vivasoft

Learnathon 2.0 - Next.js

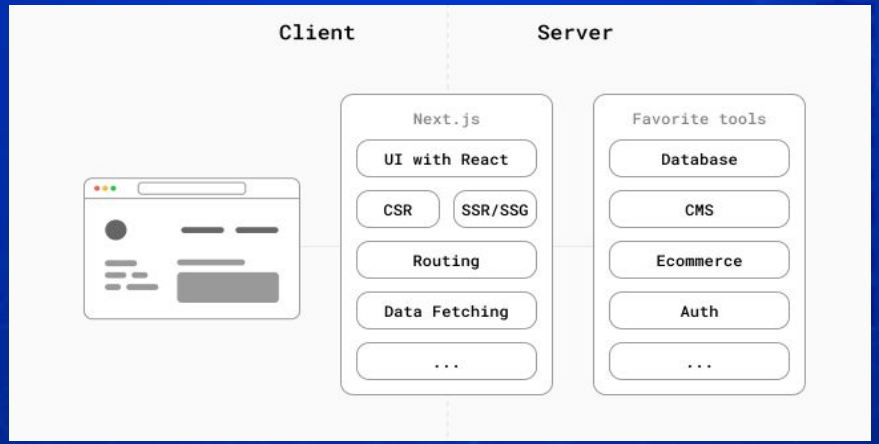
Tasfia Islam - Senior Software Engineer (L-1), Vivasoft Limited.

Learnathon - 2.0

vivasoft

Let’s talk about Next.js

- A React.js Framework
- Handles tooling and configuration needed for React
- Additional structures and features for optimization
- Faster development process



vivasoft

Use Cases

- Use React.js for SPAs, flexibility and control, Lightweight projects
- Use Next.js for SSR and SEO optimization, hybrid applications with combining SSR and CSR, Streamlined faster development.

vivasoft

Next.js 13 New Features

- App Directory - built on React Server Components
- Suspense Boundary
- Client vs Server Component
- Routing
- Layout etc.

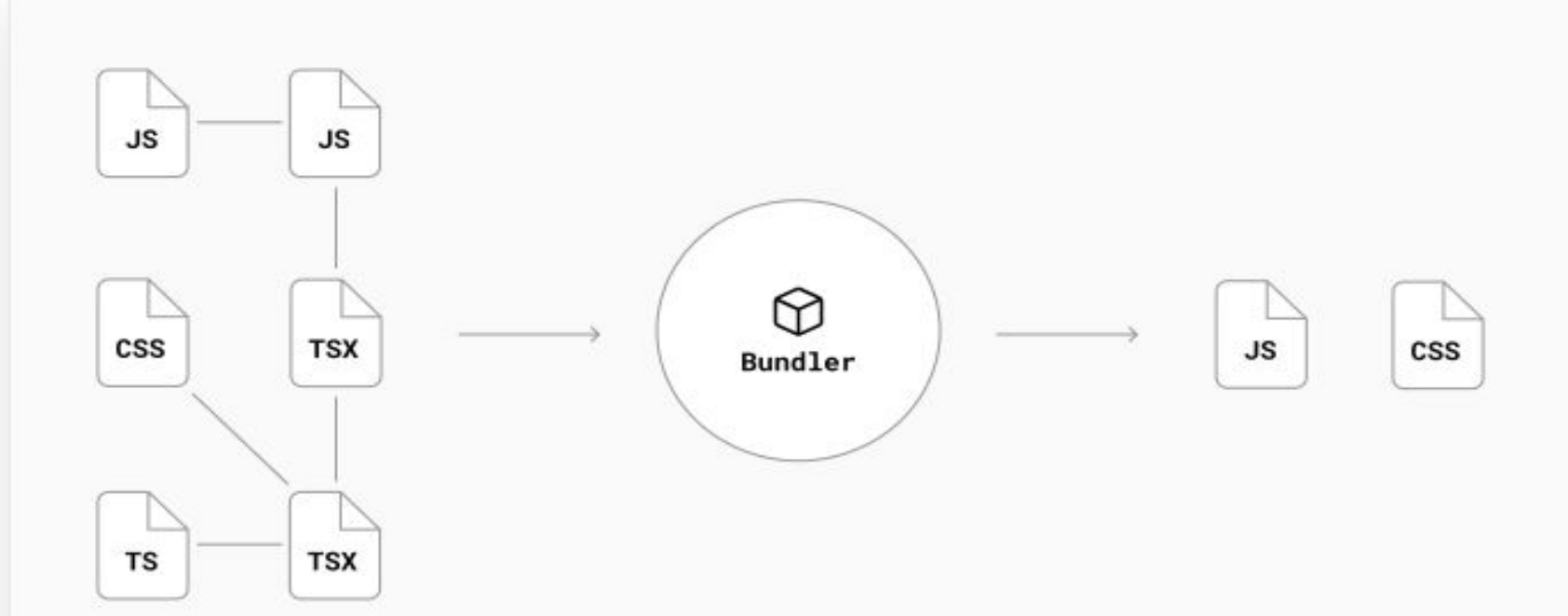
vivasoft

Few terms to be familiar with

- Hydration - converts pre-rendered HTML from the server into a fully interactive application
- Suspense Boundary - display fallback until children have finished loading
- Bundle
- SSR, CSR

vivasoft

Bundler

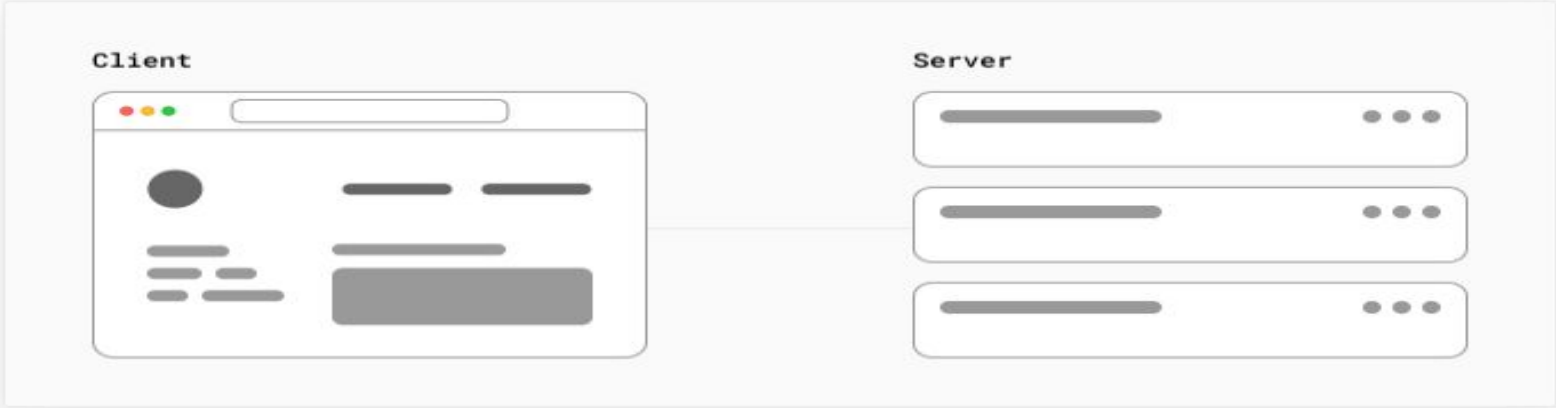


vivasoft

Client & Server

Client and Server

In the context of web applications, the **client** refers to the browser on a user's device that sends a request to a server for your application code. It then turns the response it receives from the server into an interface the user can interact with.



vivasoft

SSR & Server Components

- No hydration needed
- Improve performance
- Good for SEO
- For security of sensitive information from API

With SSR, there's a series of steps that need to be completed before a user can see and interact with a page:

1. First, all data for a given page is fetched on the server.
2. The server then renders the HTML for the page.
3. The HTML, CSS, and JavaScript for the page are sent to the client.
4. A non-interactive user interface is shown using the generated HTML, and CSS.
5. Finally, React hydrates the user interface to make it interactive.

vivasoft

CSR & Client Components

- Big JS Bundle
- Bad for SEO
- More load time in browser
- Interactivity
- Access to Browser APIs

Browser requests HTML file from server

Server responds very quickly with a simple HTML document and links to CSS, JavaScript etc.

Blank page or page with loading animations is displayed while the browser executes JavaScript and the rest

Required data is retrieved, views are generated and integrated into the DOM.

Web application is fully loaded and ready to use.

Client-Side Rendering

vivasoft

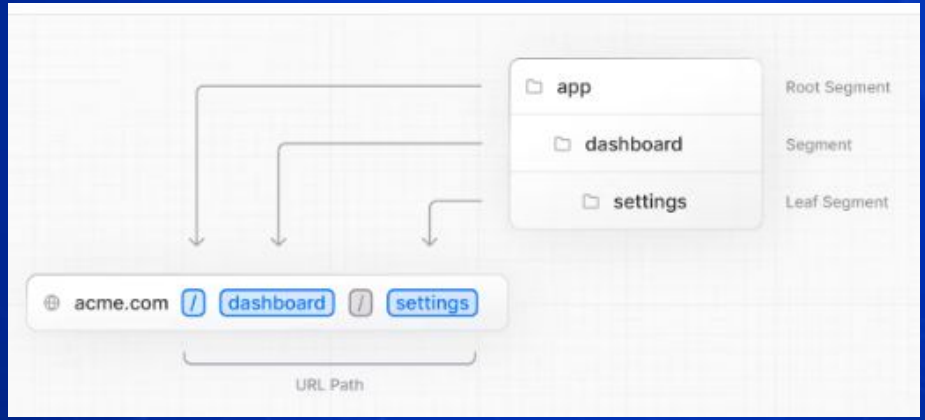
File Conventions 📂

- Page Files - page.tsx
- Layout Files, Template - layout.tsx, template.tsx
- API Routes - route.tsx
- Error, Loading - error.tsx, not-found.tsx, loading.tsx
- Metadata Files

vivasoft

Routing

- App Router - default server components
- Pages and layouts
- Route Groups
- Dynamic Routes



vivasoft

Linking and Navigation

- **<Link>** Component - built-in component , extends HTML **<a>** tag

- **useRouter()** hook: imported from **next/navigation,** used only in **Client components**

vivasoft

Data Fetching

- On the server - fetch (caches automatically)
- On the server - 3rd party libraries
- On the client - Route handler
- On the client - 3rd party libraries

vivasoft

Any queries?

Learnathon - 2.0

vivasoft

Learnathon - 2.0