Cognizant Digital Nurture 4.0

Week-7

Name - Brishti Das

Superset ID - 6363729

React Hands-on Exercise 10

# Define JSX

* JSX (JavaScript XML) is a syntax extension for JavaScript used with React.
* It allows you to write HTML elements in JavaScript and place them in the DOM without using methods like createElement or appendChild.
* JSX makes code easier to write and understand.

# Explain about ECMA Script

* ECMAScript (ES) is a scripting-language specification standardized by ECMA International.
* It serves as the foundation for JavaScript.
* ES6 (also known as ECMAScript 2015) introduced important features like let, const, arrow functions, classes, template literals, and modules.

# Explain React.createElement()

* React.createElement() is a method used to create a React element without using JSX.
* It returns a plain JavaScript object that represents a virtual DOM element.
* Syntax: React.createElement(type, props, children)
* Example: React.createElement('h1', null, 'Hello World')

# Explain how to create React nodes with JSX

* React nodes can be created using JSX by embedding HTML-like tags in JavaScript.
* Example: const element = <h1>Hello, JSX!</h1>;
* JSX allows nesting, conditional expressions, and JavaScript expressions.

# Define how to render JSX to DOM

* JSX can be rendered to the DOM using ReactDOM.render().
* Syntax: ReactDOM.render(element, document.getElementById('root'));
* ReactDOM takes the JSX and mounts it to the specified DOM node.

# Explain how to use JavaScript expressions in JSX

* You can embed any JavaScript expression in JSX by wrapping it in curly braces `{}`.
* Example: const name = 'John'; const element = <h1>Hello, {name}</h1>;
* Expressions can include variables, functions, or any valid JavaScript code.

# Explain how to use inline CSS in JSX

* Inline styles in JSX are written as JavaScript objects, using camelCase property names.
* Example: const style = { color: 'blue', fontSize: '20px' };
* Then use it: <h1 style={style}>Styled Text</h1>