page.tsx

loading.tsx

error.tsx

layout.tsx

not-found.tsx

api\hello\route.tsx

export async function GET(request: Request) {

  return new Response('Hello, Next.js!')

}

---

router.push(`/${search}/`) // working like a redirect

---

generateMetadata

metadata

<main></main>

---

import { useRouter } from "next/navigation"

const router = useRouter()

---

const handleSubmit = (e) => {

    e.preventDefault()

    setSearch('')

    router.push(`/${search}/`)

}

---

import Link from "next/link"

<Link href="/">WikiRocket!</Link>

---

lib/getAllUsers.ts

lib/getUser.ts

export default async function getUser(userId: string) {

    const res = await fetch(`https://jsonplaceholder.typicode.com/users/${userId}`)

    if (!res.ok) return undefined

    or

    if (!res.ok) throw new Error('failed to fetch data')

    return res.json()

}

{users.map(user => {

    return (

        <>

            <p key={user.id}>

                <Link href={`/users/${user.id}`}>{user.name}</Link>

            </p>

            <br />

        </>

    )

})}

import { Suspense } from "react"

<Suspense fallback={<h2>Loading...</h2>}>

    <UserPosts promise={userPostsData} />

</Suspense>

export async function generateStaticParams() {

    const usersData: Promise<User[]> = getAllUsers()

    const users = await usersData

    return users.map(user => ({

        userId: user.id.toString()

    }))

}

<body>

<nav>

  <h1>My Navbar</h1>

</nav>

{children}

</body>

const fileNames = fs.readdirSync(postsDirectory);

const fullPath = path.join(postsDirectory, fileName);

const fileContents = fs.readFileSync(fullPath, 'utf8');

const matterResult = matter(fileContents);

allPostsData.sort((a, b) => a.date < b.date ? 1 : -1);

const processedContent = await remark()

    .use(html)

    .process(matterResult.content);

export default function getFormattedDate(dateString: string): string {

    return new Intl.DateTimeFormat('en-US', { dateStyle: 'long' }).format(new Date(dateString))

}

    const handleSubmit = async (e: FormEvent<HTMLFormElement>) => {

        e.preventDefault()

        console.log(JSON.stringify(data))

        const { name, email, message } = data

        // Send data to API route

        const res = await fetch('http://localhost:3000/api/feedback', {

            method: 'POST',

            headers: {

                'Content-Type': 'application/json'

            },

            body: JSON.stringify({

                name, email, message

            })

        })

        const result = await res.json()

        console.log(result)

        // Navigate to thank you

        router.push(`/thank-you/`)

    }

import { useRouter } from "next/navigation"

import { useState, useTransition, FormEvent, ChangeEvent } from 'react'

import { usePathname } from "next/navigation"