**1. Getting Started with React (myfirstreact)**

## Objectives

* • Define SPA and its benefits
* • Define React and identify its working
* • Identify the differences between SPA and MPA
* • Explain Pros & Cons of Single-Page Application
* • Explain about React
* • Define virtual DOM
* • Explain Features of React

## In this hands-on lab, you will learn how to:

* • Set up a React environment
* • Use create-react-app

## Steps

Install Node.js and npm from https://nodejs.org/en/download/

Install Create React App:  
  
npm install -g create-react-app

Create app:  
  
npx create-react-app myfirstreact

Navigate to project:  
  
cd myfirstreact

Open project in VS Code:  
  
code .

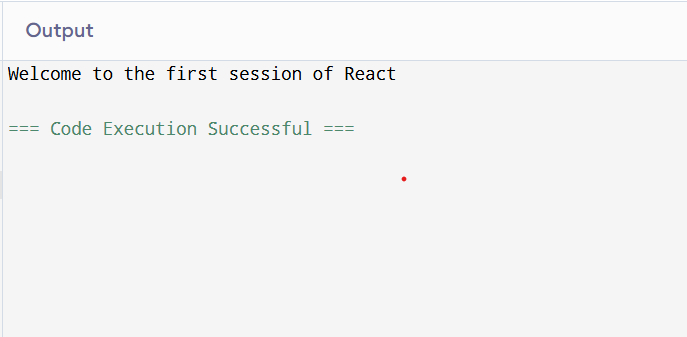
Open App.js under src folder, remove existing content and replace with:

import React from 'react';  
  
function App() {  
 return (  
 <div>  
 <h1>Welcome to the first session of React</h1>  
 </div>  
 );  
}  
  
export default App;

Run the app:  
  
npm start

Open browser and go to: http://localhost:3000

Output:



# 2. Multiple Components in StudentApp

## Objectives

* • Explain React components
* • Identify the differences between components and JavaScript functions
* • Identify the types of components
* • Explain class component
* • Explain function component
* • Define component constructor
* • Define render() function

## In this hands-on lab, you will learn how to:

* • Create a class component
* • Create multiple components
* • Render a component

## Steps

Create project:  
  
npx create-react-app StudentApp

Create 'Components' folder under src.

Add 'Home.js':

import React from 'react';  
  
function Home() {  
 return <h2>Welcome to the Home page of Student Management Portal</h2>;  
}  
  
export default Home;

Add 'About.js':

import React from 'react';  
  
function About() {  
 return <h2>Welcome to the About page of the Student Management Portal</h2>;  
}  
  
export default About;

Add 'Contact.js':

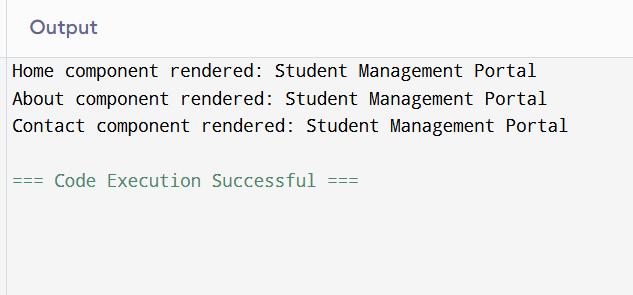
import React from 'react';  
  
function Contact() {  
 return <h2>Welcome to the Contact page of the Student Management Portal</h2>;  
}  
  
export default Contact;

Edit App.js to include:

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

Run the app:  
  
npm start  
Go to http://localhost:3000

Output:



# 3. Functional Component with Styling (scorecalculatorapp)

## Objectives

* • Explain React components
* • Differentiate components and JS functions
* • Identify component types
* • Explain function component

## In this hands-on lab, you will learn how to:

* • Create a function component
* • Apply style to components
* • Render a component

## Steps

Create project:  
  
npx create-react-app scorecalculatorapp

Under src, create folder 'Components' and file 'CalculateScore.js':

import React from 'react';  
import '../Stylesheets/mystyle.css';  
  
function CalculateScore() {  
 const name = 'John';  
 const school = 'ABC High School';  
 const total = 450;  
 const goal = 500;  
 const average = (total / goal) \* 100;  
  
 return (  
 <div className='score-box'>  
 <h2>Score Calculator</h2>  
 <p>Name: {name}</p>  
 <p>School: {school}</p>  
 <p>Average Score: {average.toFixed(2)}%</p>  
 </div>  
 );  
}  
  
export default CalculateScore;

Create folder 'Stylesheets' and file 'mystyle.css':

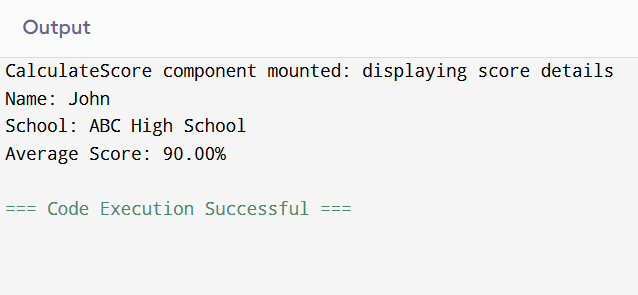
.score-box {  
 width: 300px;  
 margin: 10px;  
 padding: 10px 20px;  
 border: 1px solid black;  
 border-radius: 10px;  
 display: inline-block;  
}

Edit App.js:

import React from 'react';  
import CalculateScore from './Components/CalculateScore';  
  
function App() {  
 return (  
 <div>  
 <CalculateScore />  
 </div>  
 );  
}  
  
export default App;

Run the app:  
  
npm start  
Go to http://localhost:3000

Output:



# 4. Component Lifecycle Methods (blogapp)

## Objectives

* • Understand benefits of component life cycle
* • Identify lifecycle hook methods
* • List rendering steps

## In this hands-on lab, you will learn how to:

* • Implement componentDidMount()
* • Implement componentDidCatch()

## Steps

Create project:  
  
npx create-react-app blogapp

Open project in VS Code

Create 'Post.js' and 'Posts.js' under src folder

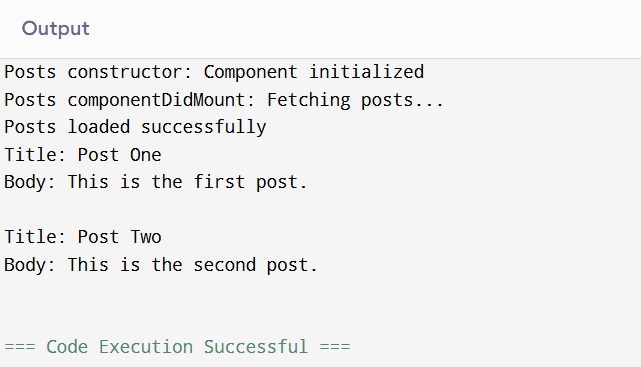
Edit Posts.js:

import React, { Component } from 'react';  
  
class Posts extends Component {  
 constructor(props) {  
 super(props);  
 this.state = { posts: [], hasError: false };  
 }  
  
 componentDidMount() {  
 this.loadPosts();  
 }  
  
 loadPosts = async () => {  
 try {  
 const response = await fetch('https://jsonplaceholder.typicode.com/posts');  
 const data = await response.json();  
 this.setState({ posts: data });  
 } catch (error) {  
 this.setState({ hasError: true });  
 }  
 };  
  
 componentDidCatch(error, info) {  
 alert('Error occurred: ' + error);  
 }  
  
 render() {  
 if (this.state.hasError) return <h2>Something went wrong.</h2>;  
  
 return (  
 <div>  
 {this.state.posts.slice(0, 5).map(post => (  
 <div key={post.id}>  
 <h3>{post.title}</h3>  
 <p>{post.body}</p>  
 </div>  
 ))}  
 </div>  
 );  
 }  
}  
  
export default Posts;

Add Posts component to App.js

Run the app:  
  
npm start  
Open http://localhost:3000

Output:



# 5. Styling with CSS Modules (Cohort Dashboard)

## Objectives

* • Understand need for styling
* • Work with CSS Modules and inline styles

## In this hands-on lab, you will learn how to:

* • Style components
* • Use CSS Module
* • Apply className and style properties

## Steps

Unzip provided React app and open in VS Code

Install dependencies:  
  
npm install

Create 'CohortDetails.module.css' with:

.box {  
 width: 300px;  
 display: inline-block;  
 margin: 10px;  
 padding: 10px 20px;  
 border: 1px solid black;  
 border-radius: 10px;  
}  
  
dt {  
 font-weight: 500;  
}

Import CSS Module in component:  
  
import styles from './CohortDetails.module.css';

Apply class:  
  
<div className={styles.box}>...</div>

Set h3 color to green inline or via CSS

Run:  
  
npm start  
Go to http://localhost:3000

Output:

