

SSR vs CSR

◆ What is CSR (Client-Side Rendering)?

CSR মানে হলো:

👉 Browser (client) নিজে HTML তৈরি করে JavaScript ব্যবহার করে

Flow (CSR)

User → Server → empty HTML + JS

User Browser → JS runs → API call → HTML render

Example (React – CSR)

```
function App() {  
  const [data, setData] = useState([]);  
  
  useEffect(() => {  
    fetch("/api/posts")  
      .then(res => res.json())  
      .then(setData);  
  }, []);  
  
  return <div>{data.map(p => <p>{p.title}</p>)}</div>;  
}
```

Characteristics

- ✓ Initial page blank থাকে
- ✓ JS load হলে data আসে
- ✓ Heavy JS dependency

◆ Pros of CSR

- ✓ Smooth SPA experience
- ✓ Less server load
- ✓ Good for dashboards, admin panels
- ✓ Rich interactions

◆ Cons of CSR

- ✗ Slow first paint
- ✗ Bad SEO (without tricks)
- ✗ JS disabled হলে page broken
- ✗ Poor performance on slow devices

◆ What is SSR (Server-Side Rendering)?

SSR মানে হলো:

👉 Server আগে থেকেই **HTML তৈরি করে**, তারপর browser এ পাঠায়

Flow (SSR)

User → Server

Server → Fetch data → Render HTML

User ← Fully rendered HTML

Example (Next.js – SSR)

```
export async function getServerSideProps() {
  const res = await fetch("https://api.example.com/posts");
  const data = await res.json();

  return { props: { data } };
}

export default function Page({ data }) {
  return data.map(p => <p>{p.title}</p>);
}
```

◆ Pros of SSR

- ✓ Fast first content load
- ✓ Excellent SEO
- ✓ Better for low-end devices
- ✓ Social media preview works

◆ Cons of SSR

- ✗ Higher server cost
- ✗ Slower TTFB sometimes
- ✗ Every request hits server
- ✗ More complex infra

● SSR vs CSR (Direct Comparison)

Feature	CSR	SSR
Rendering happens	Browser	Server
First page load	Slow	Fast
SEO	Poor	Excellent
JS dependency	High	Medium
Server load	Low	High
Best for	Dashboard, SPA	Blog, e-commerce
FAANG usage	Mixed	Mixed

🔥 Interview Trick: When to use WHAT?

✅ Use CSR when:

- Logged-in user dashboard
- Real-time apps (chat, analytics)
- SEO not important

✅ Use SSR when:

- Public pages
- SEO critical pages
- First impression matters

🧠 Hybrid Approach (FAANG style)

Modern apps use **BOTH** ⚡

Next.js supports:

- CSR
- SSR
- SSG
- ISR

Example:

Home Page → SSR

Blog → SSG

Dashboard → CSR

◆ Real-life Analogy 🍽️

CSR:

Order → Ingredients → Cook at table
(You wait)

SSR:

Order → Cooked food arrives
(Instant eating 😊)

◆ One-liner for Interview 🎯

"CSR renders in the browser after JS loads, while SSR renders HTML on the server, improving SEO and first contentful paint."

◆ TL;DR

- ✓ CSR = client renders
- ✓ SSR = server renders
- ✓ SSR = SEO + fast first paint
- ✓ CSR = interactive apps
- ✓ Best apps = hybrid

ওয়েব ডেভেলপমেন্টে **SSR** এবং **CSR** হলো দুটি প্রধান পদ্ধতি যা নির্ধারণ করে একটি ওয়েবসাইট কীভাবে ব্রাউজারে রেন্ডার (প্রদর্শন) হবে।

১. SSR (Server-Side Rendering)^২

SSR পদ্ধতিতে যখন একজন ইউজার কোনো ওয়েবসাইটের জন্য রিকোয়েস্ট পাঠান, তখন সার্ভার নিজেই সেই পেজটির সম্পূর্ণ HTML কোড তৈরি করে ব্রাউজারে পাঠায়।^৩ ব্রাউজার শুধু সেই HTML ফাইলটি রিসিভ করে সরাসরি স্ক্রিনে দেখায়।^৪

- **কিভাবে কাজ করে:** ইউজার রিকোয়েস্ট করেন → সার্ভার ডাটাবেস থেকে তথ্য নিয়ে পেজ তৈরি করে → সম্পূর্ণ পেজ ব্রাউজারে পাঠায় → ইউজার সাথে সাথে কন্টেন্ট দেখতে পান।^৫
- **সুবিধা:**
 - **SEO Friendly:** সার্চ ইঞ্জিন ক্রলাররা (যেমন Google Bot) খুব সহজেই কন্টেন্ট ইনডেক্স করতে পারে।^৬
 - **দ্রুত কন্টেন্ট লোডিং:** ইউজারের কাছে আগে থেকেই তৈরি করা HTML আসে, তাই প্রথমবার লোড হতে সময় কম লাগে।^৭
- **অসুবিধা:**
 - প্রতিটি নতুন পেজের জন্য সার্ভারে রিকোয়েস্ট পাঠাতে হয়, যা সার্ভারের ওপর চাপ বাড়ায়।

২. CSR (Client-Side Rendering)

CSR পদ্ধতিতে সার্ভার থেকে শুরুতে একটি ফাঁকা বা খুব সাধারণ HTML ফাইল এবং একটি বড় JavaScript ফাইল পাঠানো হয়।^৮ এরপর ব্রাউজার (ক্লায়েন্ট) সেই জাভাস্ক্রিপ্ট রান করে ওয়েবসাইটটির বাকি কন্টেন্ট তৈরি বা রেন্ডার করে।^৯

- **কিভাবে কাজ করে:** ইউজার রিকোয়েস্ট করেন → সার্ভার একটি ফাঁকা HTML ও JS পাঠায় → ব্রাউজার JS ডাউনলোড ও রান করে → এরপর ডাটা ফেচ করে পেজ তৈরি করে ইউজারকে দেখায়।
- **সুবিধা:**
 - একবার লোড হয়ে গেলে অন্য পেজে যাওয়ার সময় আর পেজ রিলোড হয় না (Single Page Application বা SPA)।¹⁰
 - সার্ভারের ওপর কাজের চাপ কম থাকে কারণ রেন্ডারিংয়ের কাজ ইউজারের ব্রাউজার করে।¹¹
- **অসুবিধা:**
 - প্রথমবার লোড হতে কিছুটা বেশি সময় নিতে পারে।
 - SEO-এর জন্য এটি কিছুটা চ্যালেঞ্জিং হতে পারে।

৩. SSR vs CSR: মূল পার্থক্য

বৈশিষ্ট্য	SSR (Server-Side)	CSR (Client-Side)
রেন্ডারিং কোথায় হয়?	ওয়েব সার্ভারে (Server)	ইউজারের ব্রাউজারে (Client)
SEO	চমৎকার (SEO Friendly)	কিছুটা দুর্বল (SEO Challenges)
Initial Load Speed	খুব দ্রুত	কিছুটা ধীর (JS লোড হতে সময় লাগে)
Caching	ভালো কাজ করে	কিছুটা জটিল
ব্যবহারের ক্ষেত্র	ব্লগ, ই-কমার্স, নিউজ সাইট	ড্যাশবোর্ড, সোশ্যাল মিডিয়া অ্যাপ

কোনটি কখন ব্যবহার করবেন?

- যদি আপনার ওয়েবসাইটটি কন্টেন্ট-নির্ভর হয় এবং **SEO** খুবই গুরুত্বপূর্ণ হয় (যেমন: ব্লগ বা নিউজ পোর্টাল), তবে **SSR** ব্যবহার করা ভালো।¹²
- যদি আপনি এমন কোনো অ্যাপ্লিকেশন তৈরি করেন যেখানে প্রচুর ইউজার ইন্টারঅ্যাকশন প্রয়োজন এবং একবার লোড হওয়ার পর সুপার ফাস্ট হতে হবে (যেমন: Facebook বা Gmail), তবে **CSR** সেরা।¹³

বর্তমানে **Next.js**-এর মতো ফ্রেমওয়ার্ক ব্যবহার করে একই অ্যাপে প্রয়োজন অনুযায়ী SSR এবং CSR উভয় সুবিধা নেওয়া যায়।

আপনি কি **Next.js** বা **React**-এর কন্টেক্সটে এগুলো কীভাবে ইমপ্লিমেন্ট করে তা জানতে চান?