10-REACT-JS-HOL

1. Define JSX

JSX (JavaScript XML) is a syntax extension for JavaScript used with React to describe what the UI should look like.

- JSX looks like HTML but is actually syntactic sugar for React.createElement() calls.
- JSX allows embedding HTML-like code directly in JavaScript.

Example: const element = <h1>Hello,

World!</h1>; Under the hood, it becomes:

javascript CopyEdit const element = React.createElement('h1', null,

'Hello, World!');

2. Explain ECMAScript (ES)

ECMAScript (often abbreviated as ES) is the standardized version of JavaScript defined by ECMA International.

- It ensures JavaScript works consistently across browsers and platforms.
- Major versions: ES5 (2009), ES6/ES2015 (biggest update), followed by annual updates like ES7, ES8, etc.

ES6 introduced features like: let, const, arrow functions, classes, template literals, destructuring, spread/rest operators, Map, Set, promises, and modules.

3. Explain React.createElement()

This is the core function used by React to create a virtual DOM node.

Syntax:

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

- type: HTML tag or React component (e.g., 'div' or MyComponent)
- props: an object with properties (e.g., { className: "box" })
- children: nested elements or text Example:

React.createElement('h1', { style: { color: 'blue' } }, 'Welcome!');

4. How to Create React Nodes with JSX

JSX allows you to create React nodes in a readable, HTML-like format:

const element = <div><h1>Hello JSX</h1></div>; You can use:

- Self-closing tags: <img />
- Nested elements: <div><p>Text</p></div>

Each JSX element is compiled to React.createElement().

5.How to Render JSX to DOM

You use ReactDOM.render() (for older React versions) or createRoot().render() in React 18+. jsx

CopyEdit import React from 'react'; import

ReactDOM from 'react-dom/client'; const

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

root =

ReactDOM.createRoot(document.getElemen

tById('root')); root.render(element);

Make sure the HTML has a div with id root:

<div id="root"></div>

6. Using JavaScript Expressions in JSX

You can use any JavaScript expression inside {} in JSX:

const name = "Harshi"; const element = <h2>Hello,

{name}</h2>; You can also use:

<p>{2 + 3}</p>

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

<p>{isLoggedIn ? "Logout" : "Login"}</p>

7. Using Inline CSS in JSX

In JSX, inline CSS is written using an object (camelCase for properties): jsx

CopyEdit const styleObj = {

color: "white",

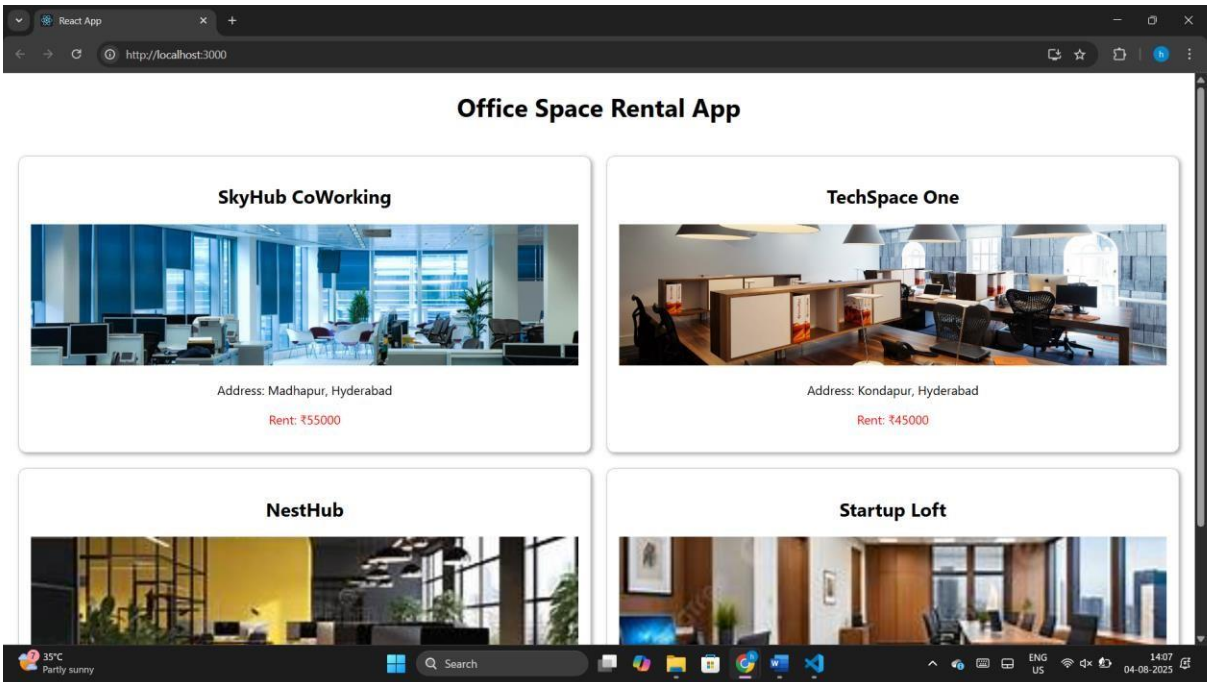
backgroundColor: "black",

padding: "10px"

```
};

const element = <h1 style={styleObj}>Styled Text</h1>;
```

Output -



Office Space Rental App

SkyHub CoWorking

Address: Madhapur, Hyderabad

Rent: ₹55000

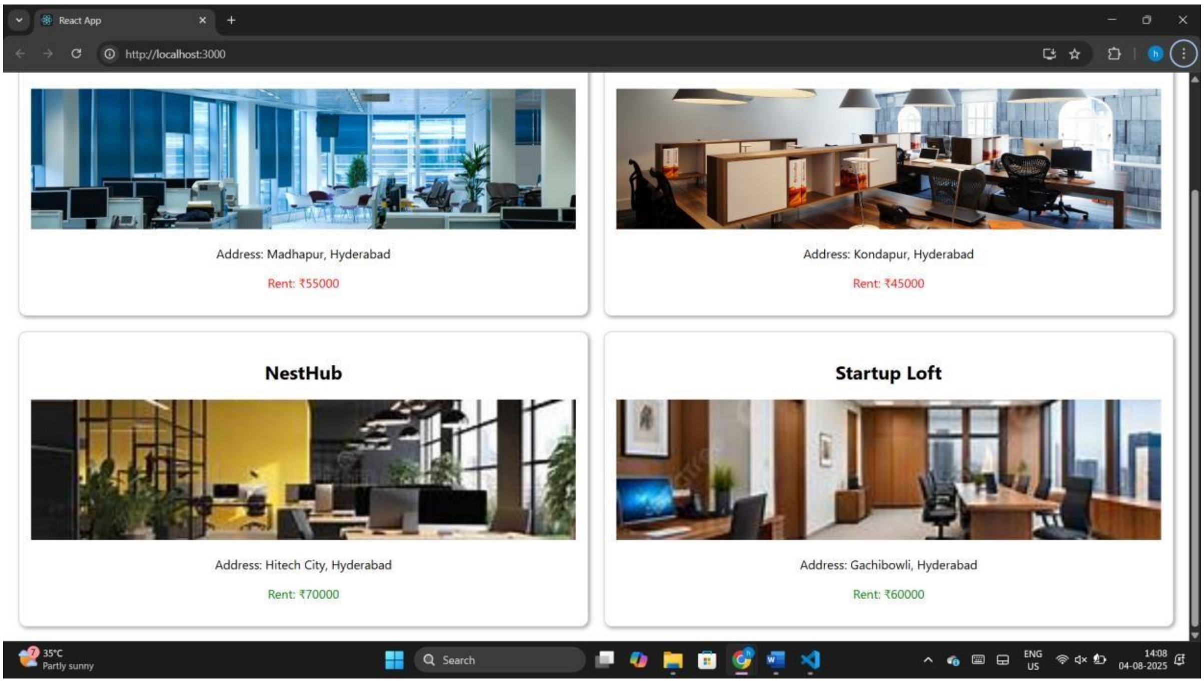
TechSpace One

Address: Kondapur, Hyderabad

Rent: ₹45000

NestHub

Startup Loft



Address: Madhapur, Hyderabad

Rent: ₹55000

Address: Kondapur, Hyderabad

Rent: ₹45000

NestHub

Address: Hitech City, Hyderabad

Rent: ₹70000

Startup Loft

Address: Gachibowli, Hyderabad

Rent: ₹60000