**10. ReactJS-HOL**

**Objectives**

* **Define JSX**

JSX (JavaScript XML) is a syntax extension for JavaScript used in React. It allows you to write HTML-like code inside JavaScript.

Example:

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

JSX is not valid JavaScript, so it gets transpiled by tools like Babel into React.createElement() calls.

* **Explain about ECMA Script**

ECMAScript (ES) is the standard specification that JavaScript is based on.

Key Versions:

* ES5 (2009): Strict mode, array methods, JSON support.
* ES6 (2015): let, const, arrow functions, classes, promises, destructuring, modules, etc.

React uses many features from ES6 and newer versions for modern, clean coding.

* **Explain React.createElement()**

React.createElement() is the method that creates a virtual DOM element.

**Syntax:**

React.createElement(type, props, ...children);

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

create React nodes using JSX by assigning HTML-like tags to variables.

Example:

const element = <p>This is a paragraph.</p>;

* **Define how to render JSX to DOM**

JSX elements are rendered using ReactDOM.render().

Example:

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

ReactDOM.render(

element,

document.getElementById('root')

);

This renders the element into the DOM element with id="root" in index.html.

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

Embed **JavaScript expressions** inside JSX using { }.

Example:

const name = "Lucifer";

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

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

React allows you to style elements inline using JavaScript objects.

Example:

<h2 style={{ color: 'red', fontWeight: 'bold' }}>Inline CSS</h2>

**Creating a React Application named “officespacerentalapp” which uses React JSX to create elements, attributes and renders DOM to display the page.**

**Creating an element to display the heading of the page.**

**Attribute to display the image of the office space**

**Creating an object of office to display the details like Name, Rent and Address.**

**Creating 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.**

1. **Project Setup**

npx create-react-app officespacerentalapp

cd officespacerentalapp

1. **App.js**

import React from "react";

import "./App.css";

import featuredImage from "./image1.jpeg";

import officeListImg from './image2.jpeg';

import weworkImg from './image3.jpeg';

import mindspaceimg from './image4.jpeg'

function App() {

// Single office

const office = {

name: "DBS Business Center",

rent: 55000,

address: "T.Nagar, Chennai",

image: featuredImage,

};

// Office list with image URLs

const officeList = [

{

name: "Tidel Park",

rent: 65000,

address: "OMR, Chennai",

image: officeListImg,

},

{

name: "WeWork",

rent: 45000,

address: "Indiranagar, Bangalore",

image: weworkImg,

},

{

name: "Mindspace",

rent: 70000,

address: "HiTech City, Hyderabad",

image: mindspaceimg,

},

];

return (

<div className="App">

<h1>Office Space</h1>

{/\* Featured Office \*/}

<h2>At Affordable Range</h2>

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

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

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

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

</p>

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

<hr />

{/\* Office list \*/}

<h2>Other Available Offices</h2>

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

<div key={index}>

<img src={item.image} alt={item.name} className="office-image" />

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

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

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

</p>

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

</div>

))}

</div>

);

}

export default App;

1. **App.css**

.App {

text-align: center;

font-family: Arial, sans-serif;

margin: 20px;

}

.office-image {

width: 300px;

height: auto;

margin: 20px 0;

}

1. **Run the project:**

Npm start

<http://localhost:3000>





