**What is Next.js?**

**Next.js** is a **React framework** developed by **Vercel** that helps you build **fast, scalable, and SEO-friendly** web applications.

"React ek JavaScript library hai jo UI banane ke liye use hoti hai, specially single-page applications (SPA). React ka main focus hai component-based architecture, jisme hum UI ko chhote-chhote reusable components me tod ke likhte hain. React virtual DOM ka use karta hai, jisse performance fast hoti hai.

"Virtual DOM ek virtual representation hai Real DOM ka. Jab bhi koi state ya props change hoti hai, React pehle Virtual DOM me changes karta hai, diff karta hai old vs new, aur sirf required jagah Real DOM me update karta hai — jisse performance fast ho jati hai."

**React Life Cycle**

Jab bhi ek **React component** banta hai, screen pe aata hai, update hota hai ya destroy hota hai — uss **poore process** ko React ka **Life Cycle** kehte hain.

**1. Mounting Phase (Component born hota hai)**

Jab component **first time DOM me render hota hai.**

**2. Updating Phase (Component update hota hai - props/state change hone par)**

Jab component ko naye props milte hain ya state change hoti hai.

**3 Unmounting Phase (Component destroy ho raha hai)**

Jab component **DOM se hata diya jaata hai.**

The React lifecycle is the process a component goes through from being created, updated, and finally removed. When it first loads, it mounts, then updates when data or props change, and unmounts when it’s no longer needed. In functional components, we use useEffect to handle all these phases.

Redux is a state management library used in React apps to handle global state. It stores data in a central place called a **store**, and components can access or update this data using **actions** and **reducers**. This makes it easier to manage and share state across different parts of the app.

**Reducers:**

A reducer is a pure function that updates the state based on the action it receives. It modifies the state in response to a specific action. In Redux, the state is immutable, meaning every update creates a new state object and doesn't change the old state.

 **State**: The data in your application.

  **Action**: A plain JavaScript object that carries the type of action and optional payload (data).

*  **Reducer**: A pure function that updates the state based on the action type.