



## Next.js Advanced ขั้นสูง



ในหลักสูตร Next.js ขั้นสูงนี้ ผู้เรียนจะได้ก้าวข้ามจากการ "สร้างเว็บได้" ไปสู่การ "สร้างเว็บที่รองรับการขยายตัว (Scalable) และมีประสิทธิภาพสูงสุด" โดยเนื้อหาจะเจาะลึกฟีเจอร์ระดับสูงของ Next.js 16 บนสถาปัตยกรรม App Router

เราจะเรียนรู้เทคนิคที่ใช้ในงานจริงระดับองค์กร เช่น การทำ Authentication แบบปลอดภัยด้วย Auth.js (NextAuth v5), การจัดการข้อมูลที่ซับซ้อนด้วย Server Actions & Optimistic UI, การเรนเดอร์ประสิทธิภาพด้วย Partial Prerendering (PPR) และ Caching Strategy, ตลอดจนการจัดการ Global State และ URL State ที่ถูกต้อง

### วัตถุประสงค์:

- เข้าใจและสามารถใช้งานฟีเจอร์ขั้นสูงของ App Router เช่น Parallel Routes และ Intercepting Routes ได้
- สามารถสร้างระบบ Authentication & Authorization ที่ปลอดภัยและจัดการ Session ได้อย่างถูกต้อง
- สามารถออกแบบ Server Actions ที่ซับซ้อน พร้อมระบบ Validation และการจัดการ Error ขั้นสูง
- เข้าใจเทคนิค Optimistic UI เพื่อสร้างประสบการณ์ผู้ใช้ที่ลื่นไหล (ไม่ต้องรอโหลด)
- สามารถจัดการ Advanced Data Table (Sort, Filter, Pagination) แบบ Server-side ร่วมกับ Shadcn/UI
- เข้าใจการทำ Dockerization และการ Deploy ขึ้น Production Server จริง

### กลุ่มเป้าหมาย:

- Web Developer ที่ใช้งาน Next.js พื้นฐานได้แล้ว และต้องการต่อยอดสู่ระดับ Senior/Tech Lead
- Full-stack Developer ที่ต้องการเน้นความเชี่ยวชาญด้าน Modern React Stack
- ผู้ที่ผ่านการอบรมหลักสูตร Basic Next.js for Beginner มาก่อน

### ความรู้พื้นฐาน:

- เข้าใจ Next.js App Router พื้นฐาน (File structure, Pages, Layouts)
- เข้าใจ React Hooks (useState, useEffect, useTransition)
- เข้าใจ Tailwind CSS เบื้องต้น
- เข้าใจพื้นฐาน HTTP, API และ Database (SQL)

### ระยะเวลาในการอบรม:

- 18 ชั่วโมง (3 วัน)



#### วิทยากรผู้สอน:

- อาจารย์สามิตร โทยม

#### คอร์สที่ควรอบรมก่อนหน้า:

- Basic Next.js สำหรับผู้เริ่มต้น

#### คอร์สต่อเนื่องที่แนะนำ:

- ไม่มี

#### เนื้อหาการอบรม:

##### Module 1: Advanced Routing Patterns & Architecture

- Project Structure for Scale: การจัดโครงสร้างโปรเจกต์ขนาดใหญ่ (Feature-based vs Type-based)
- Parallel Routes (@folder): การแสดงผลหลาย Page ในหน้าจอกเดียว (เช่น Dashboard ที่มีหลาย Panel โหลดแยกกัน)
- Intercepting Routes (..): การทำ Modal/Dialog ที่มี URL เป็นของตัวเอง (เช่น กดรูปภาพแล้วเด้งเป็น Modal แต่พอกด Refresh กลายเป็นหน้าเต็ม)
- Route Groups & Multiple Layouts: การจัดการ Layout ที่ซับซ้อนสำหรับ User Role ที่ต่างกัน (Admin vs User)
- Middleware Mastery: การเขียน Middleware ดักจับ Request, Rewrite URL และจัดการ Headers ขั้นสูง

##### Module 2: Advanced Data Mutation & Server Actions

- Server Actions Deep Dive: การใช้งาน bind เพื่อส่ง Argument และการ Secure Server Actions
- Form Handling with useActionState: (New React Hook) การจัดการ Form State และ Validation Errors จาก Server แบบมืออาชีพ
- Zod Integration: การทำ Schema Validation ทั้งฝั่ง Client และ Server ให้ตรงกัน
- Optimistic UI (useOptimistic): เทคนิคทำให้หน้าเว็บตอบสนองทันทีโดยไม่ต้องรอ Server (เช่น กด Like แล้วขึ้นทันที ถ้า Error ค่อยเด้งกลับ)

##### Module 3: Authentication & Authorization (Auth.js v5)

- Setup Auth.js (NextAuth v5): การติดตั้งและ Config แบบล่าสุด
- Credentials & OAuth Provider: การทำ Login ด้วย Email/Password และ Google/GitHub
- Database Adapter: การเชื่อมต่อ Auth.js เข้ากับ Database (Prisma/Drizzle) เพื่อเก็บ User Session
- Session Management: การเข้าถึง Session ทั้งฝั่ง Client (useSession) และ Server (auth())
- Role-Based Access Control (RBAC): การป้องกัน Route และ Component ให้เข้าได้เฉพาะ Admin หรือ User ที่มีสิทธิ์



#### Module 4: Advanced UI Patterns with Shadcn/UI

- Server-Side Data Table: สร้างตารางข้อมูล (Data Table) ที่รองรับ Pagination, Sorting, Filtering ที่ประมวลผลผ่าน URL Search Params (ไม่ใช่ Client state)
- Reusable Components: การสร้าง Custom Component ขั้นสูงจาก Shadcn Primitives
- Dark Mode & Theming Deep Dive: การจัดการ Theme ขั้นสูง
- Toaster & Error Handling: การแสดงผลแจ้งเตือนจาก Server Actions ผ่าน Toast

#### Module 5: State Management & URL State

- Why not just useState?: ปัญหาของ Client State และทำไมต้องใช้ URL State
- URL as State Source of Truth: การใช้ Library nuqs (Next.js URL Query States) เพื่อจัดการ Filter/Search ใน URL
- Server State vs Client State: เส้นแบ่งที่ชัดเจนว่าข้อมูลไหนควรอยู่บน Server ข้อมูลไหนควรอยู่บน Client
- Zustand (Optional): การใช้ Global State สำหรับข้อมูลที่ไม่เกี่ยวกับ Server (เช่น Sidebar Open/Close, Complex UI Config)

#### Module 6: Performance & Caching Strategy

- Next.js Caching System: เข้าใจ Request Memoization, Data Cache, Full Route Cache
- unstable\_cache & Tag Revalidation: การคุม Cache ระดับฟังก์ชันและการสั่น Cache เฉพาะจุด
- Partial Prerendering (PPR): (New in v16) การผสม Static Shell กับ Dynamic Content ให้โหลดเร็วที่สุด
- Image & Font Optimization: เทคนิคการลด LCP (Largest Contentful Paint)

#### Module 7: Testing & Monitoring (Quality Assurance)

- Unit Testing: การเขียน Test Server Actions และ Utility Functions ด้วย Vitest
- Integration Testing: การทดสอบ Component
- E2E Testing: การใช้ Playwright จำลองการกดใช้งานเว็บจริง
- Error Tracking: การติดตั้ง Sentry เพื่อดักจับ Error บน Production (Concept)

#### Module 8: Deployment & DevOps

- Environment Variables: การจัดการ .env สำหรับ Dev/Staging/Production
- Build Optimization: การวิเคราะห์ Bundle Size
- Dockerizing Next.js: การเขียน Dockerfile แบบ Multi-stage build สำหรับ Next.js (Standalone output)
- CI/CD Pipeline (GitHub Actions): เบื้องต้นการทำ Automation Build & Deploy