



Basic Next.js สำหรับผู้เริ่มต้น



ในยุคปัจจุบัน เทคโนโลยีการพัฒนาเว็บแอปพลิเคชันก้าวหน้าไปอย่างรวดเร็ว Next.js ได้กลายเป็นมาตรฐานใหม่สำหรับ React Framework ที่นักพัฒนาทั่วโลกเลือกใช้ โดยเฉพาะเวอร์ชันล่าสุด Next.js 16 (App Router) ที่มาพร้อมฟีเจอร์ล้ำสมัยอย่าง Server Components และ Server Actions ซึ่งช่วยให้เว็บโหลดเร็วขึ้น ปลอดภัยขึ้น และเขียนโค้ดได้กระชับกว่าเดิม

หลักสูตรนี้ออกแบบมาเพื่อให้คุณก้าวข้ามจากการเขียน React แบบเดิม สู่การเป็น Modern Full-stack Developer โดยเน้นการลงมือทำจริงผ่าน Workshop สร้างระบบจัดการสินค้า (Inventory Management) คุณจะได้เรียนรู้การตกแต่งหน้าเว็บที่สวยงามและรวดเร็วด้วย Tailwind CSS ผสานพลังกับ Shadcn/UI ซึ่งเป็น Library ยอดนิยมที่สุดในขณะนี้ ที่ช่วยให้คุณ Copy & Paste คอมโพเนนต์ระดับมืออาชีพมาใช้ได้ทันที ไม่ว่าคุณต้องการอัปเดตเพื่อสมัครงาน หรือสร้างโปรเจกต์ของตัวเอง คอร์สนี้จะปูพื้นฐานที่ถูกต้องและทันสมัยที่สุดให้กับคุณ

วัตถุประสงค์:

- เข้าใจโครงสร้างและการทำงานของ Next.js 16 (App Router) และความแตกต่างระหว่าง Server vs. Client Components
- สามารถสร้างและจัดการ Routing, Layouts และ Navigation ในรูปแบบใหม่ (File-system based routing) ได้อย่างถูกต้อง
- สามารถใช้งาน Tailwind CSS ร่วมกับ Shadcn/UI เพื่อสร้าง User Interface ที่สวยงาม ทันสมัย และรองรับทุกหน้าจอ (Responsive)
- เข้าใจและประยุกต์ใช้ Server Actions เพื่อจัดการการส่งข้อมูล (Form Submission) และเชื่อมต่อกับระบบหลังบ้านโดยไม่ต้องเขียน API Route แบบเดิม
- สามารถสร้างระบบ Authentication และ CRUD (Create, Read, Update, Delete) ที่ทำงานได้จริง พร้อมระบบ Validation ที่ได้มาตรฐาน

กลุ่มเป้าหมาย:

- Frontend Developers ที่เขียน React ได้แล้ว และต้องการอัปเกรดสกีลมาใช้ Next.js เวอร์ชันล่าสุด
- Programmers / Software Engineers ที่ต้องการเปลี่ยนสายมาทำ Web Development ด้วย Modern Stack
- นักเรียน / นักศึกษา ที่ต้องการทำโปรเจกต์จบ หรือเตรียมความพร้อมก่อนเข้าสู่ตลาดแรงงานสาย Tech
- Freelance / Startup Founders ที่ต้องการสร้างเว็บแอปพลิเคชันที่มีประสิทธิภาพสูง (SEO & Performance) ด้วยตัวเอง

ความรู้พื้นฐาน:

- พื้นฐานการเขียนโปรแกรมมิ่ง
- พื้นฐานการทำงานกับเว็บแอปพลิเคชัน
- HTML & CSS: เข้าใจโครงสร้างเว็บ และการจัด Layout พื้นฐาน
- Modern JavaScript (ES6+)
- React Basics: เข้าใจคอนเซปต์ Component, Props



ระยะเวลาในการอบรม:

- 12 ชั่วโมง (2 วัน)

วิทยากรผู้สอน:

- อาจารย์สามิตร ไทยม

คอร์สที่ควรอบรมก่อนหน้า:

- ไม่มี

คอร์สต่อเนื่องที่แนะนำ:

- ไม่มี

เนื้อหาการอบรม:

Module 1: Introduction to Next.js 16 & App Router

- Next.js 16 Overview: ทำไมต้อง Next.js? (SEO, Performance, Full-stack capabilities)
- Installation: สร้างโปรเจกต์ด้วย create-next-app (TypeScript + Tailwind + ESLint)
- App Directory Structure: เข้าใจโครงสร้าง app/ ที่มาแทน pages/
- Server Components (RSC) vs Client Components:
 - คอนเซ็ปต์การ Render (Server-First)
 - การใช้คำสั่ง "use client" และ "use server"
 - กฎการใช้งาน: เมื่อไหร่ควรใช้ Server หรือ Client Component?
- Project Configuration: การตั้งค่า next.config.mjs และ tsconfig.json เบื้องต้น

Module 2: Routing, Layouts & Navigation

- File-System Based Routing: การสร้าง Route ผ่าน Folder และไฟล์ page.tsx
- Layouts & Templates:
 - layout.tsx (Root Layout & Nested Layouts)
 - การแฉ่ UI (Header/Sidebar) จ้าหน้า
- Navigation:
 - การใช้ <Link> Component
 - การใช้ Hook useRouter, usePathname, useSearchParams (จาก next/navigation)
- Dynamic Routes: การรับค่า Params [id] และ Query Strings
- Special Files:
 - loading.tsx (Streaming / Suspense)
 - error.tsx (Error Handling)
 - not-found.tsx (Custom 404)



Module 3: Styling with Tailwind CSS & Fonts

- Tailwind CSS in Next.js: นวนวนการตั้งค่า tailwind.config.ts
- Styling Fundamentals:
 - Utility Classes (Flexbox, Grid, Spacing, Typography)
 - Responsive Design (sm:, md:, lg:)
 - Pseudo-classes (hover:, focus:)
- Next/Font: การจัดการ Font (Google Fonts) เพื่อ Performance ที่ดีที่สุด
- CSS Modules (Optional): การใช้ .module.css ร่วมกับ Tailwind ในกรณีที่ต้องการ Custom Styles เฉพาะจุด

Module 4: Data Fetching & Server Actions (New Standard)

- Fetching Data in Server Components: การใช้ async/await และ fetch โดยตรง (ไม่ต้องใช้ useEffect)
- Caching & Revalidation: เข้าใจ Static Rendering vs Dynamic Rendering (force-dynamic, revalidate)
- Introduction to Server Actions:
 - การสร้าง Function สำหรับ Handle Form (action={submitFunc})
 - การใช้ revalidatePath เพื่ออัปเดตหน้าเว็บหลังแก้ไขข้อมูล
 - การใช้ redirect เพื่อเปลี่ยนหน้าฝั่ง Server

Module 5: Integrate Shadcn/UI (Modern UI Library)

- Introduction to Shadcn/UI:
 - แนวคิด "Not a library, It's code" (Copy & Paste)
 - ความแตกต่างจาก Material UI หรือ Bootstrap
- Setup & Configuration:
 - การติดตั้งผ่าน CLI (npm shadcn@latest init)
 - โครงสร้างโฟลเดอร์ components/ui และ lib/utils
- Theming: การปรับแต่งสี globals.css (CSS Variables) และ Radius
- Essential Components Workshop:
 - Layout Elements: Card, Separator, Aspect Ratio
 - Interactive: Button, Dropdown Menu, Dialog (Modal), Sheet (Drawer)
 - Form Elements: Input, Label, Textarea, Checkbox
- Dark Mode: การติดตั้ง next-themes เพื่อรองรับ Dark/Light Mode



Module 6: Workshop - Inventory Management System

- Workshop Structure: วางโครงสร้าง Layout (Login Layout vs Dashboard Layout)
- Building Authentication UI:
 - ออกแบบหน้า Login ด้วย Shadcn Card & Form
 - การใช้ React Hook Form + Zod เพื่อทำ Validation (Best Practice)
- Connecting to Backend (Simulation/API):
 - เขียน Server Action เพื่อรับค่า Login และเก็บ Cookie (HTTP-Only)
- CRUD Operations (Stock Page):
 - Read: แสดงรายการสินค้าด้วย Shadcn Table
 - Create: สร้างฟอร์มเพิ่มสินค้าใน Dialog หรือ Sheet
 - Update/Delete: การส่งค่า ID ไปยัง Server Action เพื่อแก้ไข/ลบ
- Production Deployment: การ Build (npm run build) และ Deploy ขึ้น Vercel