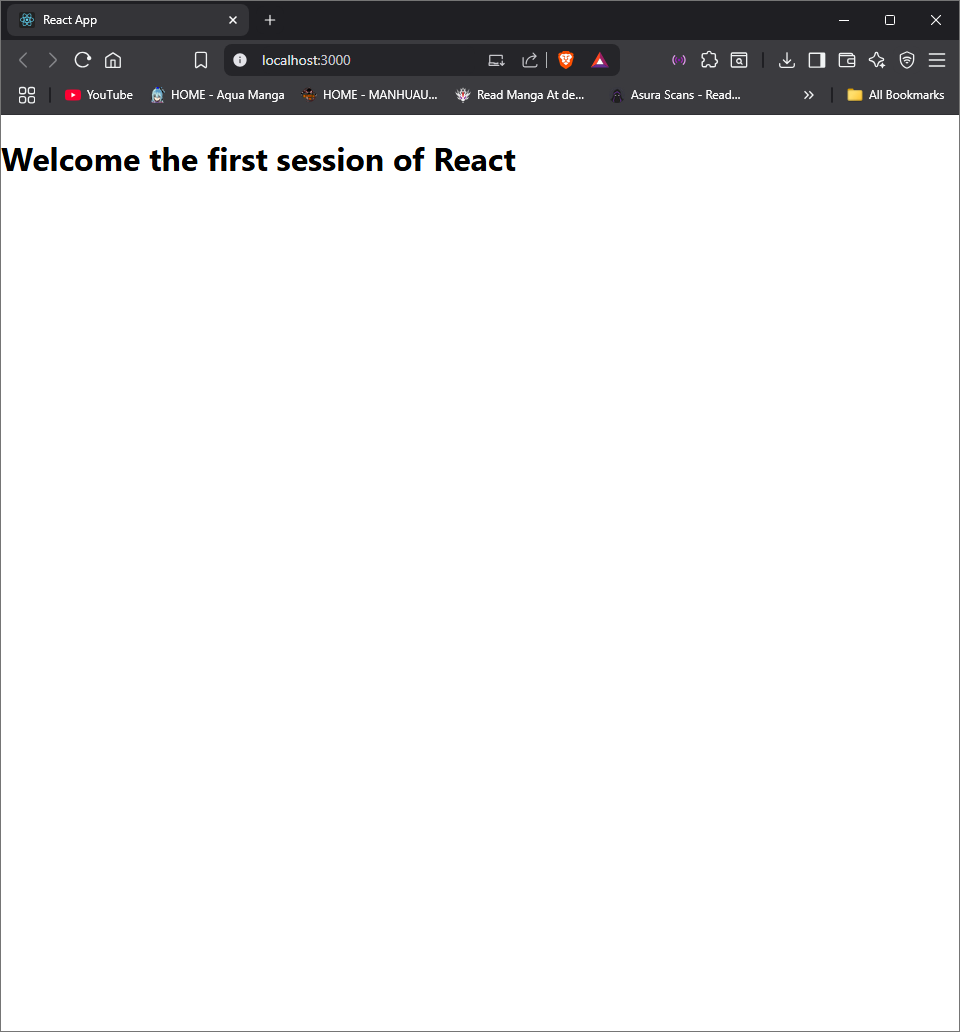
ReactJS – Hands-on Lab Exercises

# Exercise 1: 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.

Code:

function App() {  
 return (<h1>Welcome the first session of React </h1>);  
}  
  
export default App;

Expected Output:



# Exercise 2: 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.

**Code:**

Home.js:

import React, { Component } from "react";

export class Home extends Component {

render() {

return (

<div>

<h3>

Welcome to the Home Page of Student Management Portal

</h3>

</div>

);

}

}

About.js:

import React, { Component } from "react";

export class About extends Component {

render() {

return (

<div>

<h3>

Welcome to the About Page of Student Management Portal

</h3>

</div>

);

}

}

Contact.js:

import React, { Component } from "react";

export class Contact extends Component {

render() {

return (

<div>

<h3>

Welcome to the Contact Page of Student Management Portal

</h3>

</div>

);

}

}

App.js:

import logo from './logo.svg';

import './App.css';

import { Home } from './Components/Home';

import { About } from './Components/About';

import { Contact } from './Components/Contact';

function App() {

return (

<div className="container">

<Home />

<About />

<Contact />

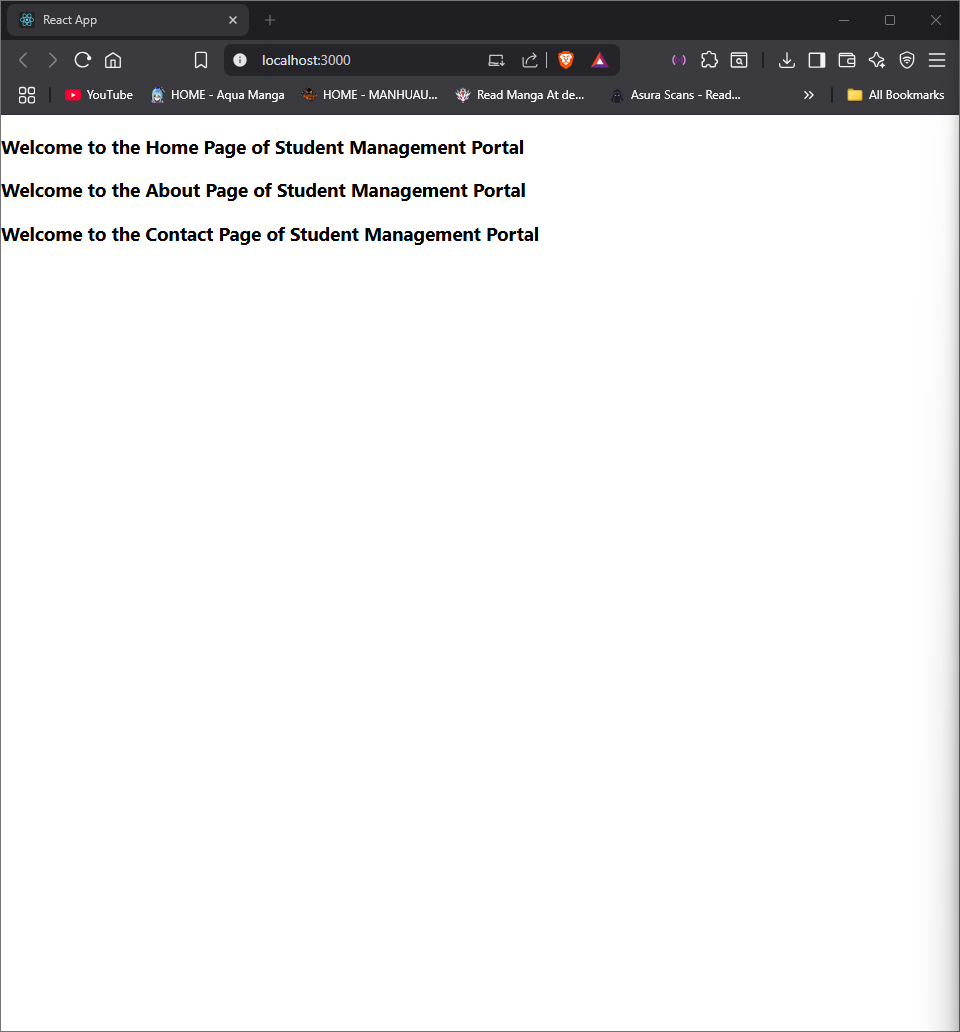
</div>

);

}

export default App;

Expected Output:



# Exercise 3: 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.

Code:

CalculateScore.js:

import '../StyleSheets/mstyle.css';

const percentToDecimal = (decimal) => {

return decimal.toFixed(2) + '%';

};

const calcScore = (total, goal) => {

return percentToDecimal(total / goal);

};

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

<div className="formatstyle">

<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>Score:</b>

<span>{calcScore(total, goal)}</span>

</div>

</div>

);

mstyle.css:

.Name {

font-weight: 300;

color: blue;

}

.School {

color: crimson;

}

.Total {

color: darkmagenta;

}

.formatstyle {

text-align: center;

font-size: large;

}

.Score {

color: forestgreen;

}

App.js:

import { CalculateScore } from '../src/Components/CalculateScore';

function App() {

return (

<div>

<CalculateScore

Name={"Steeve"}

School={"DNV Public School"}

total={284}

goal={3}

/>

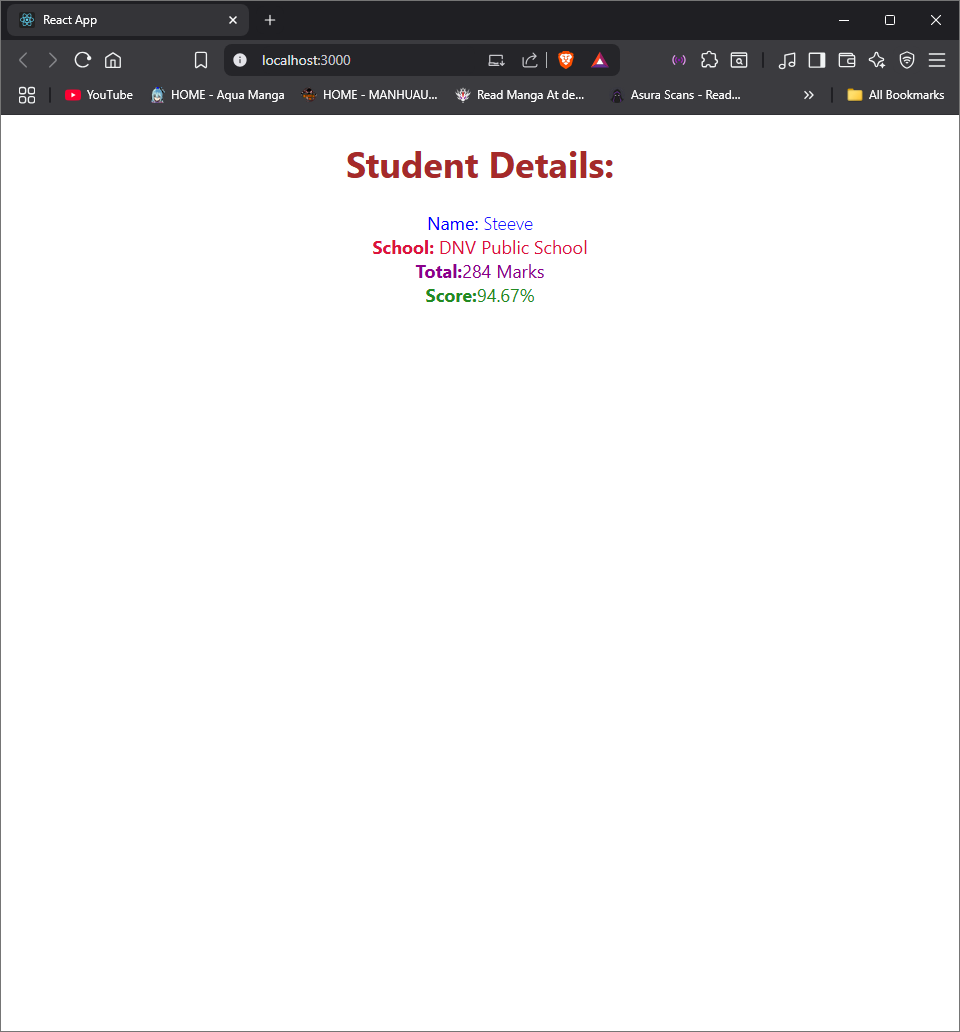
</div>

);

}

export default App;

Expected Output:



# Exercise 4: Implementing hook and hook life cycle

Code:

Post.js:

class Post {

constructor(id, title, body) {

this.id = id;

this.title = title;

this.body = body;

}

}

export default Post;

Posts.js:

import React, { Component } from 'react';

import Post from './Post';

class Posts extends Component {

constructor(props) {

super(props);

this.state = {

posts: [],

error: null

};

}

loadPosts = () => {

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

.then(response => response.json())

.then(data => {

const postsList = data.slice(0, 5).map(post =>

new Post(post.id, post.title, post.body)

);

this.setState({ posts: postsList });

})

.catch(error => {

this.setState({ error });

});

};

componentDidMount() {

this.loadPosts();

}

componentDidCatch(error, info) {

alert("An error occurred while loading posts.");

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

}

render() {

const { posts } = this.state;

return (

<div>

<h2>Blog Posts</h2>

{posts.map(post => (

<div key={post.id}>

<h3>{post.title}</h3>

<p>{post.body}</p>

<hr />

</div>

))}

</div>

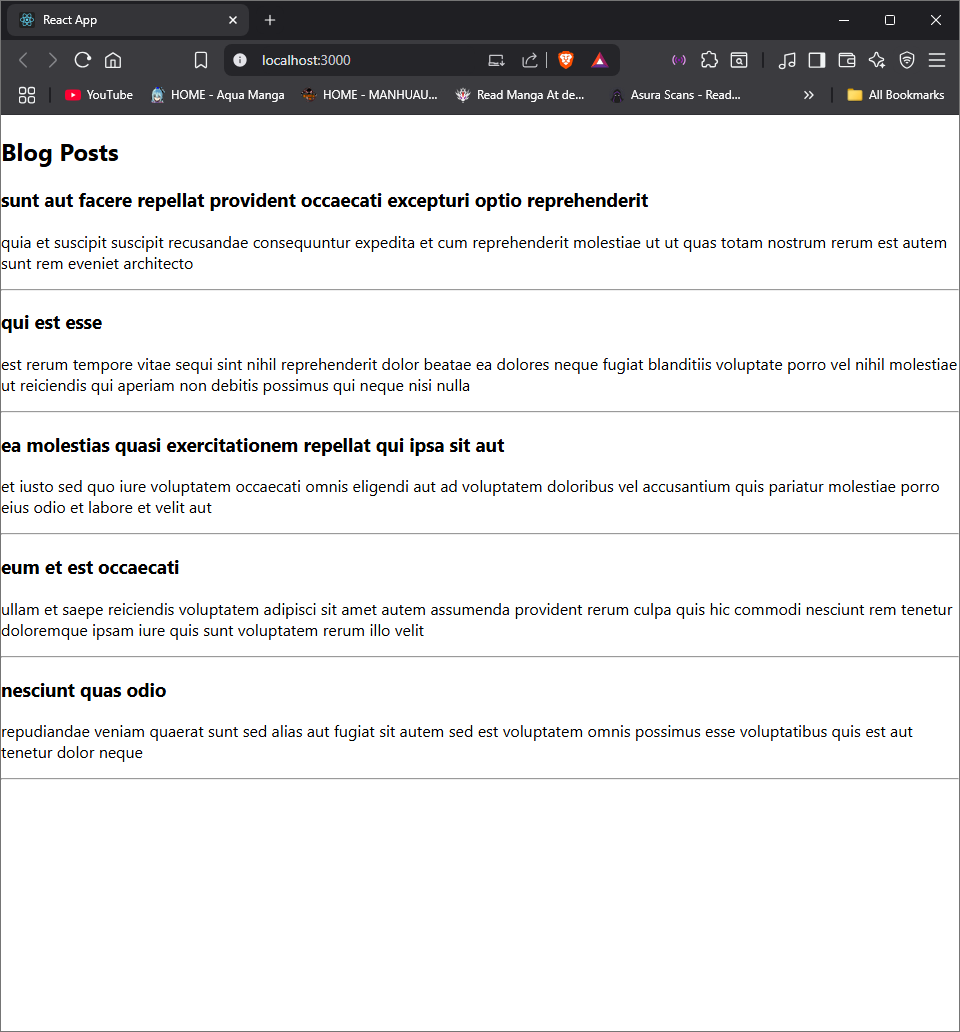
);

}

}

export default Posts;

Expected Output:



# Exercise 5: My Academy team at Cognizant wants to create a dashboard containing the details of ongoing and completed cohorts. A React application is created which displays the details of the cohorts using a React component. You are assigned the task of styling these React components.

Code:

CohorDetails.module.css:

.box {

width: 300px;

display: inline-block;

margin: 10px;

padding: 10px 20px;

border: 1px solid black;

border-radius: 10px;

}

dt {

font-weight: 500;

}

CohorDetails.js:

import React from 'react';

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

const CohortDetails = ({ cohort }) => {

const headingStyle = {

color: cohort.status === "ongoing" ? "green" : "blue"

};

return (

<div className={styles.box}>

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

<dl>

<dt>Batch ID:</dt>

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

<dt>Status:</dt>

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

<dt>Technology:</dt>

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

</dl>

</div>

);

};

export default CohortDetails;

App.js:

import React from 'react';

import CohortDetails from './CohortDetails';

function App() {

const cohorts = [

{

name: "React Developer Program",

batchId: "CDN-001",

status: "ongoing",

technology: "ReactJS"

},

{

name: "Java Developer Bootcamp",

batchId: "CDN-002",

status: "completed",

technology: "Java"

}

];

return (

<div>

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

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

))}

</div>

);

}

export default App;

Expected Output:

