**REACT JS**

**WEEK 7**

## **Objectives**

* Define JSX
* Explain about ECMA Script
* Explain React.createElement()
* Explain how to create React nodes with JSX
* Define how to render JSX to DOM
* Explain how to use JavaScript expressions in JSX
* Explain how to use inline CSS in JSX

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

* Use JSX syntax in React applications
* Use inline CSS in JSX

## **Prerequisites**

The following is required to complete this hands-on lab:

* Node.js
* NPM
* Visual Studio Code

## **Notes**

Estimated time to complete this lab: **60 minutes.**

Create a React Application named “officespacerentalapp” which uses React JSX to create elements, attributes and renders DOM to display the page.

Create an element to display the heading of the page.

Attribute to display the image of the office space

Create an object of office to display the details like Name, Rent and Address.

Create a list of Object and loop through the office space item to display more data.

To apply Css, Display the color of the Rent in Red if it’s below 60000 and in Green if it’s above 60000.

Output:



**Hint:**





Index.js:

import React from 'react';import ReactDOM from 'react-dom/client';import './index.css';import App from './App';import reportWebVitals from './reportWebVitals';const root = ReactDOM.createRoot(document.getElementById('root'));root.render( <React.StrictMode> <App /> </React.StrictMode>);// If you want to start measuring performance in your app, pass a function// to log results (for example: reportWebVitals(console.log))// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitalsreportWebVitals();

Index.css:

body { margin: 0; font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen', 'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue', sans-serif; -webkit-font-smoothing: antialiased; -moz-osx-font-smoothing: grayscale;}code { font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New', monospace;}

App.js

import React from 'react';import './App.css';import AmazonImg from './Amazon.jpg';import GoogleImg from './Google.jpg';import FlipkartImg from './Flipkart.jpg';function App() { const element = "Office Space"; const officeList = [ { Name: "Amazon", Rent: 100000, Address: "America", image: AmazonImg }, { Name: "Google", Rent: 150000, Address: "New Zealand", image: GoogleImg }, { Name: "Flipkart", Rent: 200000, Address: "New York", image: FlipkartImg } ]; return ( <div className="App"> <h1>{element} in Several Places with Best Prices</h1> <div className="card-container"> {officeList.map((item, index) => { let rentColor = item.Rent <= 60000 ? "textRed" : "textGreen"; return ( <div key={index} className="card"> <img src={item.image} alt={item.Name} /> <h1>Name: {item.Name}</h1> <h3 className={rentColor}>Rent: Rs. {item.Rent}</h3> <h3>Address: {item.Address}</h3> </div> ); })} </div> </div> );}export default App;

Report Web Vitals:

const reportWebVitals = onPerfEntry => { if (onPerfEntry && onPerfEntry instanceof Function) { import('web-vitals').then(({ getCLS, getFID, getFCP, getLCP, getTTFB }) => { getCLS(onPerfEntry); getFID(onPerfEntry); getFCP(onPerfEntry); getLCP(onPerfEntry); getTTFB(onPerfEntry); }); }};export default reportWebVitals;

Package.json:

{ "name": "officespacerentalapp", "version": "0.1.0", "private": true, "dependencies": { "@testing-library/dom": "^10.4.0", "@testing-library/jest-dom": "^6.6.3", "@testing-library/react": "^16.3.0", "@testing-library/user-event": "^13.5.0", "react": "^19.1.0", "react-dom": "^19.1.0", "react-scripts": "5.0.1", "web-vitals": "^2.1.4" }, "scripts": { "start": "react-scripts start", "build": "react-scripts build", "test": "react-scripts test", "eject": "react-scripts eject" }, "eslintConfig": { "extends": [ "react-app", "react-app/jest" ] }, "browserslist": { "production": [ ">0.2%", "not dead", "not op\_mini all" ], "development": [ "last 1 chrome version", "last 1 firefox version", "last 1 safari version" ] }}

Output:



