**10.ReactJs HOL**

**1. Define JSX**

JSX (JavaScript XML) is a syntax extension for JavaScript used in React.  
It lets you write HTML-like code inside JavaScript, making it easier to create UI components.

**Example:**

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

**2. Explain about ECMAScript (ES or ES6)**

ECMAScript is the standard specification that JavaScript follows.

* ES6 (ECMAScript 2015) is a major update to JavaScript.
* It introduced new features like:
  + let, const
  + Arrow functions () => {}
  + Classes
  + Template literals
  + Modules (import, export)
  + Promises

**Example:**

const name = 'React';

console.log(`Welcome to ${name}!`);

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

React.createElement() is the function used by React to create a React element.

**Syntax:**

React.createElement(type, props, children)

**Example:**

const element = React.createElement('h1', null, 'Hello, world');

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

We can create React nodes (elements/components) using JSX easily:

**Example:**

const heading = <h2>Office Rentals</h2>;

function Office() {

return <p>This is an office component</p>;

}

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

Use ReactDOM.render() to display JSX in the browser:

**Example:**

import React from 'react';

import ReactDOM from 'react-dom';

const element = <h1>Welcome!</h1>;

ReactDOM.render(element, document.getElementById('root'));

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

We can write JavaScript expressions inside {} in JSX.

**Example:**

const name = "Poorani";

const element = <h1>Hello, {name}!</h1>;

We can use:

* Variables
* Functions
* Math
* Conditions (ternary)

const price = 50000;

const color = price < 60000 ? 'red' : 'green';

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

In JSX, inline CSS is written as a JavaScript object, using camelCase for style names.

**Example:**

const style = {

color: 'green',

backgroundColor: 'lightyellow',

padding: '10px'

};

const element = <h2 style={style}>Rent Office Space</h2>;

**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**.

**React App : officespacerentalapp**

**App.js**

import React from 'react';

import './App.css';

function App() {

  const office = {

    name: 'Prestige Tech Park',

    rent: 55000,

    address: 'Whitefield, Bangalore',

    image: 'https://images.pexels.com/photos/37347/office-sitting-room-executive-sitting.jpg?auto=compress&cs=tinysrgb&dpr=2&h=200'

  };

  const officeList = [

    {

      name: 'WeWork',

      rent: 45000,

      address: 'Koramangala, Bangalore',

      image: 'https://images.pexels.com/photos/256658/pexels-photo-256658.jpeg?auto=compress&cs=tinysrgb&dpr=2&h=200'

    },

    {

      name: 'IndiQube',

      rent: 65000,

      address: 'HSR Layout, Bangalore',

      image: 'https://images.pexels.com/photos/380768/pexels-photo-380768.jpeg?auto=compress&cs=tinysrgb&dpr=2&h=200'

    },

    {

      name: '91Springboard',

      rent: 60000,

      address: 'Indiranagar, Bangalore',

      image: 'https://images.pexels.com/photos/325229/pexels-photo-325229.jpeg?auto=compress&cs=tinysrgb&dpr=2&h=200'

    }

  ];

  return (

    <div className="App">

      <h1>Office Space Rental Portal</h1>

      <div className="office-card">

        <h2>{office.name}</h2>

        <img src={office.image} alt={office.name} />

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

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

          <strong>Rent:</strong> ₹{office.rent}

        </p>

      </div>

      <h2>Available Office Spaces</h2>

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

        <div key={index} className="office-card">

          <h3>{item.name}</h3>

          <img src={item.image} alt={item.name} />

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

          <p style={{ color: item.rent > 60000 ? 'green' : 'red' }}>

            <strong>Rent:</strong> ₹{item.rent}

          </p>

        </div>

      ))}

    </div>

  );

}

export default App;

**App.css**

.App {

  font-family: Arial, sans-serif;

  padding: 20px;

}

.office-card {

  border: 1px solid #ccc;

  border-radius: 10px;

  padding: 15px;

  margin-bottom: 20px;

  max-width: 500px;

  box-shadow: 0 2px 8px rgba(0,0,0,0.1);

}

.office-card img {

  width: 100%;

  height: auto;

  border-radius: 5px;

}

**Final Output:**



