**Next.js and React 18: Key Features and Conventions**

**Next.js Routing**

**1. Router Types**

* **App Router**: Used for server-side rendering (SSR).
* **Page Router**: Used for client-side rendering (CSR).

**2. Using Routers**

* **App Router**: Utilizes getStaticProps and getServerSideProps methods in pages.
* **Page Router**: Latest version fevers using App Router.

**3. File-Based Routing Mechanism**

* All routes are placed inside the app folder.
* Files containing routing names must be named page.tsx or page.jsx.

**4. Catch-All Segments**

* Rename files to [...slug] for catch-all routing.
* Access parameters with:

typescript

{

params: {

slug: string [];

};

}

**5. Private Folders**

* Cannot navigate or route through private folders.
* Name private folders starting with an underscore (\_folder Name).

**6. Route Groups**

* Used to organize related routes.
* Example: For login, sign-up, and password recovery, create an auth folder with login, signup, and forgot-password files inside.
* Access pages without the auth prefix in the URL.

**7. Route Group Layouts**

* Create a layout.tsx in the folder to apply a layout to all files within.
* Example: Create layout.tsx in the auth folder to apply to login, signup, etc.
* If only layout.tsx is present and no page.tsx, the layout displays alone.

**8. Metadata**

* Define metadata in root layout or specific pages.
* Use the generateMetadata function:

typescript

import {Metadata} from "next";

export const generateMetadata = ({params}: Props): Metadata => {

return {

title: "Home Page",

description: "This is the home page",

};

}

* Define metadata as an object:

typescript

import {Metadata} from "next";

export const metadata: Metadata = {

title: {

absolute: "",

default: "",

template: "%s | Next.js"

},

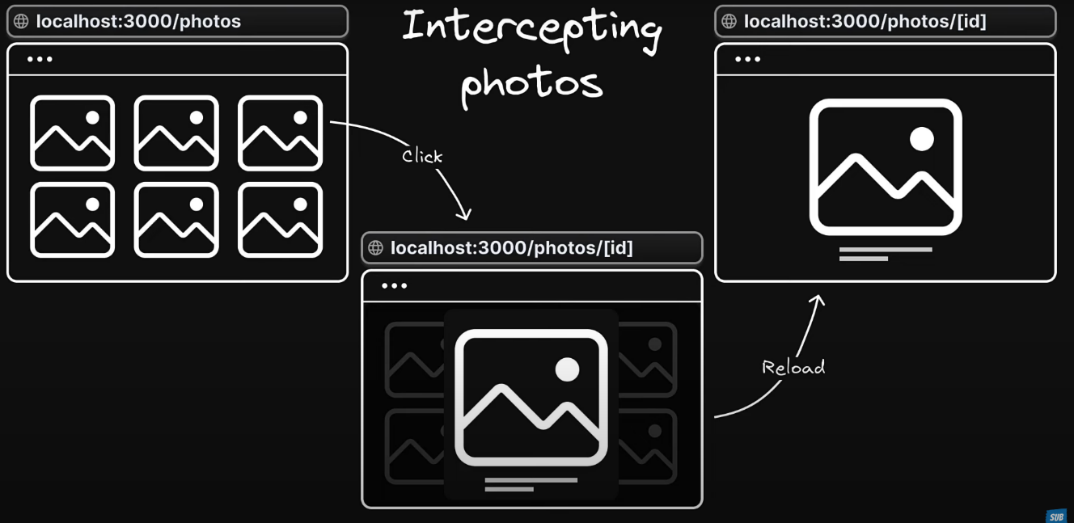
description: "This is the home page",

}

**9. Parallel Routes**

* Used for independent UI sections.
* Example: Create components in a folder starting with @ (e.g., @notification).
* Import components with @ directly into page.tsx.

**10. Intercepting Routes**

* Displays UI before the main page.
* Create intercepting routes with a folder name starting with a dot (.).
* Example: For a login intercepting route, create .login folder.
* Use .. to go one level up in the hierarchy:
  + .. for parent level
  + ... for root level
* This is the example image for batter under standing
  + 

**React 18 Components**

**1. Server Components**

* Access data from databases and read files.
* Fetch data from the server.
* Cannot interact with the user.

**2. Client Components**

* Interact with the user and render UI.
* Cannot access databases or read files.
* Mark a component as client-side by adding "use client"; at the top of the file.