



สถาบันไอทีจีเนียส
IT GENIUS INSTITUTE

COURSE ID

WDC-70

COURSE OUTLINE · รายละเอียดหลักสูตรอบรม

WEB DEVELOPMENT / REACT

Next.js Advanced

สร้างเว็บแอปขนาดใหญ่ด้วย Next.js 16 ให้ปลอดภัย เร็ว และพร้อมขึ้นระบบจริง

18 ชั่วโมง

3 วัน

พื้นฐาน Next.js App Router และ React Hooks

ระดับ Advanced



ระยะเวลา: 3 วัน

18 ชั่วโมง (3 วัน วันละ 6 ชม.) · 09:00-16:00 น.



รูปแบบ: Classroom / Hybrid

Onsite หรือ Online ผ่าน Microsoft Teams · Public / In-house



รายละเอียดหลักสูตร

www.itgenius.co.th

Next.js เป็นเฟรมเวิร์ก React ที่ทีมพัฒนาเว็บเลือกใช้สร้างระบบจริงมากที่สุดตัวหนึ่งในรุ่น 16 ระบบเปลี่ยนไปมาก ทั้ง Proxy ที่มาแทน Middleware, Turbopack ที่เป็นค่าเริ่มต้น และ Cache Components ที่ทำให้การ Cache เป็นแบบเลือกเปิดเองด้วย use cache ซึ่งทีม Next.js แนะนำให้ใช้กับทุกแอปตั้งแต่รุ่น 16.4 นักพัฒนาที่เคยเขียนแบบเดิมจึงต้องปรับวิธีคิดเรื่องการดึงข้อมูล การ Cache และการแยกงานระหว่าง Server กับ Client

หลักสูตรนี้ต่อยอดจากพื้นฐาน Next.js สู่ระดับที่ใช้สร้างระบบขนาดใหญ่ เริ่มจากโครงสร้างโปรเจกต์และ Routing ขั้นสูง จัดการฟอร์มและข้อมูลด้วย Server Actions, Zod และ Optimistic UI ทำระบบล็อกอินและกำหนดสิทธิ์ด้วย Better Auth เชื่อมฐานข้อมูลด้วย ORM สร้างตารางข้อมูลที่ประมวลผลฝั่ง Server แยก URL State กับ Global State ให้ถูกที่ ปรับความเร็วด้วย Cache Components แล้วเขียน Test และ Deploy ด้วย Docker กับ GitHub Actions ปิดท้ายด้วย Workshop สร้างระบบ Dashboard ครบวงจร (อบรม 3 วัน วันละ 6 ชั่วโมง รวม 18 ชั่วโมง ระดับ Advanced)

เครื่องมือ

N Next.js 16

R React 19

TS TypeScript

TW Tailwind CSS 4

PW Playwright

ราคาค่าอบรม · PUBLIC / IN-HOUSE

9,500 บาท / ท่าน

สมัครผ่านเว็บไซต์ ลด 10%

ราคายังไม่รวมภาษีมูลค่าเพิ่ม (VAT 7%) · รวมเอกสารประกอบการอบรม Lab Guide ซอร์สโค้ดตัวอย่างครบทุก Lab ไฟล์ Dockerfile และ Workflow สำหรับ Deploy และ Certificate · รับจัด In-house ขอใบเสนอราคาได้



เพิ่มเพื่อน
LINE

☎ 02-570-8449

☎ 088-807-9770

LINE @itgenius

🌐 www.itgenius.co.th

01 วัตถุประสงค์ Objectives

- ออกแบบโครงสร้างโปรเจกต์ขนาดใหญ่และใช้ Parallel Routes, Intercepting Routes และ Route Groups ได้
- แยกหน้าที่ Server Component กับ Client Component และเขียน Proxy ได้อย่างเหมาะสม
- จัดการฟอร์มและการแก้ไขข้อมูลด้วย Server Actions, useActionState, Zod และ useOptimistic
- ทำระบบล็อกอินและกำหนดสิทธิ์ตามบทบาท (RBAC) ด้วย Better Auth อย่างปลอดภัย
- จัดการ URL State และ Global State ได้ถูกต้องด้วย nuqs และ Zustand
- ใช้ Cache Components, use cache, cacheTag และ updateTag ปรับความเร็วของหน้าเว็บ
- เขียน Unit Test และ E2E Test ด้วย Vitest และ Playwright
- Deploy ด้วย Docker แบบ Standalone และทำ CI/CD ด้วย GitHub Actions

02 กลุ่มเป้าหมาย Target Audience

- Web Developer ที่ใช้ Next.js พื้นฐานได้แล้วและต้องการต่อยอดสู่ระดับ Senior หรือ Tech Lead
- Full-stack Developer ที่ต้องการความเชี่ยวชาญด้าน React และ Next.js รุ่นปัจจุบัน
- ทีมที่ดูแลแอป Next.js รุ่นเก่าและต้องการย้ายมาใช้แนวทางของ Next.js 16
- ผู้ที่ผ่านหลักสูตร Basic Next.js สำหรับผู้เริ่มต้นมาแล้ว

03 จุดเด่นของหลักสูตร Highlights

- อัปเดตเป็น **Next.js 16.4** Cache Components, Proxy และ React 19.3
- **Routing ขั้นสูง** Parallel, Intercepting Routes และ Route Groups
- **ฟอร์มและข้อมูล** Server Actions, Zod และ Optimistic UI
- **ล็อกอินและสิทธิ์** Better Auth พร้อม RBAC
- **พร้อมขึ้นระบบจริง** Vitest, Playwright, Docker และ CI/CD

04 ความรู้พื้นฐาน Prerequisites

- เข้าใจ Next.js App Router พื้นฐาน เช่น โครงสร้างไฟล์, Page, Layout และ Link
- เขียน React ด้วย Hooks ได้ เช่น useState, useEffect และ useTransition
- เขียน TypeScript และ Tailwind CSS ได้เบื้องต้น
- เข้าใจพื้นฐาน HTTP, API และฐานข้อมูล SQL
- โน้ตบุ๊กที่ติดตั้ง Node.js 24 LTS, Docker Desktop และ VS Code ได้ พร้อมบัญชี GitHub ของตนเอง

05 สิ่งที่ต้องเตรียมและเครื่องมือ Setup & Tools

รายการ	รายละเอียด
อุปกรณ์	Notebook Windows, macOS หรือ Linux เชื่อมต่ออินเทอร์เน็ต ติดตั้ง Docker Desktop ได้
เฟรมเวิร์ก	Next.js 16.4, React 19.3, TypeScript และ Node.js 24 LTS
UI และ State	Tailwind CSS 4, shadcn/ui, nuqs และ Zustand
ข้อมูลและ Auth	PostgreSQL uu Docker, Drizzle ORM หรือ Prisma, Zod และ Better Auth
ทดสอบและ Deploy	Vitest, Testing Library, Playwright, Docker และ GitHub Actions

หมายเหตุ: หลักสูตรนี้ปรับปรุงใหม่ปี 2026 ให้ตรงกับ Next.js 16.4 ที่แนะนำ Cache Components สำหรับทุกแอป ใช้ Proxy แทน Middleware และเปลี่ยนตัวอย่างระบบล็อกอินจาก Auth.js (NextAuth v5) เป็น Better Auth ซึ่งเป็นแนวทางที่นิยมสำหรับโปรเจกต์ใหม่ โดยยังอธิบายความต่างสำหรับทีมที่ใช้ Auth.js อยู่ เพิ่มหัวข้อ Server Component กับ Client Component, ชับข้อมูลด้วย ORM และ CI/CD ทุกเครื่องมือใช้ได้ฟรี ผู้เรียนใช้บัญชี GitHub ของตนเอง ส่วนบริการ Hosting หรือ Error Tracking แบบเสียเงินเป็นทางเลือก ไม่บังคับ เนื้อหาอ้างอิงรุ่นล่าสุดในวันอบรม

06 ภาพรวมการเรียนรู้ 3 วัน Big Picture

วัน	หัวข้อหลัก
Day 1 Section 1-5	จัดโครงสร้างโปรเจกต์ขนาดใหญ่, Routing ขั้นสูงด้วย Parallel และ Intercepting Routes, Proxy, Server Component กับ Client Component และ Server Actions คู่กับ useActionState, Zod และ Optimistic UI
Day 2 Section 6-10	ระบบล็อกอินและ RBAC ด้วย Better Auth, ชับข้อมูลด้วย Drizzle หรือ Prisma, ตารางข้อมูลฝั่ง Server ด้วย shadcn/ui, URL State ด้วย nuqs และ Global State ด้วย Zustand
Day 3 Section 11-15	Cache Components ด้วย use cache, cacheLife และ updateTag, ปรับ Performance และ Core Web Vitals, Testing ด้วย Vitest และ Playwright, Deploy ด้วย Docker และ GitHub Actions แล้วปิดท้ายด้วย Workshop

DAY
1

สถาปัตยกรรม Routing และ Server Actions

เนื้อหาการอบรม · Section 1-5

รวม
6 ชั่วโมง

1 Lab: โครงสร้างโปรเจกต์สำหรับระบบขนาดใหญ่

- สิ่งที่เปลี่ยนใน Next.js 16: Turbopack เป็นค่าเริ่มต้น, Proxy และ Cache Components
- จัดไฟล์เดอร์แบบ Feature-based เทียบกับ Type-based และ Private Folder
- ตั้งค่า TypeScript แบบเข้มงวด, Path Alias, ESLint และ Prettier
- Typed Routes และการจัดการ Environment Variables อย่างปลอดภัย
- Lab: สร้างโปรเจกต์ Dashboard ตั้งต้นด้วย create-next-app

2 Routing ขั้นสูง

- Route Groups และหลาย Layout สำหรับผู้ใช้ต่างบทบาท เช่น Admin กับ User
- Parallel Routes (@folder) แสดงหลาย Panel ที่โหลดแยกกันในหน้าเดียว
- Intercepting Routes ทำ Modal ที่มี URL ของตัวเองและเปิดเป็นหน้าต่างเต็มเมื่อ Refresh
- loading, error, not-found และ Streaming ด้วย Suspense
- Lab: หน้า Gallery ที่เปิดรูปเป็น Modal และ Dashboard หลาย Panel

3 Server Component, Client Component และ Proxy

- เส้นแบ่ง use client และการส่ง Props ข้ามขอบเขต Server กับ Client
- ดึงข้อมูลใน Server Component และป้องกันโค้ดฝั่ง Server รั่วไป Client
- Proxy (proxy.ts ที่มาแทน middleware.ts): Redirect, Rewrite และจัดการ Header
- Route Handler สำหรับ Webhook และ API ภายนอก
- Lab: Proxy ตรวจสอบ Cookie แล้วพาผู้ใช้ที่ยังไม่ล็อกอินไปหน้า Login

4 Lab: Server Actions และฟอร์ม

- Server Actions เชิงลึก: ส่ง Argument ด้วย bind และตรวจสอบสิทธิ์ทุกครั้ง
- useActionState จัดการสถานะฟอร์มและ Error จาก Server
- Zod 4 ตรวจสอบข้อมูลด้วย Schema เดียวกันทั้งฝั่ง Client และ Server
- useFormStatus และ Pending State ของปุ่ม
- Lab: ฟอร์มเพิ่มและแก้ไขข้อมูลพร้อมแสดง Error ที่ละเอียด

5 Optimistic UI และประสบการณ์ผู้ใช้

- useOptimistic ทำให้หน้าเว็บตอบสนองทันทีและย้อนกลับเมื่อเกิด Error
- useTransition สำหรับงานที่ไม่ต้องบล็อกหน้าจอ
- แจ้งผลจาก Server Actions ด้วย Toast และจัดการ Error อย่างเป็นระบบ
- View Transitions ใน React 19.3 สำหรับการเปลี่ยนหน้าที่สลับไหล
- Lab: ปุ่ม Like และรายการ Todo แบบ Optimistic

DAY
2

Authentication ข้อมูล และ State

เนื้อหาการอบรม · Section 6-10

รวม
6 ชั่วโมง

6 Lab: ชั้นข้อมูลด้วย ORM

- รับ PostgreSQL ด้วย Docker และออกแบบ Schema
- Drizzle ORM หรือ Prisma: Schema, Migration และ Seed ข้อมูล
- แยก Data Access Layer และใช้ server-only ป้องกันการเรียกจาก Client
- Transaction และการป้องกัน N+1 Query
- Lab: ชั้นข้อมูลของระบบ Dashboard ที่ใช้ต่อทั้งหลักสูตร

7 Lab: Authentication ด้วย Better Auth

- ภาพรวมทางเลือกระบบล็อกอินของ Next.js และเหตุผลที่เลือก Better Auth
- ติดตั้ง Better Auth, ต่อฐานข้อมูล และ Route Handler สำหรับ /api/auth
- ล็อกอินด้วย Email และ Password และ OAuth กับ Google หรือ GitHub
- อ่าน Session ใน Server Component, Server Actions และฝั่ง Client
- Lab: หน้า Sign up, Sign in และ Sign out ที่ใช้งานได้จริง

8 Authorization และ Role-Based Access Control

- กำหนด Role และ Permission ด้วย Admin Plugin ของ Better Auth
- ป้องกัน Route, Component และ Server Actions ตามสิทธิ์
- ตรวจสอบสิทธิ์ใกล้ข้อมูลเสมอ ไม่พึ่ง Proxy อย่างเดียว
- ความต่างเมื่อทีมใช้ Auth.js อยู่แล้ว และแนวทางย้ายระบบ
- Lab: หน้า Admin ที่เข้าได้เฉพาะผู้มีสิทธิ์

9 Lab: Data Table ฝั่ง Server ด้วย shadcn/ui

- ตั้งค่า shadcn/ui กับ Tailwind CSS 4 และการปรับ Theme รวมถึง Dark Mode
- Data Table ที่แบ่งหน้า เรียงลำดับ และกรองข้อมูลบน Server
- สร้าง Component ที่ใช้ซ้ำได้จาก Primitive ของ shadcn/ui
- φόρμใน Dialog ที่ทำงานร่วมกับ Server Actions
- Lab: หน้าจัดการผู้ใช้และคำสั่งซื้อพร้อมค้นหาและกรอง

10 URL State และ Global State

- ทำไมไม่ใช้ useState ทุกอย่าง: แยก Server State, URL State และ Client State
- nuqs เก็บ Filter, Search และหน้าปัจจุบันไว้ใน URL แบบ Type-safe
- Zustand สำหรับสถานะ UI เช่น Sidebar และค่าตั้งค่าที่ซับซ้อน
- หลีกเลี่ยงการซ้ำข้อมูลระหว่าง Server กับ Client Store
- Lab: ตารางที่แชร์ลิงก์พร้อม Filter ได้



DAY
3

Performance, Testing และ Deploy

เนื้อหาการอบรม · Section 11-15

รวม
6 ชั่วโมง

11 Lab: Cache Components

- เปิด cacheComponents และเข้าใจโมเดลการ Cache แบบเลือกเปิดเอง
- use cache, cacheLife และ cacheTag ระดับฟังก์ชันและ Component
- updateTag และ revalidateTag ล้าง Cache เฉพาะจุดหลังแก้ไขข้อมูล
- Static Shell คู่กับ Dynamic Content ใน Suspense และ use cache: private
- Lab: Dashboard ที่โหลดเร็วแต่ข้อมูลส่วนตัวยังถูกต้อง

12 Performance และ Core Web Vitals

- วัด LCP, INP และ CLS และอ่านผลจาก Lighthouse
- next/image, next/font และการโหลด Script ภายนอก
- Prefetch ของ Link และการเลื่อนงานที่ไม่จำเป็นออกไป
- วิเคราะห์ขนาด Bundle และรู้จัก React Compiler
- Lab: ปรับหน้าแรกให้คะแนน Core Web Vitals ดีขึ้น

13 Lab: Testing

- Unit Test ด้วย Vitest สำหรับ Utility, Schema และ Server Actions
- ทดสอบ Component ด้วย Testing Library
- E2E Test ด้วย Playwright จำลองการคลิกอื่นและใช้งานจริง
- แนวคิด Error Tracking และ Logging บน Production
- Lab: ชด Test ของฟอร์มและหน้า Admin

14 Lab: Deploy ด้วย Docker และ CI/CD

- จัดการ Environment Variables สำหรับ Dev, Staging และ Production
- output standalone และ Dockerfile แบบ Multi-stage
- รันแอปคู่ PostgreSQL ด้วย Docker Compose
- GitHub Actions: Lint, Test, Build และสร้าง Image อัตโนมัติ
- ทางเลือก Hosting: Server ของตนเองหรือ Platform แบบ Managed

15 Workshop: ระบบ Dashboard ครบวงจร

- โจทย์: ระบบจัดการคำสั่งซื้อที่มีผู้ใช้หลายบทบาท
- ออกแบบ Route, ชั้นข้อมูล และสิทธิ์ แล้วลงมือสร้างด้วยแนวทางที่เรียนมา
- ใส่ Cache, Test และ Pipeline ให้พร้อม Deploy
- นำเสนอผลงาน ใช้ AI Assistant ช่วยเขียนโค้ดอย่างมีการตรวจทาน และสรุปแนวทางต่อยอด

07 เรียนจบแล้วได้อะไร Outcomes

- วางโครงสร้างระบบใหญ่ได้ Routing ขั้นสูงและ Proxy
- จัดการฟอร์มอย่างมืออาชีพ Server Actions, Zod และ Optimistic UI
- ทำระบบล็อกอินได้ Better Auth และ RBAC
- จัดการ State ถูกที่ Server, URL และ Client State
- ทำเว็บให้เร็ว Cache Components และ Core Web Vitals
- นำขึ้นระบบจริงได้ Testing, Docker และ GitHub Actions

08 สิ่งที่คุณเรียนจะได้รับ Deliverables

- เอกสารประกอบการอบรมฉบับสมบูรณ์ครบทั้ง 3 วัน และ Lab Guide ทีละขั้นตอน
- ซอร์สโค้ดของระบบ Dashboard ตัวอย่างและ Workshop ครบทุก Lab
- Dockerfile, ไฟล์ docker-compose และ Workflow ของ GitHub Actions สำหรับนำไปปรับใช้
- ประกาศนียบัตรจบหลักสูตร (Certificate of Completion) และช่องทางสอบถามวิทยากรหลังจบหลักสูตร

09 ข้อมูลผู้สอน Instructor

อ.สามิตร โทยม ผู้สอนและนักพัฒนา Full-stack ที่ใช้ Next.js และ React สร้างระบบจริงให้องค์กร จุดเด่นคือการสอนจากโปรเจกต์ Dashboard ชุดเดียวที่ต่อเติมตลอดหลักสูตร อธิบายเหตุผลเบื้องหลังแต่ละแนวทางว่าทำไมจึงปลอดภัยและเร็วกว่า และพาผู้เรียนลงมือทำจนนำระบบขึ้น Production ได้ด้วยตนเอง

ราคาค่าอบรม 9,500 บาท/ท่าน · สนใจ IN-HOUSE ขอใบเสนอราคา

9,500 บาท / ท่าน (ยังไม่รวม VAT 7%)

สมัครผ่านเว็บไซต์ ลด 10%

☎ 02-570-8449

📞 @itgenius

☎ 088-807-9770

✉ contact@itgenius.co.th



เพิ่มเพื่อน
LINE