**General Informations**

Next.js is an open-source web development framework created by the private company Vercel providing React-based web applications with server-side rendering and static website generation.

OR

Next.js is a popular open-source framework for building web applications and websites with React. It is based on the Node.js runtime and provides a streamlined development experience for creating server-rendered or statically generated React applications.

OR

Next.js is a React framework that gives you building blocks to create web applications. By framework, we mean Next.js handles the tooling and configuration needed for React, and provides additional structure, features, and optimizations for your application.

**Why we use Next.js?**

Next.js is widely used in the web development community due to its developer-friendly features, performance optimizations, and the ability to build modern web applications with ease. It has gained popularity for building static websites, blogs, e-commerce platforms, and even larger-scale applications.

**Features of Next.js:**

Here are some key features and concepts of Next.js:

1. Server-side rendering (SSR): Next.js allows you to render React components on the server and send HTML to the client, which improves performance and enables search engine optimization (SEO). This means that your web pages can be pre-rendered on the server and delivered as fully-rendered HTML to the browser, providing a faster initial load time and better SEO visibility.
2. Static site generation (SSG): Next.js supports generating static HTML files at build time, allowing you to pre-render pages in advance and serve them directly from a CDN (Content Delivery Network). This approach is ideal for websites with content that doesn't change frequently, providing fast and scalable performance.
3. Client-side rendering (CSR): Next.js can also handle client-side rendering when needed. This allows you to build dynamic, interactive web applications by fetching data on the client side and updating the UI without reloading the entire page.
4. Automatic code splitting: Next.js automatically splits your JavaScript code into smaller chunks, which are loaded only when needed. This improves initial load times and ensures that users only download the code they actually need for a particular page.
5. Routing: Next.js provides a simple and intuitive routing system. You can define dynamic routes, nested routes, and catch-all routes easily, making it straightforward to create complex applications with multiple pages.
6. API routes: Next.js includes an API routes feature that allows you to define serverless endpoints within your application. These endpoints can handle HTTP requests and perform server-side logic, providing a convenient way to build backend functionality alongside your frontend code.
7. TypeScript support: Next.js has built-in TypeScript support, enabling you to write type-safe code and benefit from enhanced tooling and developer productivity.

**Note:-**

* In Next.Js Project, routes will be created as per Folder & files name. These names are case sensitive in the routes, so make sure all folder & file name in lower-case.  
  Ex:  
  Let say we want to create an api to update task in todo app, where id is going to take from the request as query. So id would be dynamic. To make id as dynamic route, we need to make id file, where its file name would look like [id].  
  Then it will access like, http://localhost/api/task/[id]  
  Where in the place of [id], would be dynamic id like: 6482f221a904344b8cd8d8d4