

একদম ডিটেইলসে বলছি। 😊

♦ **SEO.js** ফাইলের আসল উদ্দেশ্য

একটা ওয়েবসাইটের **SEO (Search Engine Optimization)** খুব গুরুত্বপূর্ণ। বিশেষ করে তুমি যেহেতু **Headly** প্রজেক্টটা **Next.js** দিয়ে বানাচ্ছ, SEO ম্যানেজ করার জন্য একটা **Reusable Utility Component** দরকার হবে।

SEO.js ফাইলটা সাধারণত একটি **React component / helper** হয়, যেখানে তুমি প্রতিটি পেইজে SEO-র জন্য আলাদা আলাদা **meta tags, Open Graph tags, Twitter card tags** সেট করতে পারবে।

♦ কেন আলাদা ফাইল দরকার?

ধরে নাও, তোমার ওয়েবসাইটে আছে —

- Home page
- Blog page
- Product page
- About page

প্রতিটি পেজেই আলাদা আলাদা **title, description, keywords, image preview** দরকার।

যদি তুমি প্রতিটি পেজে আলাদা করে **<Head>** ট্যাগ লিখো, তাহলে কোড বারবার রিপিট হবে।

SEO.js ফাইলটা বানিয়ে রাখলে তুমি কেবল পেজে বসিয়ে **props** পাঠিয়ে দিবে, আর এটা তোমার জন্য **meta tags** জেনারেট করে দিবে।

♦ **SEO.js** কিভাবে কাজ করে

উদাহরণ (Next.js):

- `// src/components/common/SEO.js`
- `import Head from "next/head";`
-
- `export default function SEO({ title, description, keywords, image, url }) {`
- `const defaultTitle = "Headly CMS";`
- `const defaultDescription = "Fast headless CMS for modern sites.";`
- `const defaultImage = "/default-og.png";`
- `const siteUrl = "https://headlycms.com";`

-
- return (
- <Head>
- { /* Basic SEO */
- <title>{title ? `\${title} | \${defaultTitle}` : defaultTitle}</title>
- <meta name="description" content={description || defaultDescription} />
- <meta name="keywords" content={keywords || "CMS, headless, Next.js, SEO"} />
-
- { /* Open Graph (for Facebook, LinkedIn, etc.) */
- <meta property="og:type" content="website" />
- <meta property="og:title" content={title || defaultTitle} />
- <meta property="og:description" content={description || defaultDescription} />
- <meta property="og:image" content={image || defaultImage} />
- <meta property="og:url" content={url || siteUrl} />
-
- { /* Twitter Card */
- <meta name="twitter:card" content="summary_large_image" />
- <meta name="twitter:title" content={title || defaultTitle} />
- <meta name="twitter:description" content={description || defaultDescription} />
- <meta name="twitter:image" content={image || defaultImage} />
- </Head>
-);
- }

♦ কিভাবে ব্যবহার করবে

ধরা যাক, তোমার একটা ব্লগ ডিটেইল পেজ আছে:

- import SEO from "@components/common/SEO";
-
- export default function BlogDetail({ blog }) {
- return (
- <>
- <SEO
- title={blog.title}
- description={blog.excerpt}
- image={blog.coverImage}
- url={`https://headlycms.com/blog/\${blog.slug}`}
- />
- <article>
- <h1>{blog.title}</h1>

- `<p>{blog.content}</p>`
- `</article>`
- `</>`
- `);`
- `}`

♦ কী কী সুবিধা দেবে

1. **Reusable** → প্রতিটি পেজে বারবার `<Head>` লেখার দরকার হবে না।
2. **Consistency** → সব পেজে একই স্ট্রাকচারের SEO tags থাকবে।
3. **Dynamic SEO** → প্রতিটি ব্লগ বা প্রোডাক্টের জন্য আলাদা meta tags সেট করতে পারবে।
4. **Social Sharing Optimization** → Facebook, LinkedIn, Twitter-এ লিঙ্ক শেয়ার করলে সুন্দর preview আসবে।
5. **Performance** → Next.js এর SSR + SEO কম্বিনেশন গুগলে ভালো র‍্যাঙ্ক করবে।

-