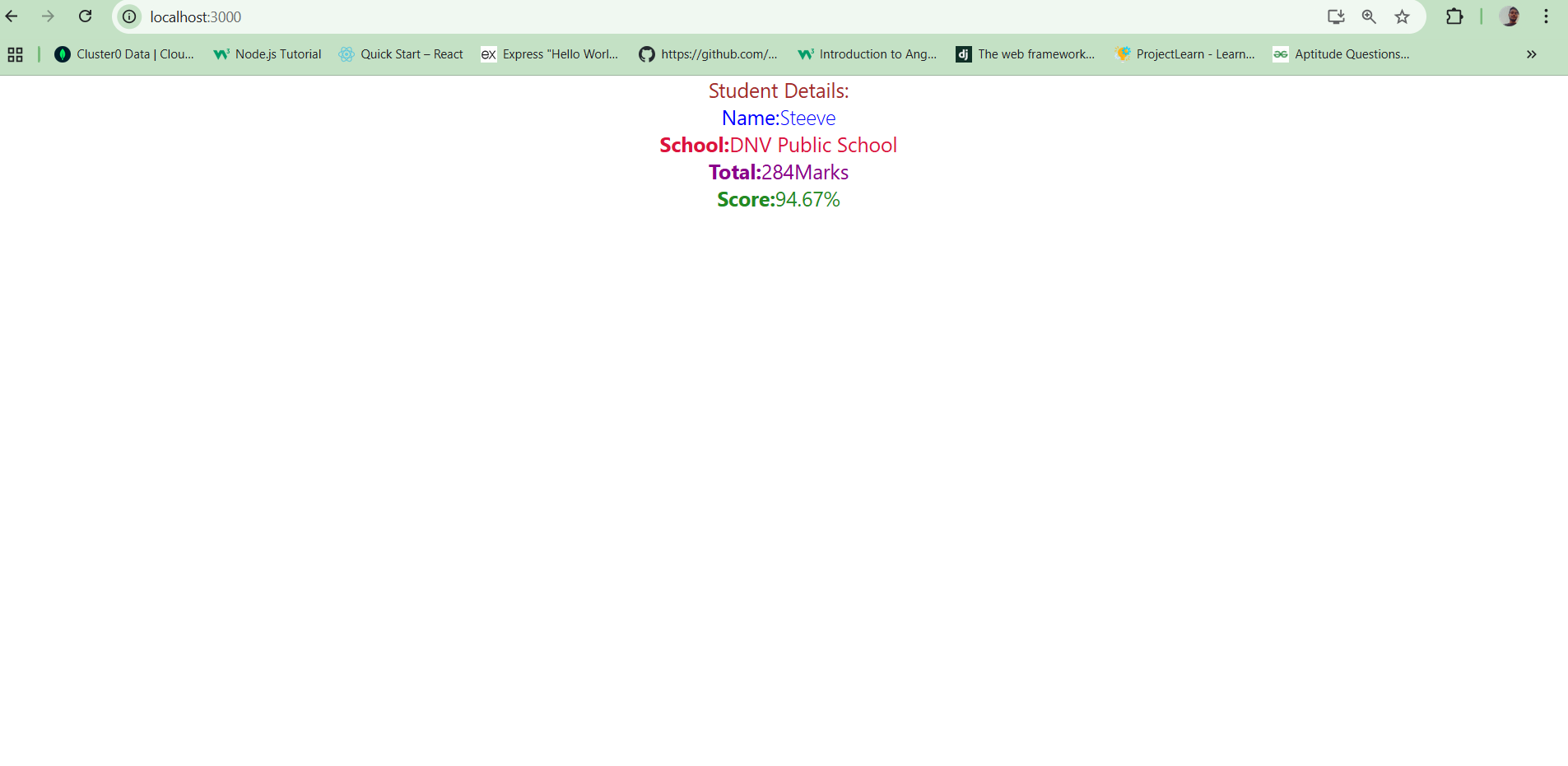
On Your Network: http://192.168.1.23:3000

Note that the development build is not optimized.

To create a production build, use npm run build.

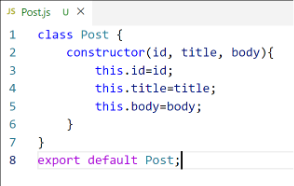
webpack compiled successfully

**Ouptut:**



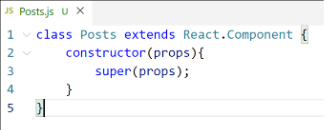
**4.ReactJS - HOL**

1. Create a new react application using *create-react-app* tool with the name as “blogapp”
2. Open the application using VS Code
3. Create a new file named as **Post.js** in **src folder** with following properties



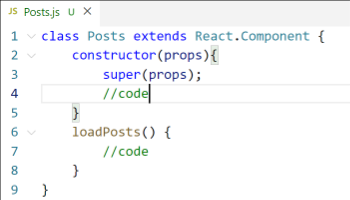
*Figure 2: Post class*

1. Create a new class based component named as **Posts** inside **Posts.js** file



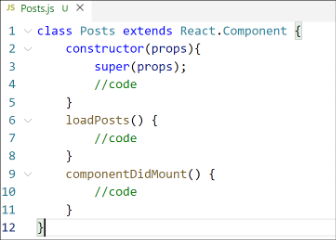
*Figure 3: Posts Component*

1. Initialize the component with a list of Post in state of the component using the constructor
2. Create a new method in component with the name as **loadPosts()** which will be responsible for using Fetch API and assign it to the component state created earlier. To get the posts use the url (<https://jsonplaceholder.typicode.com/posts>)



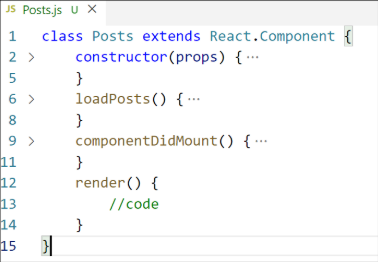
*Figure 4: loadPosts() method*

1. Implement the **componentDidMount()** hook to make calls to **loadPosts()** which will fetch the posts



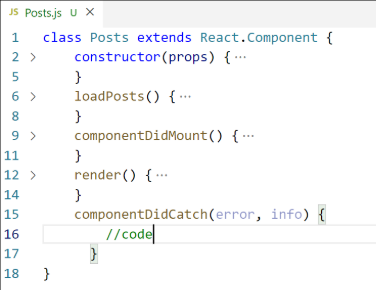
*Figure 5: componentDidMount() hook*

1. Implement the **render()** which will display the title and post of posts in html page using heading and paragraphs respectively.



*Figure 6: render() method*

1. Define a **componentDidCatch()** method which will be responsible for displaying any error happing in the component as alert messages.



*Figure 7: componentDidCatch() hook*

1. Add the Posts component to App component.
2. Build and Run the application using *npm start* command.

**Source Code:**

**App.js**

import "./App.css";

import Posts from "./Posts";

function App() {

return (

<div className="App">

<Posts />

</div>

);

}

export default App;

**Post.js**

import { Component } from "react";

class Post extends Component {

render() {

const { title, body } = this.props;

return (

<div

style={{ border: "1px solid #ddd", margin: "10px", padding: "10px" }}

>

<h3>{title}</h3>

<p>{body}</p>

</div>

);

}

}

export default Post;

**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((response) => {

if (!response.ok) {

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

}

return response.json();

})

.then((data) => {

console.log("Fetched posts:", data);

this.setState({ posts: data });

})

.catch((error) => {

console.error("Error fetching posts:", error);

alert("Failed to load posts!");

});

};

componentDidMount() {

this.loadPosts();

}

componentDidCatch(error, info) {

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

alert("Something went wrong!");

this.setState({ hasError: true });

}

render() {

if (this.state.hasError) {

return <h2>Something went wrong while displaying posts.</h2>;

}

return (

<div>

<h1>Blog Posts</h1>

{this.state.posts.length === 0 ? (

<p>Loading posts...</p>

) : (

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

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

))

)}

</div>

);

}

}

export default Posts;

**Command Line:**

webpack compiled successfully

Compiling...

Compiled successfully!

You can now view blogapp in the browser.

Local: http://localhost:3000

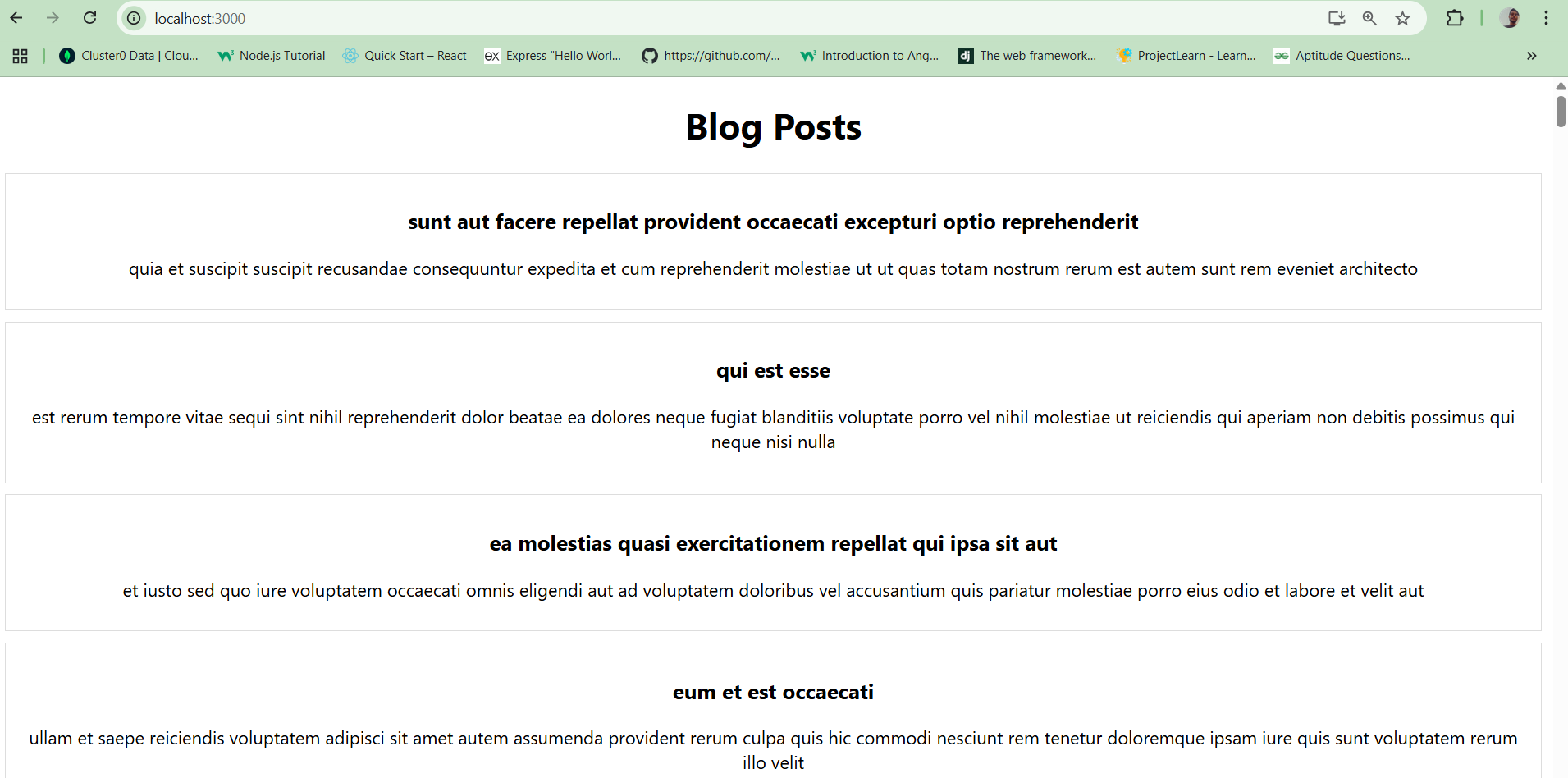
On Your Network: http://192.168.1.23:3000

Note that the development build is not optimized.

To create a production build, use npm run build.

webpack compiled successfully

**Output:**



**Note:**

"Here I pasted only the initial page; there are many more posts further down."

**5.ReactJS - HOL**

My Academy team at Cognizant want to create a dashboard containing the details of ongoing and completed cohorts. A react application is created which displays the detail of the cohorts using react component. You are assigned the task of styling these react components.

Download and build the attached react application.

\

1. Unzip the react application in a folder
2. Open command prompt and switch to the react application folder
3. Restore the node packages using the following commands



*Figure 1: Restore packages*

1. Open the application using VS Code
2. Create a new CSS Module in a file called “CohortDetails.module.css”
3. Define a css class with the name as “box” with following properties

*Width = 300px;*

*Display = inline block;*

*Overall 10px margin*

*Top and bottom padding as 10px*

*Left and right padding as 20px*

*1 px border in black color*

*A border radius of 10px*

1. Define a css style for html <dt> element using tag selector. Set the font weight to 500.
2. Open the cohort details component and import the CSS Module
3. Apply the box class to the container div
4. Define the style for <h3> element to use “green” color font when cohort status is “ongoing” and “blue” color in all other scenarios.
5. Final result should look similar to the below image



*Figure 2: Final Result*

**Source Code:**