Softzino Academy

Welcome to NEXT JS

Presented by: Imran Hossain
Software Engineer

React Code Features

- **Component-Based Architecture**
- **Virtual DOM (Document Object Model)**
- **JSX (JavaScript XML)**
- **Declarative UI**
- **One-Way Data Binding**
- **Hooks** (like `useState`, `useEffect`, etc.)
- **State Management**
- **Concurrent Rendering**
- **Suspense**
- **Server Components**
- **Cross-Platform Support**

Next js code features

- Rendering
  - Server-Side Rendering (SSR)
  - Static Site Generation (SSG)
  - Incremental Static Regeneration (ISR)
  - Dynamic HTML Streaming
  - Cache Components
- Routing
  - File-system based routing
  - Advanced Routing & Nested Layouts
- Data Fetching (Server Actions, `fetch` API Extensions)
- Optimizations (Image, Font, Script, Code Splitting, Built-in CSS and JavaScript Bundling)
- Development Experience (Fast Refresh, Turbopack, React Compiler Support)
- API Routes
- TypeScript Support
- Internationalization (i18n) Support

- **Pre-rendering:** Next.js pre-renders every page **by default**, which means HTML is generated ahead of time for better performance and SEO. There are two main types:
    - **Static Generation (SSG):** HTML is generated **at build time**.

jsx Copy code

```jsx
export async function getStaticProps() {
  const res = await fetch('https://api.example.com/posts')
  const posts = await res.json()

  return {
    props: { posts },
  }
}

export default function Blog({ posts }) {
  return (
    <>
      <h1>Blog</h1>
      {posts.map(post => <p key={post.id}>{post.title}</p>)}
    </>
  )
}
```

- **Server-side Rendering (SSR):** HTML is generated **on every request**.

jsx

Copy code

```jsx
export async function getServerSideProps() {
  const res = await fetch('https://api.example.com/user')
  const user = await res.json()

  return {
    props: { user },
  }
}

export default function Profile({ user }) {
  return <h1>Hello, {user.name}</h1>
}
```

✅ *Benefit*: Choose between performance (SSG) or fresh data (SSR).

- **Incremental Static Regeneration (ISR):** You can **update static pages** after deployment — without rebuilding the whole site.

jsx Copy code

```jsx
export async function getStaticProps() {
  const res = await fetch('https://api.example.com/products')
  const products = await res.json()

  return {
    props: { products },
    revalidate: 10, // re-generate every 10 seconds
  }
}
```

✅ *Benefit*: Combines the best of static and dynamic worlds.

- **File-based Routing:** Next.js uses your project's `app/` or `pages/` directory to automatically create routes.

🧩 **Example:**

bash — Copy code

```
/pages/index.js       →   '/'
/pages/about.js       →   '/about'
/pages/blog/[id].js   →   '/blog/1', '/blog/2', etc.
```

`pages/about.js`

jsx — Copy code

```jsx
export default function About() {
  return <h1>About Page</h1>
}
```

`pages/blog/[id].js`

jsx — Copy code

```jsx
import { useRouter } from 'next/router'

export default function BlogPost() {
  const { id } = useRouter().query
  return <h1>Blog Post ID: {id}</h1>
}
```

✅ *Benefit*: No need for React Router — routes are automatically handled.

- **API Routes:** You can create **serverless API endpoints** directly inside Next.js — no need for a separate backend.

bash Copy code

```
/pages/api/hello.js
```

jsx Copy code

```jsx
export default function handler(req, res) {
  res.status(200).json({ message: 'Hello API!' })
}
```

➡️ Access via `/api/hello`

✅ *Benefit*: Build full-stack apps inside one project.

- **Image Optimization:** Next.js automatically optimizes images with the `<Image>` component.

jsx Copy code

```jsx
import Image from 'next/image'

export default function Home() {
  return (
    <Image
      src="/profile.png"
      alt="Profile Picture"
      width={200}
      height={200}
    />
  )
}
```

✅ *Benefit*: Automatic resizing, lazy loading, and WebP conversion.

- **Middleware (Edge Functions):** Middleware runs **before a request is completed** — great for authentication, redirects, etc.

`middleware.js`

jsx Copy code

```jsx
import { NextResponse } from 'next/server'

export function middleware(request) {
  const token = request.cookies.get('token')
  if (!token) {
    return NextResponse.redirect(new URL('/login', request.url))
  }
  return NextResponse.next()
}
```

✅ *Benefit*: Run logic close to the user (fast and secure).

- **Server Actions:** You can write **server-side logic directly in React components** (no separate API route needed).

jsx Copy code

```jsx
'use server'

export async function savePost(formData) {
  const title = formData.get('title')
  await db.post.create({ data: { title } })
}
```

In your component:

jsx Copy code

```jsx
<form action={savePost}>
  <input name="title" />
  <button type="submit">Save</button>
</form>
```

✅ *Benefit*: Cleaner, full-stack code without creating `/api` endpoints.

- **Authentication (with NextAuth.js):** You can easily integrate authentication.

bash Copy code

```
npm install next-auth
```

`app/api/auth/[...nextauth]/route.js`

jsx Copy code

```jsx
import NextAuth from 'next-auth'
import GitHubProvider from 'next-auth/providers/github'

export const authOptions = {
  providers: [
    GitHubProvider({
      clientId: process.env.GITHUB_ID,
      clientSecret: process.env.GITHUB_SECRET,
    }),
  ],
}

export default NextAuth(authOptions)
```

✅ *Benefit*: Secure and simple authentication flows.

- **Internationalization (i18n):** Next.js supports built-in multilingual routing.

`next.config.js`

```js
module.exports = {
  i18n: {
    locales: ['en', 'fr', 'bn'],
    defaultLocale: 'en',
  },
}
```

➡️ `/`, `/fr`, `/bn`

✅ *Benefit*: Native support for multi-language websites.