FEMI.

Node JS

Node.js is an open-source, server-side runtime environment that allows JavaScript to run outside the browser, enabling developers to build fast, scalable network applications, APIs, and server-side scripts. It uses an event-driven, non-blocking I/O model, making it efficient for handling concurrent requests.

NPM package

The npm package is the Node Package Manager, a tool for managing JavaScript libraries and dependencies in Node.js projects. It allows easy installation, updating, and management of packages from a registry with over 2 million packages.

React Fragment

A React Fragment is a lightweight wrapper in React that groups multiple elements without adding extra nodes to the DOM. It's used to return multiple children from a component without a parent element, like <div>. Syntax: <React.Fragment> or shorthand <>.

React Component

A React component is a reusable, self-contained piece of code that defines a part of a user interface in React. It can be a function or class, takes props as input, and returns JSX to render UI elements. Components can manage their own state and lifecycle.

React Props

React props (short for properties) are immutable data passed from a parent component to a child component in React. They allow components to receive and use dynamic data or configuration, enabling reusability and customization. Props are accessed as an object in the child component.

useState

useState is a React Hook that allows functional components to manage state. It returns a state variable and a function to update it, enabling dynamic UI updates when the state changes. Example: const [count, setCount] = useState(0);

useEffect

useEffect is a React Hook that handles side effects in functional components, like data fetching, DOM manipulation, or subscriptions. It runs after render and takes a callback function and an optional dependency array to control when it re-runs. Example: useEffect(() => { /* effect */ }, [deps]);