Week 6

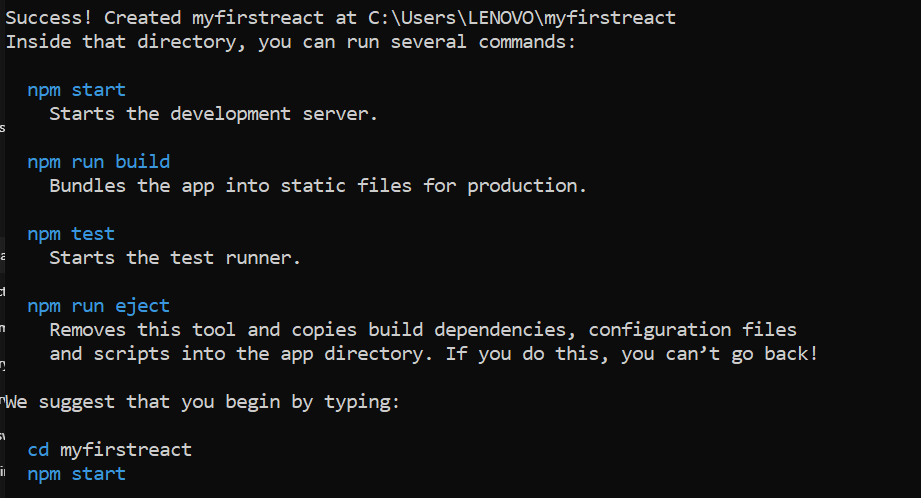
Skill name : React

Hands-on 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.

#### Step 1:- Install create-react-app globally

Open **Command Prompt** or **Terminal** and run:

npm install -g create-react-app

npx create-react-app myfirstreact

Step 2:- Navigate into the project folder

//App.js

import React from 'react';

function App() {

return (

<div>

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

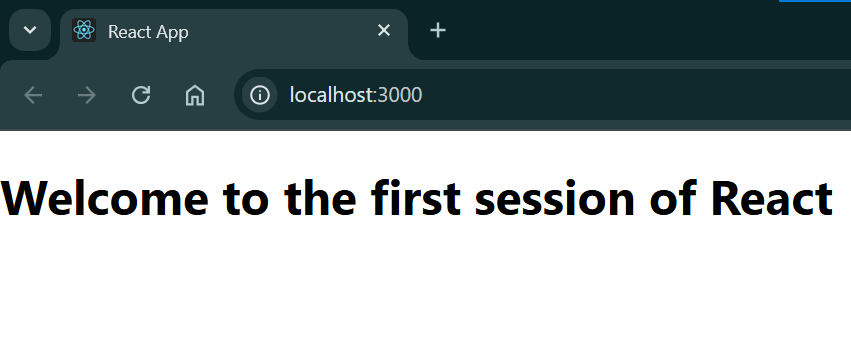
</div>

);

}

export default App;

Output:-

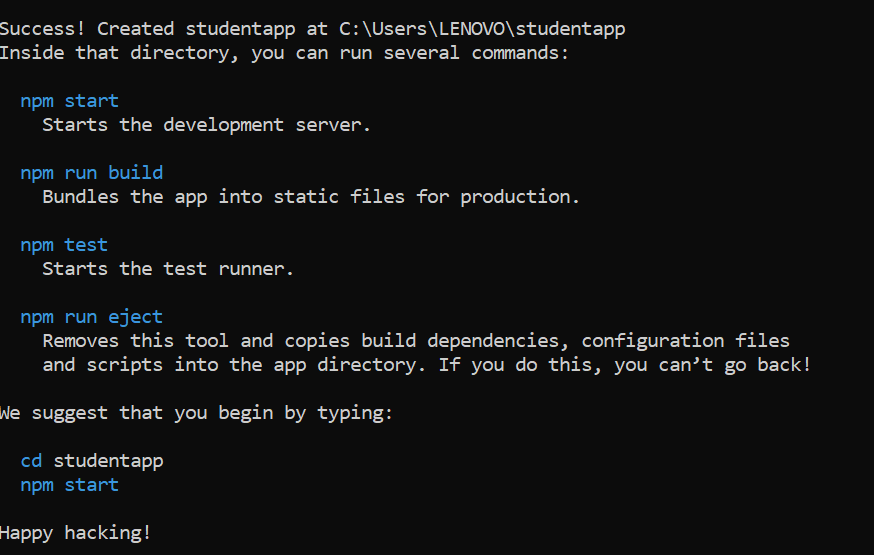


Hands-on 2:

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.

Step 1:-

npx create-react-app StudentApp



Step 2:-

//Home.js

import React from 'react';

function Home() {

return (

<div>

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

</div>

);

}

export default Home;

//About.js

import React from 'react';

function About() {

return (

<div>

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

</div>

);

}

export default About;

//Contact.js

import React from 'react';

function Contact() {

return (

<div>

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

</div>

);

}

export default Contact;

//App.js

import React from 'react';

import Home from './Components/Home';

import About from './Components/About';

import Contact from './Components/Contact';

function App() {

return (

<div className="App">

<Home />

<About />

<Contact />

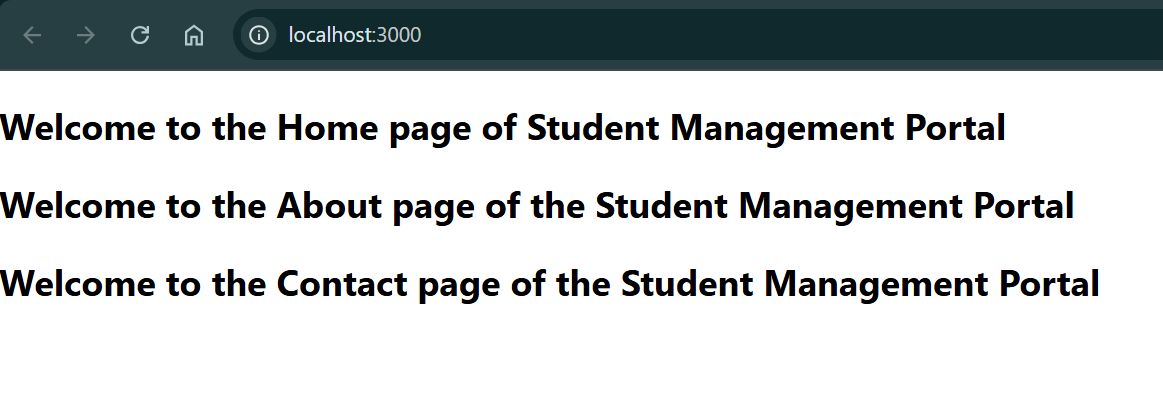
</div>

);

}

export default App;

Output:-

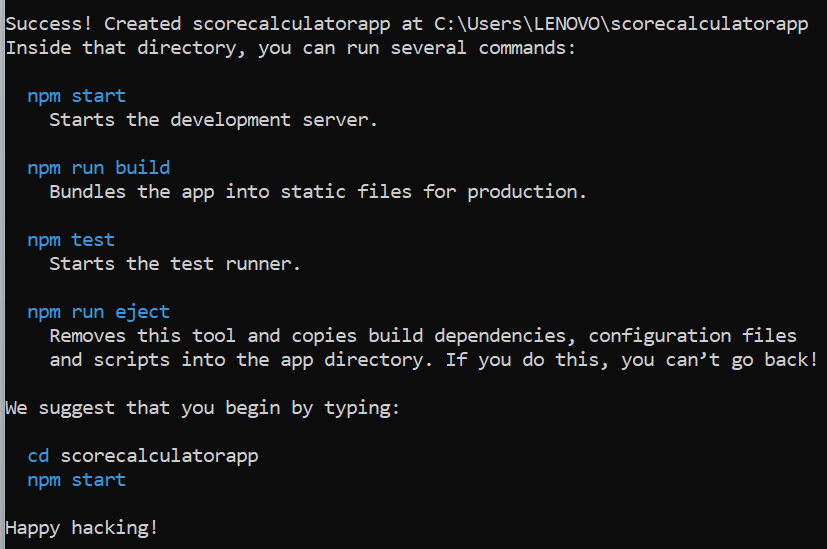


Hands-on 3:-

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.

Step 1:

npx create-react-app scorecalculatorapp

//src/Components/CalculateScore.js

import React from 'react';

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

function CalculateScore() {

const name = "Ananya";

const school = "Green Valley High School";

const total = 450;

const goal = 500;

const average = (total / goal) \* 100;

return (

<div className="score-container">

<h2>Student Score Report</h2>

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

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

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

<p><strong>Goal Marks:</strong> {goal}</p>

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

</div>

);

}

export default CalculateScore;

//src/Stylesheets/mystyle.css

.score-container {

margin: 30px auto;

padding: 20px;

border: 2px solid #4caf50;

border-radius: 10px;

max-width: 500px;

background-color: #f9f9f9;

font-family: Arial, sans-serif;

box-shadow: 2px 2px 10px rgba(0,0,0,0.1);

}

.score-container h2 {

color: #2e7d32;

text-align: center;

}

.score-container p {

font-size: 16px;

margin: 8px 0;

}

//App.js

import React from 'react';

import './App.css';

import CalculateScore from './Components/CalculateScore';

function App() {

return (

<div className="App">

<CalculateScore />

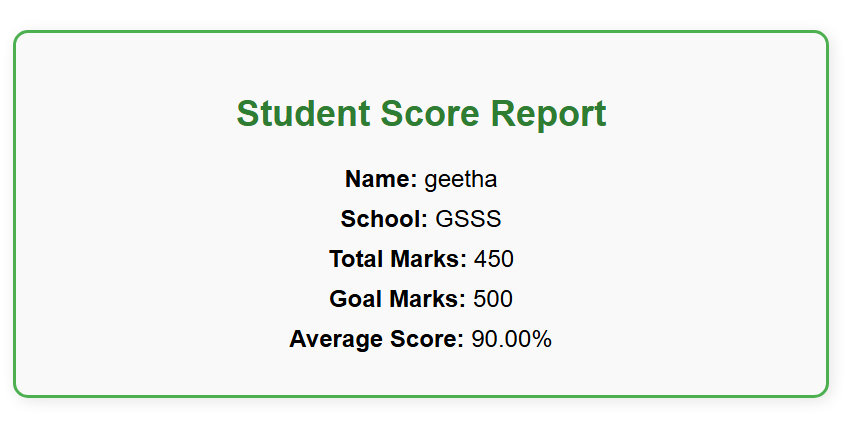
</div>

);

}

export default App;

Output:-



**Exercise 4: ReactJS Lifecycle Methods - blogapp**

This exercise demonstrates React component lifecycle methods using a blog app that fetches posts and handles errors.

**Step 1: Create React App**

npx create-react-app blogapp  
cd blogapp  
code .

**Step 2: Create Post.js**

// src/Post.js  
import React from 'react';  
  
class Post extends React.Component {  
 render() {  
 const { title, body } = this.props;  
 return (  
 <div>  
 <h2>{title}</h2>  
 <p>{body}</p>  
 </div>  
 );  
 }  
}  
  
export default Post;

**Step 3: Create Posts.js**

// src/Posts.js  
import React from 'react';  
import Post from './Post';  
  
class Posts extends React.Component {  
 constructor(props) {  
 super(props);  
 this.state = {  
 posts: [],  
 hasError: false  
 };  
 }  
  
 loadPosts = () => {  
 fetch('https://jsonplaceholder.typicode.com/posts')  
 .then(response => response.json())  
 .then(data => this.setState({ posts: data.slice(0, 5) }))  
 .catch(error => {  
 console.error('Error fetching posts:', error);  
 this.setState({ hasError: true });  
 });  
 };  
  
 componentDidMount() {  
 this.loadPosts();  
 }  
  
 componentDidCatch(error, info) {  
 alert("An error occurred: " + error.toString());  
 console.error("Error boundary caught:", error, info);  
 }  
  
 render() {  
 if (this.state.hasError) {  
 return <h2>Something went wrong while loading posts.</h2>;  
 }  
  
 return (  
 <div>  
 {this.state.posts.map(post => (  
 <Post key={post.id} title={post.title} body={post.body} />  
 ))}  
 </div>  
 );  
 }  
}  
  
export default Posts;

**Step 4: Modify App.js**

// src/App.js  
import React from 'react';  
import Posts from './Posts';  
  
function App() {  
 return (  
 <div className="App">  
 <h1>Blog Posts</h1>  
 <Posts />  
 </div>  
 );  
}  
  
export default App;

Run the application using `npm start`. The posts will be displayed using componentDidMount and errors handled using componentDidCatch.

**Exercise 5: Styling React Components with CSS Modules**

This exercise focuses on styling React components using CSS Modules and inline styles.

**Step 1: Create CSS Module**

/\* src/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;  
}

**Step 2: Create CohortDetails.js**

// src/CohortDetails.js  
import React from 'react';  
import styles from './CohortDetails.module.css';  
  
function CohortDetails({ name, status, startDate, endDate }) {  
 const titleStyle = {  
 color: status === 'ongoing' ? 'green' : 'blue'  
 };  
  
 return (  
 <div className={styles.box}>  
 <h3 style={titleStyle}>{name}</h3>  
 <dl>  
 <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;

**Step 3: Modify App.js**

// src/App.js  
import React from 'react';  
import CohortDetails from './CohortDetails';  
  
function App() {  
 return (  
 <div>  
 <CohortDetails name="React Bootcamp" status="ongoing" startDate="01 July 2025" endDate="15 August 2025" />  
 <CohortDetails name="Node.js Track" status="completed" startDate="01 May 2025" endDate="15 June 2025" />  
 </div>  
 );  
}  
  
export default App;

**Exercise 5: Styling React Components with CSS Modules**

This exercise focuses on styling React components using CSS Modules and inline styles.

**Step 1: Create CSS Module**

/\* src/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;  
}

**Step 2: Create CohortDetails.js**

// src/CohortDetails.js  
import React from 'react';  
import styles from './CohortDetails.module.css';  
  
function CohortDetails({ name, status, startDate, endDate }) {  
 const titleStyle = {  
 color: status === 'ongoing' ? 'green' : 'blue'  
 };  
  
 return (  
 <div className={styles.box}>  
 <h3 style={titleStyle}>{name}</h3>  
 <dl>  
 <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;

**Step 3: Modify App.js**

// src/App.js  
import React from 'react';  
import CohortDetails from './CohortDetails';  
  
function App() {  
 return (  
 <div>  
 <CohortDetails name="React Bootcamp" status="ongoing" startDate="01 July 2025" endDate="15 August 2025" />  
 <CohortDetails name="Node.js Track" status="completed" startDate="01 May 2025" endDate="15 June 2025" />  
 </div>  
 );  
}  
  
export default App;

The component will display green or blue text based on the cohort status. Boxes are styled using the CSS Module.