**10. REACTJS – HOL**

**React App - Office Space Rental**

**Define JSX**

**JSX (JavaScript XML)** is a syntax extension for JavaScript used with React to describe what the UI should look like. It allows writing HTML-like code inside JavaScript files, making it easier to build and visualize UI components.

Example:

const element = <h1>Hello, JSX!</h1>;

**Explain ECMA Script**

**ECMAScript (ES)** is the standard specification that JavaScript follows. Modern JavaScript uses ES6 and later versions which introduced features like let/const, arrow functions, classes, template literals, destructuring, and more.

Example (Arrow function in ES6):

const greet = () => console.log("Hello!");

**Use React.createElement()**

This is the standard React API to manually create a React element without JSX. It takes three arguments:

1. Type of the element (like 'div', 'h1')
2. Props (attributes)
3. Children (text or other elements)

Example:

React.createElement('h1', { className: 'title' }, 'Hello from React');

**Create React nodes using JSX**

React nodes (or elements) can be created using JSX for simpler and readable code.

Example:

const element = <h2>This is a React node</h2>;

**Render JSX to DOM**

To display JSX on the browser, it must be rendered using ReactDOM.render() (in React 17 or below) or createRoot().render() (React 18+).

Example:

import ReactDOM from 'react-dom/client';

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(<App />);

**Use JavaScript expressions in JSX**

You can embed any JavaScript expression inside {} in JSX.

Example:

const user = 'Jayashree';

const greeting = <h1>Hello, {user}!</h1>;

**Use inline CSS in JSX**

Inline styles are written as JavaScript objects in JSX. The keys are camelCase versions of CSS property names.

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

const heading = <h1 style={style}>Styled Text</h1>;

**Step 1: Create React App**

Open terminal and run:

npx create-react-app officespacerentalapp

**Step 2: Open Project in VS Code**

cd officespacerentalapp

code .

**Step 3: Modify App.js**

import React from "react";

const officeSpaces = [

{

name: "SkyView Offices",

rent: 55000,

address: "123 Anna Salai, Chennai",

image: "https://via.placeholder.com/300x200?text=SkyView"

},

{

name: "Green Tech Hub",

rent: 75000,

address: "456 T Nagar, Chennai",

image: "https://via.placeholder.com/300x200?text=Green+Tech"

},

{

name: "Ocean View Tower",

rent: 60000,

address: "789 OMR, Chennai",

image: "https://via.placeholder.com/300x200?text=Ocean+View"

}

];

function App() {

return (

<div style={{ padding: "20px", fontFamily: "Arial" }}>

<h1 style={{ textAlign: "center", color: "#333" }}>

Office Space Rental App

</h1>

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

<div

key={index}

style={{

border: "1px solid #ccc",

padding: "10px",

marginBottom: "15px",

borderRadius: "10px",

boxShadow: "2px 2px 12px #aaa"

}}

>

<img

src={office.image}

alt={office.name}

style={{ width: "100%", borderRadius: "10px" }}

/>

<h2>{office.name}</h2>

<p>

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

</p>

<p style={{ color: office.rent > 60000 ? "green" : "red" }}>

<strong>Rent:</strong> ₹{office.rent}

</p>

</div>

))}

</div>

);

}

export default App;

**Step 4: Start the React App**

In terminal:

npm start

**Step 5: Output**

