

# Introduction to NextJs

Course Code: CSC 4182    Course Title: Advanced Programming In Web Technologies



**Dept. of Computer Science**  
**Faculty of Science and Technology**

<b>Lecture No:</b>	<b>1</b>	<b>Week No:</b>	<b>09</b>	<b>Semester:</b>	
<b>Lecturer:</b>	<i>Sazzad Hossain; sazzad@aiub.edu</i>				

# Lecture Outline



- ✓ What is NextJS
- ✓ NextJS Features
- ✓ File-based Routing
- ✓ Dynamic Routes
- ✓ 404 file
- ✓ What is rendering?
- ✓ Types of Rendering
- ✓ Client-Side Rendering(CSR)
- ✓ Pre-rendering
- ✓ Server-side Rendering
- ✓ Static Site Generation (SSG)
- ✓ Static Site Generation (SSG) VS Server-Side Rendering (SSR)

# What is NextJS



Next.js is a flexible **React framework** that gives you building blocks to create fast web applications.

Next.js is a **React framework** designed to facilitate the development of web applications

- Provides building blocks and tools for creating web apps
- Handles tooling and configuration needed for React
- Offers additional structure, features, and optimizations

# What is NextJS



Client

Server



Next.js

UI with React

CSR

SSR/SSG

Routing

Data Fetching

...

Favorite tools

Database

CMS

Ecommerce

Auth

...

# Building Blocks of a Web Application



There are a few things you need to consider when building modern applications. Such as:

- **User Interface** - how users will consume and interact with your application.
- **Routing** - how users navigate between different parts of your application.
- **Data Fetching** - where your data lives and how to get it.
- **Rendering** - when and where you render static or dynamic content.
- **Integrations** - what third-party services you use (CMS, auth, payments, etc) and how you connect to them.
- **Infrastructure** - where you deploy, store, and run your application code (Serverless, CDN, Edge, etc).
- **Performance** - how to optimize your application for end-users.
- **Scalability** - how your application adapts as your team, data, and traffic grow.
- **Developer Experience** - your team's experience building and maintaining your application.

# NextJS Features



- In the development stage, Next.js optimizes for the developer and their experience building the application. It comes with features that aim to improve the **Developer Experience** such the TypeScript and ESLint integration, Fast Refresh, and more.
- Since each environment has different considerations and goals, there is a lot that needs to be done to move an application from development to production. For instance, the application code needs to be compiled, bundled, minified, and code split.

# NextJS Features



- Folder based routing and Dynamic routing
- Pre-rendering
  - Server-side rendering
  - Static generation

# Folder-based Routing



**Follow following documents for Quiz and Project;**

<https://nextjs.org/docs/app/getting-started/project-structure>

<https://nextjs.org/docs/app/getting-started/layouts-and-pages>

<https://nextjs.org/docs/app/getting-started/linking-and-navigating>



# What is rendering?



**Rendering** is the process that **turns the code** you write **an application** in into something that users can interact with on a web page.

# Types of Rendering



## Pre-rendering

- Static Site Generation (SSG)
- Server Side Rendering (SSR)

## No Pre-rendering

- Client-Side Rendering(CSR)

# Client-Side Rendering(CSR)

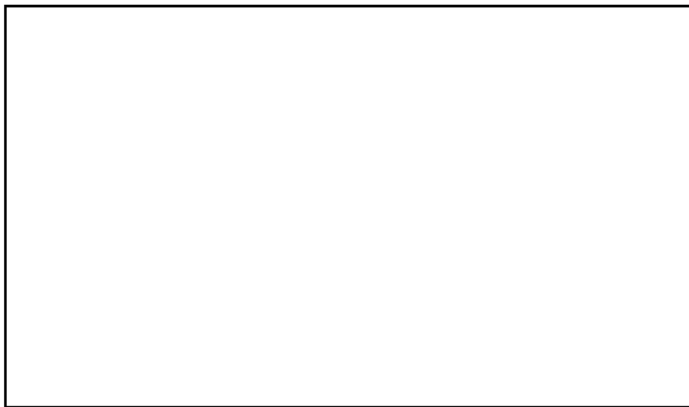


## No Pre-rendering (Plain React.js app)

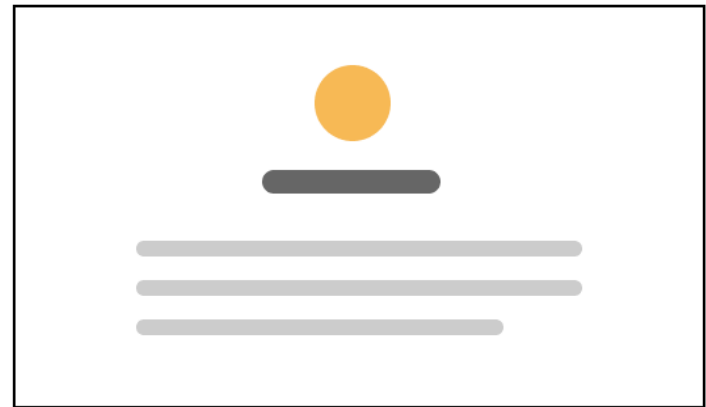
### Initial Load:

App is not rendered

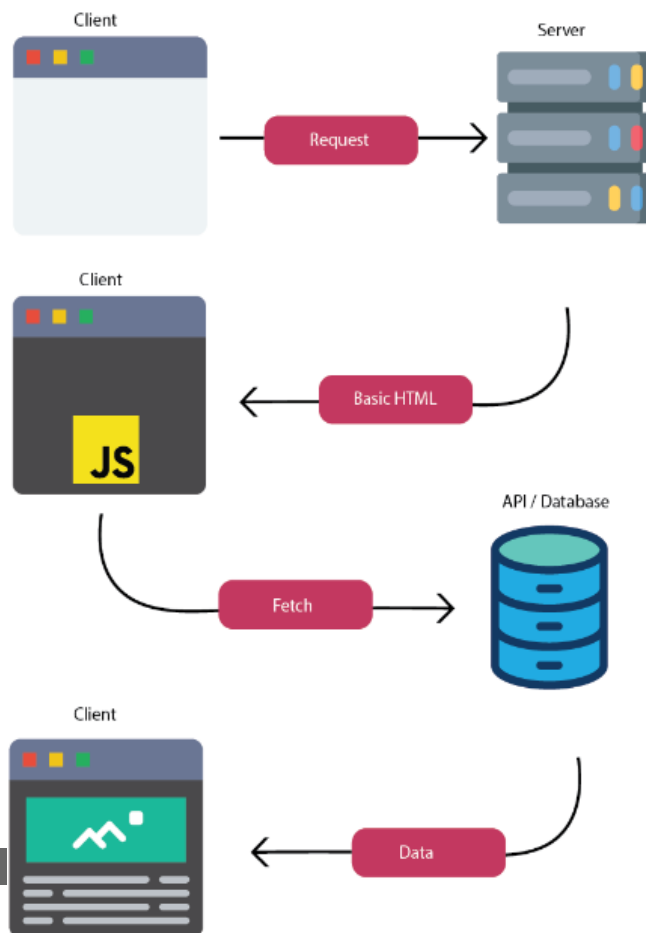
**Hydration:** React components are initialized and App becomes interactive



JS loads  
→



# Client-Side Rendering(CSR)



# Pre-rendering



- By default, Next.js pre-renders every page.
- Next.js *generates HTML for each page in advance*, instead of having it all done by client-side JavaScript.
- Pre-rendering can result in better performance and SEO.
- Each generated HTML is associated with minimal JavaScript code necessary for that page.
- When a page is loaded by the browser, its JavaScript code runs and makes the page fully interactive. (This process is called **hydration**.)

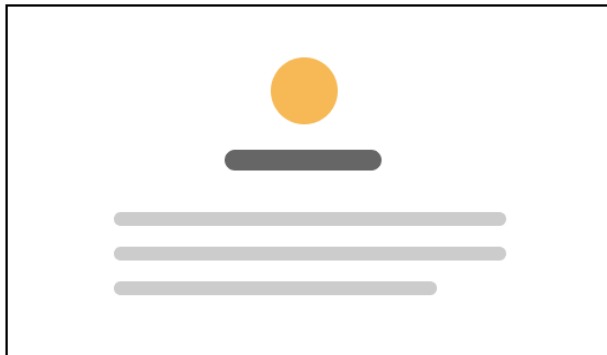
# Pre-rendering



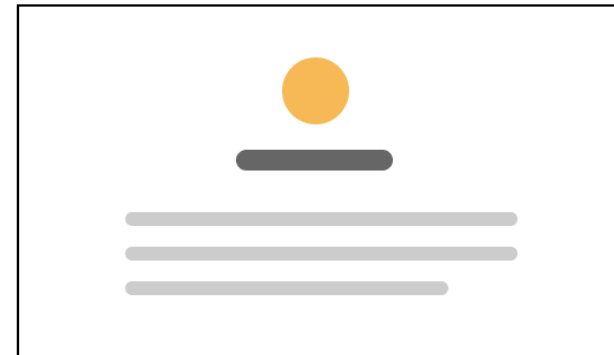
## Pre-rendering (Using Next.js)

### Initial Load:

Pre-rendered HTML is displayed

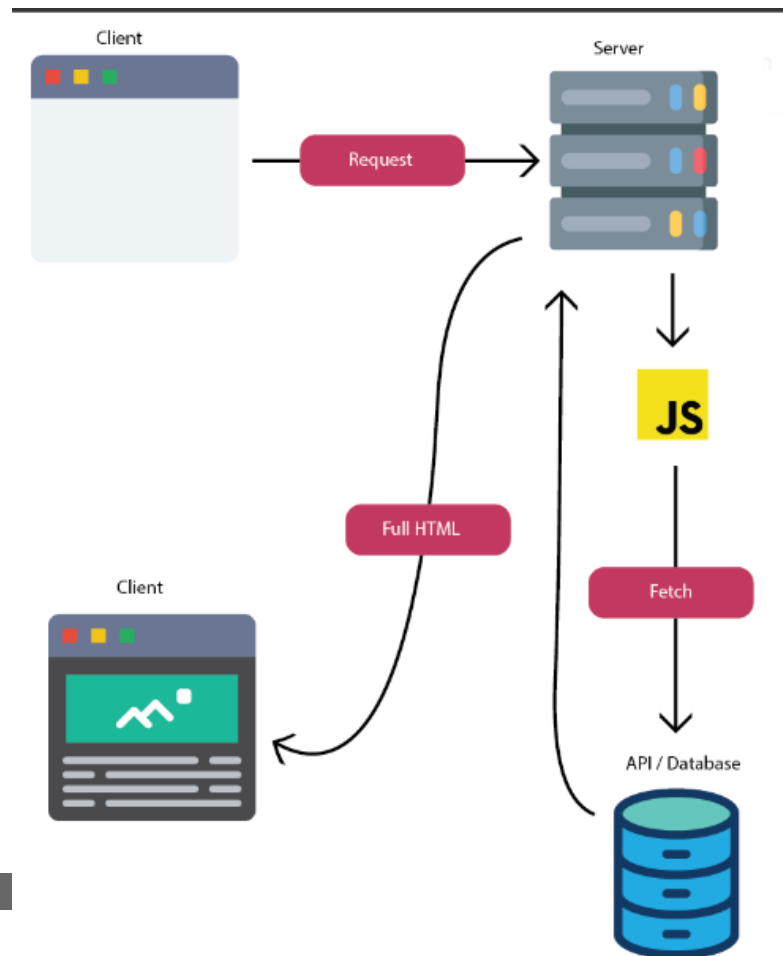


JS loads  
→



If your app has interactive components like `<Link />`, they'll be active after JS loads

# Pre-rendering



# Server-side Rendering



**Server-side Rendering** is the pre-rendering method **that generates the HTML on each request.**

Page request

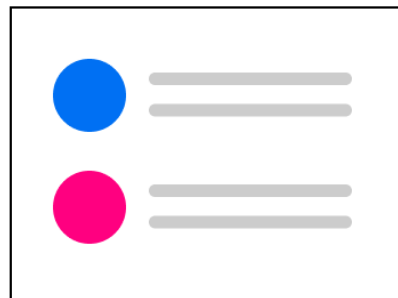


~~Next~~.js



Fetches external data

```
[  
  { img: ●, text: ... },  
  { img: ●, text: ... },  
]
```



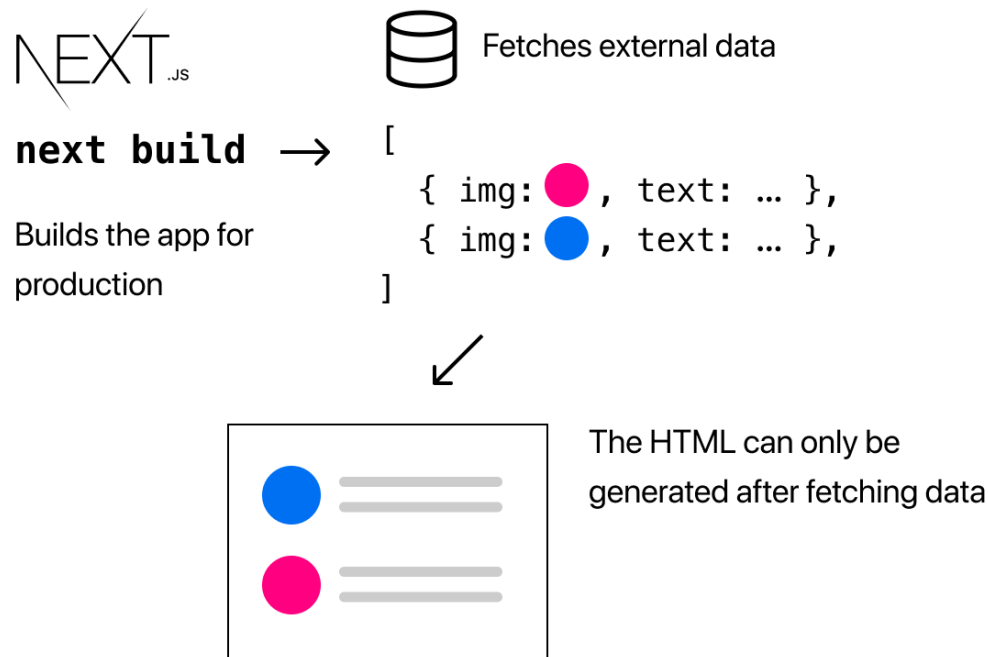
The HTML is generated



# Static Site Generation (SSG)



- Static Generation is the pre-rendering method that generates the HTML at **build time**.
- The pre-rendered HTML is then reused on each request.



# Static Site Generation (SSG) VS Server Side Rendering (SSR)



## SSR



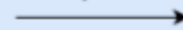
Request



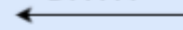
Response: Dynamically generated HTML

SERVER

Request



DATA

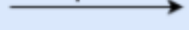


DB

## SSG

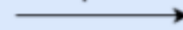


Request



SERVER

Request

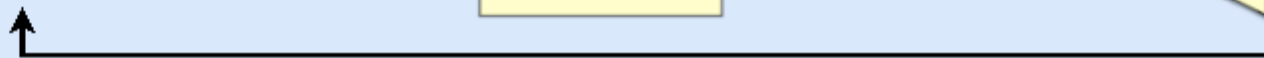


JS

HTML

CSS

Response: STATIC HTML, CSS, JS





# References

1. W3Schools Online Web Tutorials, URL: <http://www.w3schools.com>
2. Next.js, URL: <https://nextjs.org/>
3. Mozilla Developer Networks, URL: <https://developer.mozilla.org/>



Thank You!