**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.
* **Promise in JavaScript**
* A **Promise** is a JavaScript object that represents the eventual completion or failure of an asynchronous operation.  
  It has 3 states: **pending**, **fulfilled**, and **rejected**.  
  We use .then() for success and .catch() for error handling.  
  async/await is modern syntax to work with Promises in a cleaner way.

**Client-Side Rendering (CSR)**

* **"Client-Side Rendering mein server pe ek blank page bheja jata hai jisme sirf HTML structure hota hai. Browser HTML ko parse karta hai aur page ka structure (DOM tree) banata hai. Phir browser CSS aur JavaScript ko load karke page ko complete render karta hai. JavaScript page ko interactive banata hai aur jab user kisi cheez ko click karta hai, tab page dobara render hota hai without reloading the entire page. Yeh user experience ko smooth banata hai."**

**Difference from CSR:**

* **SSR** mein initial content server pe generate hota hai aur browser ko bheja jata hai.
* **CSR** mein pehle blank HTML bheja jata hai, phir client-side JavaScript use hota hai content ko render karne ke liye.

**Server-Side Rendering (SSR)**

**Server-Side Rendering mein, jab user page request karta hai, toh server pe pehle se HTML content generate hota hai aur browser ko bheja jata hai.** Yeh **content already rendered hota hai, toh page user ko jaldi dikhai deta hai. Phir JavaScript browser mein load hota hai aur page ko interactive banata hai. SSR mein initial page load fast hota hai, but interactivity ke liye client-side JavaScript ka use hota hai."**

**Page,txt**

**Next.js App Router ek file-based routing system use karta hai, jisme har route segment (yaani har folder) ke andar ek specific file page.tsx honi chahiye jo actual UI serve karti hai.**

**Isliye agar hum page.tsx ka naam badal kar kuch aur jaise home.tsx ya tinkal.tsx rakh dete hain, to Next.js use route ke liye koi valid page nahi samjhta, aur wo route break ho jaata hai.**

**Ye ek convention hai jo Next.js follow karta hai — jaise layout.tsx, error.tsx, loading.tsx bhi special files hain.**

**Components ka naam hum kuch bhi rakh sakte hain, lekin route serve karne wali file ka naam page.tsx hi hona chahiye.**

**page.tsx is a special filename in Next.js App Router.  
Changing it breaks the route because Next.js expects it to find and render a file named exactly page.tsx for each route segment.**

**What is SSG (Static Site Generation)?**

**SSG ka matlab hai Static Site Generation, jisme page ka HTML aur data dono build time pe hi generate kar liya jaata hai.  
Jab user website kholta hai, toh wo page ek static file ke form mein directly CDN se aata hai — isiliye bahut fast hota hai aur SEO friendly bhi hota hai.**

**💬 CDN kya hota hai? Content Delivery Network.**

**CDN ek network hota hai multiple servers ka, jo duniya ke alag-alag locations par hote hain.  
Iska kaam hota hai static files (jaise HTML, CSS, JS, images) ko user ke najdeek wale server se serve karna — taaki website fast load ho.**

**ISR (Incremental Static Regeneration)**

**ISR ek technique hai jisme Next.js static pages ko background mein re-generate karta hai, based on a timer — taaki naye data ke saath updated version mil jaye bina downtime ke.**

**SSR ek rendering technique hai jahan page har request pe server pe render hota hai, aur user ko dynamic data ke saath ready HTML milta hai.  
Ye SEO friendly hota hai aur jab content frequently change hota hai tab kaafi useful hota hai.**

**SSR real-time dynamic content ke liye perfect hai, especially jab hume SEO bhi chahiye aur data bhi frequently change hota ho.”**

**PPR (Partial Pre-Rendering)**

**Partial Pre-Rendering (PPR) ek hybrid rendering approach hai Next.js mein, jisme ek hi page ke kuch parts statically render hote hain (SSG/ISR) aur kuch parts dynamically server-render hote hain (SSR) — ek hi request ke andar!**