

WEEK 6 ASSIGNMENT : REACT APP CREATION

1.Creating first react app

Code:

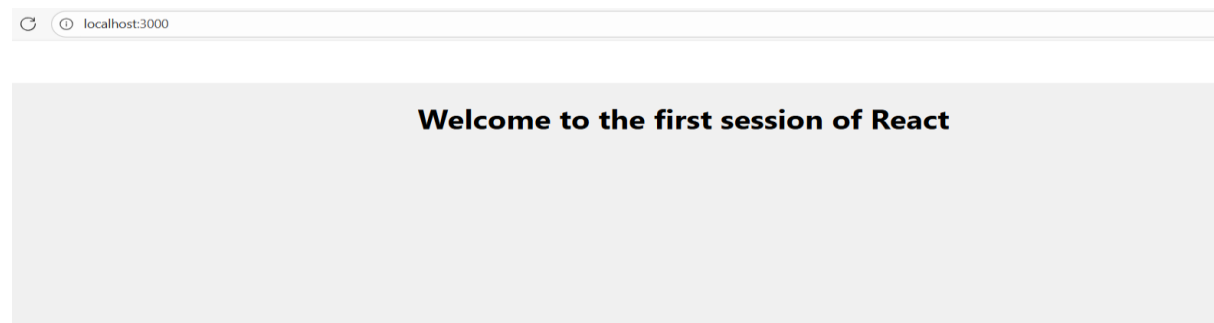
App.js

```
import React from 'react';
import './App.css'; // make sure this is imported
function App() {
  return (
    <div className="center">
      <h1>Welcome to the first session of React</h1>
    </div>
  );
}
export default App;
```

App.css

```
.center {
  display: flex;
  justify-content: center;
  align-items: flex-start;
  margin-top: 50px;
  height: 100vh;
  background-color: #f0f0f0;
}
```

OUTPUT:



2. Student Management Portal – React Application

Code:

Home.js

```
import React from 'react';

function Home() {

  return (

    <div>

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

    </div>

  );

}

export default Home;
```

About.js

```
import React from 'react';

function About() {

  return (

    <div>

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

    </div>

  );

}

export default About;
```

Contact.js

```
import React from 'react';

function Contact() {

  return (

    <div>

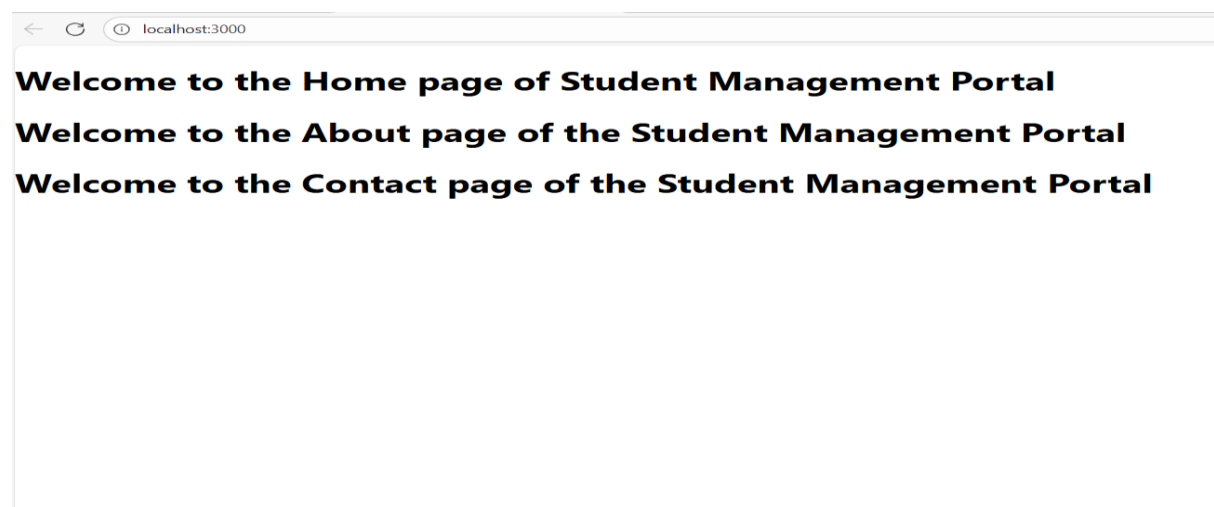
      <h1>Welcome to the Contact page of the Student Management Portal</h1>
```

```
    </div>
  );
}
export default Contact;
```

App.js

```
import React from 'react';
import Home from './Home';
import About from './About';
import Contact from './Contact';
function App() {
  return (
    <div>
      <Home />
      <About />
      <Contact />
    </div>
  );
}
export default App;
```

OUTPUT:



3.Calculating Average Score-React Application

Code:**CalculateScore.js**

```
import React, { useState } from 'react';
import '../Stylesheets/mystyle.css';

function CalculateScore() {
  const [name, setName] = useState("");
  const [school, setSchool] = useState("");
  const [total, setTotal] = useState("");
  const [goal, setGoal] = useState("");
  const [average, setAverage] = useState(null);

  const handleCalculate = () => {
    const totalNum = parseFloat(total);
    const goalNum = parseFloat(goal);

    if (!name || !school || isNaN(totalNum) || isNaN(goalNum) || totalNum <= 0) {
      alert('Please enter valid inputs');
      return;
    }

    // average = (goal / total) * 100
    const avgScore = (goalNum / totalNum) * 100;
    setAverage(avgScore.toFixed(2));
  };

  return (
    <div className="container">
      <h2>Student Score Calculator</h2>
```

```
<div className="form-group">
  <label>Name:</label>
  <input
    type="text"
    value={name}
    onChange={(e) => setName(e.target.value)}
    placeholder="Enter student name"
  />
</div>
```

```
<div className="form-group">
  <label>School:</label>
  <input
    type="text"
    value={school}
    onChange={(e) => setSchool(e.target.value)}
    placeholder="Enter school name"
  />
</div>
```

```
<div className="form-group">
  <label>Total Marks:</label>
  <input
    type="number"
    value={total}
    onChange={(e) => setTotal(e.target.value)}
    placeholder="Enter total marks"
  />
</div>
```

```
<div className="form-group">
  <label>Goal Marks:</label>
  <input
    type="number"
    value={goal}
    onChange={(e) => setGoal(e.target.value)}
    placeholder="Enter goal marks"
  />
</div>
```

```
<button onClick={handleCalculate}>Calculate Average Score</button>
```

```
{average !== null && (
  <div className="result">
    <h3>Result</h3>
    <p>
      Student <strong>{name}</strong> from <strong>{school}</strong> has
      an average score of <strong>{average}</strong>%.
    </p>
  </div>
)}
</div>
);
}
```

```
export default CalculateScore;
```

```
mystyle.css
```

```
.container {
```

```
max-width: 400px;
margin: 30px auto;
padding: 20px;
border: 2px solid #4caf50;
border-radius: 8px;
background-color: #f9f9f9;
font-family: Arial, sans-serif;
}
```

```
h2 {
  text-align: center;
  color: #4caf50;
}
```

```
.form-group {
  margin-bottom: 15px;
}
```

```
label {
  display: block;
  margin-bottom: 6px;
  font-weight: bold;
}
```

```
input[type="text"],
input[type="number"] {
  width: 100%;
  padding: 8px;
  box-sizing: border-box;
```

```
}
```

```
button {  
  background-color: #4caf50;  
  color: white;  
  padding: 10px 15px;  
  border: none;  
  border-radius: 4px;  
  cursor: pointer;  
  width: 100%;  
  font-size: 16px;  
}
```

```
button:hover {  
  background-color: #45a049;  
}
```

```
.result {  
  margin-top: 20px;  
  padding: 15px;  
  background-color: #dff0d8;  
  border: 1px solid #3c763d;  
  border-radius: 5px;  
  color: #3c763d;  
  font-weight: bold;  
}
```

App.js


```
import React from 'react';

import CalculateScore from './Components/CalculateScore';

function App() {

  return (

    <div>

      <CalculateScore />

    </div>

  );

}

export default App;
```

OUTPUT:

Student Score Calculator

Name:

School:

Total Marks:

Goal Marks:

Calculate Average Score

Result

Student **Shuruthika** from **C.E.O.A matric hr sec school** has an average score of **101.12%**.

4.BLOG APP- React Application

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  
    };  
  }  
  
  // Method to fetch posts using Fetch API  
  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 => {  
    // Convert JSON data to array of Post instances  
    const posts = data.map(p => new Post(p.id, p.title, p.body));  
    this.setState({ posts });  
  })  
  .catch(error => {  
    this.setState({ error });  
  });  
}
```

// Called automatically after component is mounted

```
componentDidMount() {  
  this.loadPosts();  
}
```

// Catch any error in rendering

```
componentDidCatch(error, info) {  
  alert('An error occurred: ' + error.message);  
}
```

```
render() {  
  const { posts, error } = this.state;
```

```
  if (error) {
```

```
    return <h2>Error loading posts</h2>;
  }

  return (
    <div>
      <h1>Blog Posts</h1>
      {posts.map(post => (
        <div key={post.id} style={{ border: '1px solid #ccc', marginBottom: '10px', padding:
'10px' }}>
          <h2>{post.title}</h2>
          <p>{post.body}</p>
        </div>
      ))}
    </div>
  );
}
}
```

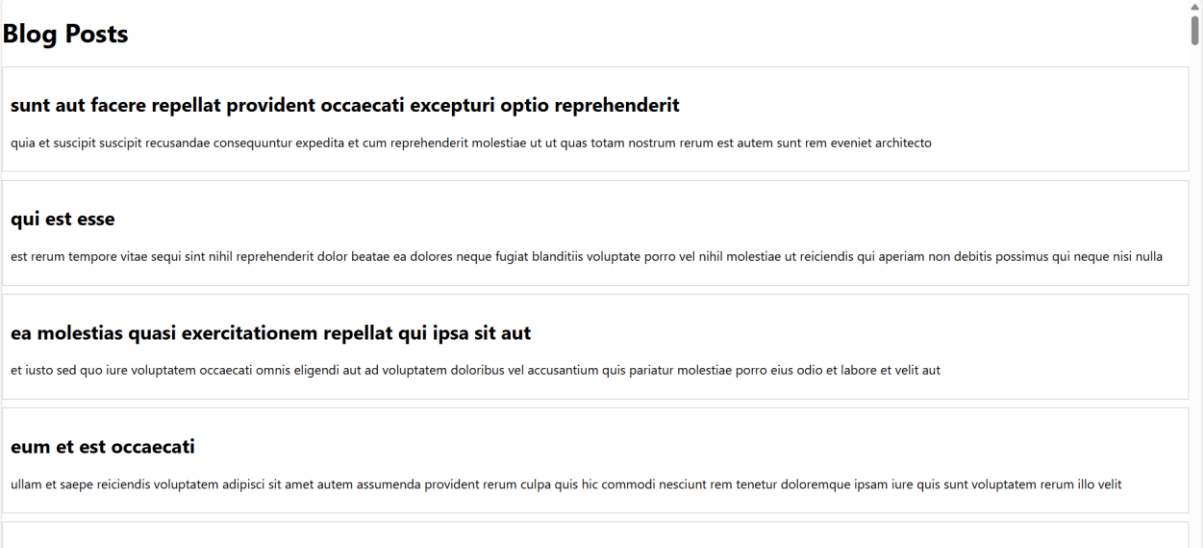
```
export default Posts;
```

App.js

```
import React from 'react';
import Posts from './Posts';
function App() {
  return (
    <div className="App">
      <Posts />
    </div>
  );
}
```

export default App;

OUTPUT:



5.Cohort Dashboard with Conditional Styling-React Application

Code:

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;  
}
```

CohortDetails.js

```
import React from 'react';

import styles from './CohortDetails.module.css'; // Import the CSS module

function CohortDetails({ cohort }) {

  const { name, trainer, status, startDate, endDate } = cohort;

  // Determine the color for the <h3> based on cohort status
  const titleColor = status.toLowerCase() === 'ongoing' ? 'green' : 'blue';

  return (
    <div className={styles.box}>
      <h3 style={{ color: titleColor }}>{name}</h3>
      <dl>
        <dt>Trainer:</dt>
        <dd>{trainer}</dd>
        <dt>Status:</dt>
        <dd>{status}</dd>
        <dt>Start Date:</dt>
        <dd>{startDate}</dd>
        <dt>End Date:</dt>
        <dd>{endDate}</dd>
      </dl>
    </div>
  );
}

export default CohortDetails;
```

App.js

```
import React from 'react';
import CohortDetails from './components/CohortDetails';

function App() {
  const cohorts = [
    {
      name: 'React Bootcamp',
      trainer: 'John Doe',
      status: 'ongoing',
      startDate: '2025-07-01',
      endDate: '2025-08-01',
    },
    {
      name: 'Node.js Mastery',
      trainer: 'Jane Smith',
      status: 'completed',
      startDate: '2025-05-15',
      endDate: '2025-06-15',
    },
  ];
  return (
    <div>
      {cohorts.map((cohort, index) => (
        <CohortDetails key={index} cohort={cohort} />
      ))}
    </div>
  );
}
```

export default App;

OUTPUT:

