**10.React-HOL**

**1. Define JSX**

JSX (JavaScript XML) is a syntax extension for JavaScript used in React. It allows writing HTML-like code in JavaScript files, which gets transformed into React elements.

**2. Explain about ECMA Script**

ECMAScript (ES) is a scripting language specification standardized by ECMA International. JavaScript is based on ECMAScript. ES6 introduced key features like let, const, arrow functions, classes, promises, destructuring, etc.

**3. Explain React.createElement()**

React.createElement() is a method used to create React elements manually.  
**Example:**

React.createElement('h1', null, 'Hello React');

This is how JSX gets compiled behind the scenes.

**4. Explain how to create React nodes with JSX**

You can write elements using JSX like this:

const element = <h1>Hello React</h1>;

JSX lets you embed variables, expressions, and components in a natural syntax.

**5. Define how to render JSX to DOM**

Use ReactDOM.render() to render JSX into a real DOM node:

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

**6. Explain how to use JavaScript expressions in JSX**

You can embed JavaScript expressions inside JSX using {}:

const name = "React";

const element = <h1>Welcome to {name}</h1>;

**7. Explain how to use inline CSS in JSX**

Inline styles in JSX are written as an object:

const style = { color: 'blue', fontSize: '20px' };

<h1 style={style}>Styled Text</h1>

**HANDS ON**

**App.js**

import React from 'react';

import officeImage from './assests/office.jpeg';

function App() {

  const office = {

    name: "DBS",

    rent: 50000,

    address: "Chennai"

  };

  const rentStyle = {

    color: office.rent < 60000 ? "red" : "green",

    fontWeight: "bold"

  };

  return (

    <div style={{ textAlign: "center", fontFamily: "Arial" }}>

      <h1>Office Space , at Affordable Range</h1>

      <img src={officeImage} alt="Office" style={{ width: "400px", height: "250px" }} />

      <h2>Name: {office.name}</h2>

      <p style={rentStyle}>Rent: Rs. {office.rent}</p>

      <p><strong>Address:</strong> {office.address}</p>

    </div>

  );

}

export default App;

**OUTPUT**

