React Lab – JSX and Inline CSS

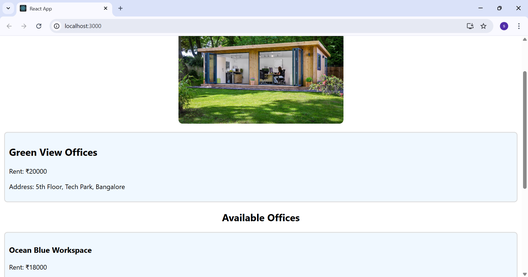
# Objectives

• 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

# Hands-On Lab Summary

This React application named “officespacerentalapp” demonstrates the use of JSX, inline CSS,   
JavaScript expressions, and mapping through an object list to display office space data. It includes a page heading, image rendering,   
a single office object display, and a looped list of office items, all styled using inline CSS.

# Output Screenshot



# React Code: App.js

import React from 'react';  
  
const officeStyle = {  
 border: '2px solid #ddd',  
 padding: '10px',  
 margin: '10px',  
 borderRadius: '8px',  
 backgroundColor: '#f0f8ff'  
};  
  
const App = () => {  
 const headingStyle = {  
 color: '#2a4d69',  
 textAlign: 'center'  
 };  
  
 const imageUrl = 'https://www.sunshineslate.com/wp-content/uploads/2020/11/Garden-Offices.jpg';  
  
 const office = {  
 name: 'Green View Offices',  
 rent: 20000,  
 address: '5th Floor, Tech Park, Bangalore'  
 };  
  
 const officeList = [  
 {  
 name: 'Ocean Blue Workspace',  
 rent: 18000,  
 address: 'Koramangala, Bangalore'  
 },  
 {  
 name: 'Skyline Corporate',  
 rent: 25000,  
 address: 'Hi-Tech City, Hyderabad'  
 },  
 {  
 name: 'UrbanHub',  
 rent: 22000,  
 address: 'Indiranagar, Bangalore'  
 }  
 ];  
  
 return (  
 <div>  
 <h1 style={headingStyle}>🏢 Office Space Rental App</h1>  
  
 <img  
 src={imageUrl}  
 alt="Office Space"  
 style={{  
 display: 'block',  
 margin: '20px auto',  
 width: '400px',  
 height: 'auto',  
 borderRadius: '10px'  
 }}  
 />  
  
 <div style={officeStyle}>  
 <h2>{office.name}</h2>  
 <p>Rent: ₹{office.rent}</p>  
 <p>Address: {office.address}</p>  
 </div>  
  
 <h2 style={{ textAlign: 'center' }}>Available Offices</h2>  
 {officeList.map((item, index) => (  
 <div key={index} style={officeStyle}>  
 <h3>{item.name}</h3>  
 <p>Rent: ₹{item.rent}</p>  
 <p>Address: {item.address}</p>  
 </div>  
 ))}  
 </div>  
 );  
};  
  
export default App;

# Explanation of JSX and Inline CSS Usage

• JSX was used to create the heading, image, and data elements.  
• Inline CSS was applied using JavaScript objects such as headingStyle and officeStyle.  
• JavaScript expressions like ₹{office.rent} were used inside JSX.  
• The officeList array was looped using map() to dynamically render multiple office blocks.