**React**

**Handson 10**

**1.Define JSX:**

* JSX stands for JavaScript XML.
* It is a syntax extension for JavaScript used with React to describe what the UI should look like.
* It looks like HTML, but is converted into JavaScript code (React elements) by Babel.

**Example:**

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

**2.Explain ECMA Script:**

* ECMAScript (ES) is the standard scripting language that defines JavaScript.
* It is maintained by ECMA International (ECMA-262 specification).
* Versions:
  + ES5, traditional JS features.
  + ES6 / ES2015, a major update with features like let, const, arrow functions, classes, and modules.
  + ES7+, smaller yearly updates, for example, async/await and optional chaining.

**3.Explain React.createElement():**

* A core React API that creates a React element.
* **Syntax:**

React.createElement(type, props, children)

**Example:**

const element = React.createElement('h1', { className: 'title' }, 'Hello, JSX!');

**4. How to create React nodes with JSX:**

You can create a React node using JSX like this:

const element = <h2>Welcome to React</h2>;

**To create multiple nodes:**

const container = ( <div> <h1>Title</h1> <p>Paragraph text</p> </div> );

This gets compiled into multiple React.createElement() calls under the hood.

**5.How to render JSX to the DOM:**

* Use ReactDOM.render() to mount JSX into the actual DOM.
* **Example:**

import React from 'react';

import ReactDOM from 'react-dom';

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

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

**6. How to use JavaScript expressions in JSX:**

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

const name = "Aish";

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

**Examples:**

<p>{name.toUpperCase()}</p>

**7.  How to use inline CSS in JSX:**

1. Define the styles as a JavaScript object.
2. Use style={{ ... }} in JSX.

**Example:**

const headingStyle = {

color: 'blue',

fontSize: '24px',

backgroundColor: 'lightgray'

};

const element = <h1 style={headingStyle}>Styled Heading</h1>;

**Or inline directly:**

<h1 style={{ color: 'green', fontWeight: 'bold' }}>Inline Styled</h1>

**App.js:**

import React from 'react';

function App() {

  const heading = <h1>Office Space Rental App</h1>;

  const officeImagePath = "/office.jpeg";

  const office = {

    name: "Tech Hub",

    rent: 75000,

    address: "123, IT Park, Bangalore"

  };

  const officeList = [

    { name: "Startup Spot", rent: 55000, address: "12, MG Road, Chennai" },

    { name: "Elite Office", rent: 95000, address: "45, Gachibowli, Hyderabad" },

    { name: "Budget Works", rent: 40000, address: "7, Koramangala, Bangalore" },

  ];

  const getRentStyle = (rent) => ({

    color: rent < 60000 ? 'red' : 'green'

  });

  return (

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

      {heading}

      <img

        src={officeImagePath}

        alt="Office"

        style={{ width: '300px', height: '200px', marginBottom: '20px' }}

      />

      <h2>Featured Office</h2>

      <p><strong>Name:</strong> {office.name}</p>

      <p><strong>Rent:</strong> <span style={getRentStyle(office.rent)}>{office.rent}</span></p>

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

      <h2>Available Office Spaces</h2>

      <ul>

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

          <li key={index} style={{ marginBottom: '10px' }}>

            <p><strong>Name:</strong> {o.name}</p>

            <p><strong>Rent:</strong> <span style={getRentStyle(o.rent)}>{o.rent}</span></p>

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

            <hr />

          </li>

        ))}

      </ul>

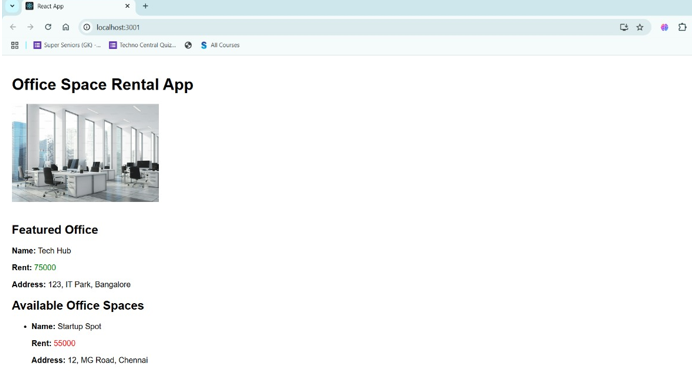
    </div>

  );

}

export default App;

**Output:**

****