**WEEK-7**

**React JS HOL- 10**

**• Define JSX**

JSX (JavaScript XML) is a syntax extension used in React. It allows writing HTML-like code within JavaScript. JSX makes it easier to create React elements that are rendered in the browser.

**• Explain about ECMA Script**

ECMAScript is the standard that defines how JavaScript works. ES6 introduced modern features like let, const, arrow functions, classes, and modules which are now commonly used in React applications.

**• Explain React.createElement()**

React.createElement() is a function provided by React to create UI elements without using JSX. It takes three arguments: the element type, its properties, and its content.

For example:  
React.createElement('h1', null, 'Hello') creates a heading element.

**• Explain how to create React nodes with JSX**

React nodes can be created using JSX by writing HTML-like tags directly in JavaScript code.

For example:  
const element = <h1>Hello React</h1>;  
This creates a React node that renders an <h1> element with the text "Hello React".

**• Define how to render JSX to DOM**

JSX can be rendered into the actual HTML DOM using ReactDOM.render(). This function tells React where to place your component in the HTML file.

Example:  
ReactDOM.render(<App />, document.getElementById('root'));

**• Explain how to use JavaScript expressions in JSX**

JavaScript expressions (like variables or functions) can be used inside JSX by wrapping them in curly braces {}.

Example:  
<h1>Hello, {name}</h1> will display the value of the name variable.

**• Explain how to use inline CSS in JSX**

Inline CSS in JSX is written as a JavaScript object with style properties in camelCase.

Example:  
<h1 style={{ color: 'blue', fontSize: '20px' }}>Styled Heading</h1>

**App.js**

import React from 'react';

import './App.css';

function App() {

  const element = "Office Space";

  const imagePath = "https://goyalco.com/wp-content/uploads/2024/09/how-to-choose-an-office-space.png";

  const officeList = [

    { Name: "DBS", Rent: 50000, Address: "Chennai" },

    { Name: "Cognizant", Rent: 70000, Address: "Hyderabad" }

  ];

  return (

    <div className="container">

      <h1>{element}, at Affordable Range</h1>

      {officeList.map((office, index) => {

        const rentClass = office.Rent <= 60000 ? 'textRed' : 'textGreen';

        return (

          <div key={index}>

            <img src={imagePath} width="15%" height="15%" alt="Office Space" />

            <h1>Name: {office.Name}</h1>

            <h3 className={rentClass}>Rent: Rs. {office.Rent}</h3>

            <h3>Address: {office.Address}</h3>

          </div>

        );

      })}

    </div>

  );

}

export default App;

**App.css**

 body {

  margin: 0;

  padding: 0;

  font-family: Arial, sans-serif;

}

.container {

  display: flex;

  flex-direction: column;

  align-items: center;

  justify-content: center;

  min-height: 100vh;

  text-align: center;

  background-color: #f9f9f9;

}

.textRed {

  color: red;

  font-weight: bold;

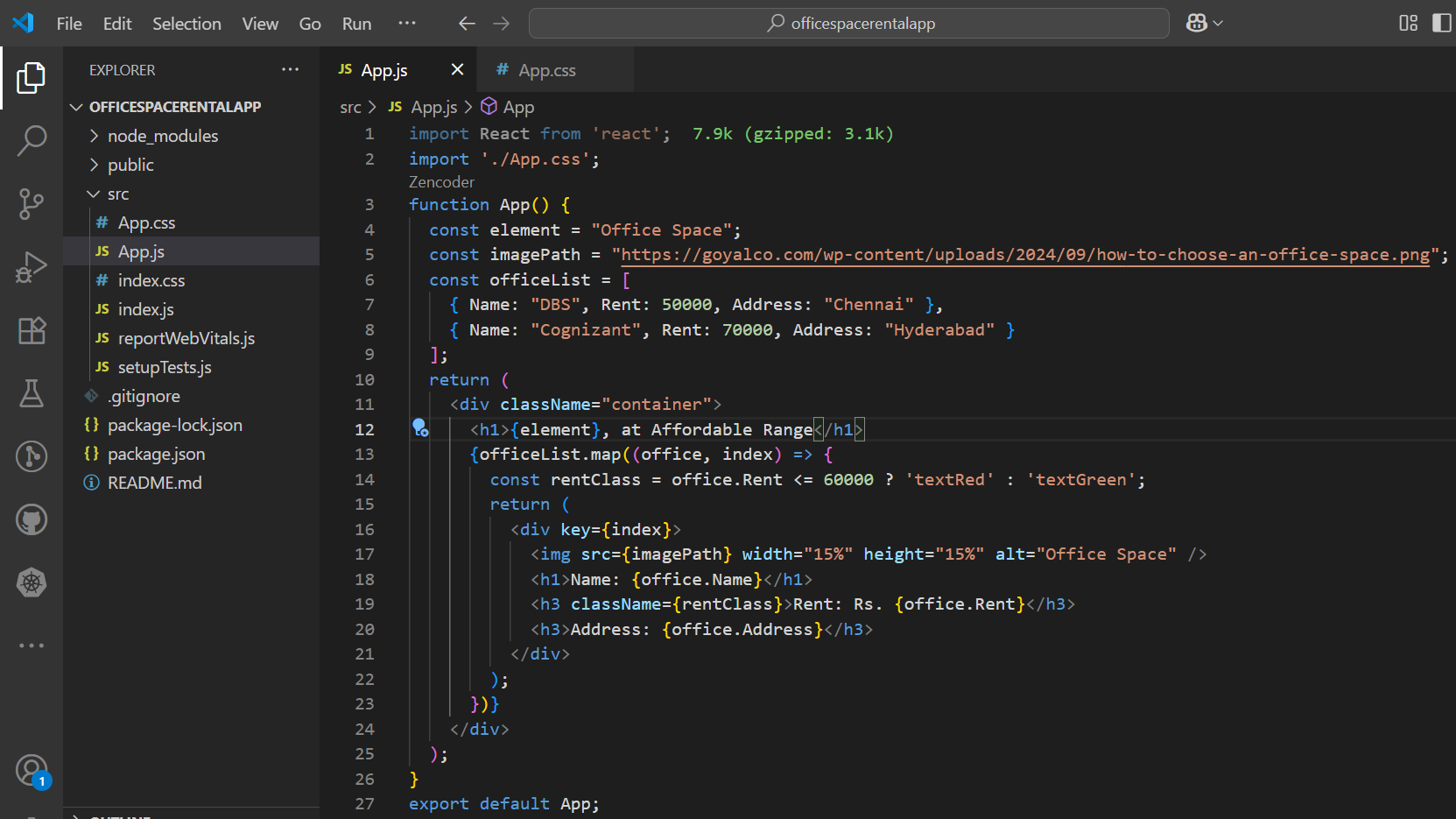
}

.textGreen {

  color: green;

  font-weight: bold;

}



**Output:**

