



ไอทีจีเนียส เอ็นจิเนียริ่ง

IT GENIUS ENGINEERING CO., LTD.

COURSE ID

NEXT-16

COURSE OUTLINE · รายละเอียดหลักสูตรอบรม

WEB DEVELOPMENT

พัฒนา Full-Stack Web Application ด้วย Next.js 16 และ Prisma ORM 7

30 ชั่วโมง

5 วัน

ระดับ Basic to Advanced



ระยะเวลา: 5 วัน

30 ชั่วโมง · 09:00 - 16:00 น.



รูปแบบ: Live Online

อบรมสดผ่าน Zoom · Hands-on Workshop



วิทยากร

อาจารย์สมิทธิ์ ไทยม

Next.js ได้กลายเป็นเฟรมเวิร์กมาตรฐานของการพัฒนาเว็บแอปพลิเคชันด้วย React ในระดับองค์กร และในปี 2026 นี้ Next.js ก้าวเข้าสู่เวอร์ชัน 16 ซึ่งเป็นการเปลี่ยนแปลงครั้งใหญ่ที่สุดนับตั้งแต่การมาของ App Router ทั้ง Turbopack ที่กลายเป็น Bundler มาตรฐาน, Cache Components ที่พลิกโมเดลการแคชทั้งหมด, การเปลี่ยนจาก middleware.ts มาเป็น proxy.ts และการปรับ Request API ทั้งหมดให้เป็น Asynchronous

หลักสูตรนี้ออกแบบมาสำหรับนักพัฒนาที่มีพื้นฐาน React มาแล้ว โดยปูพื้นฐาน App Router และ React Server Components ให้แน่น แล้วเจาะลึกหัวใจของหลักสูตรคือการออกแบบและพัฒนา API Layer ที่เชื่อมต่อฐานข้อมูลด้วย Prisma ORM 7 พร้อมเปรียบเทียบการใช้งานจริงระหว่าง PostgreSQL และ MariaDB ปิดท้ายด้วยเนื้อหาาระดับ Production ทั้ง Authentication, RBAC, Testing, Caching, Docker Deployment, CI/CD และ Capstone Project

TECH STACK



Next.js 16



React 19



TypeScript



Prisma 7



PostgreSQL 17



MariaDB 11



Node.js 22



Docker

ราคาคอร์สอบรม · PUBLIC / IN-HOUSE TRAINING

ขอใบเสนอราคา

กำหนดวันและจำนวนผู้เรียนได้ตามตกลง · สอบถามรอบอบรมล่าสุดกับเจ้าหน้าที่



เพิ่มเพื่อน
LINE

☎ 02-570-8449

☎ 088-807-9770

💬 @itgenius

🌐 www.itgenius.co.th

ข้อมูลหลักสูตร

COURSE INFORMATION

ชื่อหลักสูตร	Basic to Advanced Next.js 2026 (Next.js 16 + Prisma 7 + PostgreSQL / MariaDB)
ระยะเวลาอบรม	5 วัน (30 ชั่วโมง)
รูปแบบการอบรม	อบรมออนไลน์ผ่าน Zoom แบบสด (Live Online) บรรยายพร้อมปฏิบัติจริง
เวลาอบรม	09:00 - 16:00 น. (พักกลางวัน 12:00 - 13:00 น. พักย่อยช่วงเช้าและบ่าย)
วันที่อบรม	รอกำหนดการ (สอบถามรอบอบรมล่าสุดกับเจ้าหน้าที่)
ราคาค่าอบรม	ขอใบเสนอราคา
วิทยากรผู้สอน	อาจารย์สามิตร โกยม
ระดับหลักสูตร	Basic to Advanced (เริ่มจากพื้นฐาน Next.js ไปจนถึงระดับ Production)
คอร์สที่ควรอบรมก่อน	พื้นฐาน React 19 และ TypeScript
เอกสารประกอบ	ไฟล์เอกสารประกอบการอบรม, Source Code ตัวอย่าง และ Starter Project
เวอร์ชันที่ใช้สอน	Next.js 16.3, React 19 (Canary ที่มาพร้อม Next.js), Prisma ORM 7.9, Better Auth 1.6, Node.js 22 LTS

เกี่ยวกับหลักสูตร

ABOUT THIS COURSE

Next.js ได้กลายเป็นเฟรมเวิร์กมาตรฐานของการพัฒนาเว็บแอปพลิเคชันด้วย React ในระดับองค์กร และในปี 2026 นี้ Next.js ได้ก้าวเข้าสู่เวอร์ชัน 16 ซึ่งเป็นการเปลี่ยนแปลงครั้งใหญ่ที่สุดนับตั้งแต่การมาของ App Router ไม่ว่าจะเป็นการที่ Turbopack กลายเป็น Bundler มาตรฐานทั้งตอน Development และ Build, การมาของ Cache Components ที่พลิกโมเดลการแคชทั้งหมดให้ชัดเจนและควบคุมได้, การเปลี่ยนจาก middleware.ts มาเป็น proxy.ts รวมถึงการปรับ Request API ทั้งหมดให้เป็น Asynchronous

ในขณะเดียวกัน ฝั่งการจัดการฐานข้อมูล Prisma ORM ก็เข้าสู่เวอร์ชัน 7 ที่ยกเลิก Rust Query Engine ออกไปทั้งหมด เปลี่ยนมาใช้ Query Compiler แบบ WebAssembly กำหนดให้ต้องใช้ Driver Adapter เสมอ และย้ายการตั้งค่าออกมาไว้ที่ prisma.config.ts การเปลี่ยนแปลงเหล่านี้ส่งผลโดยตรงต่อวิธีเขียนโค้ดเชื่อมต่อฐานข้อมูลในโปรเจกต์ Next.js ทุกโปรเจกต์

หลักสูตรนี้ออกแบบมาสำหรับนักพัฒนาที่มีพื้นฐาน React มาแล้วแต่ยังไม่เคยใช้งาน Next.js โดยจะปูพื้นฐาน App Router และ React Server Components ให้แน่นในวันแรก จากนั้นจะเจาะลึกไปที่หัวใจของหลักสูตรคือ **การออกแบบและพัฒนา API Layer ที่เชื่อมต่อฐานข้อมูลด้วย Prisma** ทั้งการใช้งาน Route Handlers, Server Actions, การออกแบบ Schema, การทำ Migration, การจัดการ Transaction, การปรับจูนประสิทธิภาพของ Query รวมถึงการเปรียบเทียบการทำงานจริงระหว่าง PostgreSQL และ MariaDB พร้อมข้อควรระวังที่พบได้จริงในการใช้งาน Prisma 7 กับ MariaDB

ปิดท้ายด้วยเนื้อหาระดับ Production ทั้งระบบ Authentication และ Authorization แบบ RBAC ด้วย Better Auth, การทำ Testing ด้วย Vitest และ Playwright, การปรับแต่งประสิทธิภาพด้วย Cache Components และ Partial Prerendering ไปจนถึงการ Deploy ด้วย Docker แบบ Self-hosted และการวางกระบวนการ CI/CD และจบด้วย Capstone Project ที่ผู้เรียนจะได้สร้างระบบงานจริงตั้งแต่ต้นจนจบ

หมายเหตุ การอบรมเป็นรูปแบบบรรยายพร้อมปฏิบัติจริง (Hands-on Workshop) ตลอดหลักสูตร ผู้เรียนจะได้ลงมือสร้างโปรเจกต์เดียวกันต่อเนื่องทั้ง 5 วัน เนื้อหาอ้างอิง Next.js 16.3 และ Prisma ORM 7.9 ซึ่งเป็นเวอร์ชันเสถียรล่าสุด ณ เดือนสิงหาคม 2026

01 วัตถุประสงค์ของหลักสูตร Objectives

- เข้าใจสถาปัตยกรรมของ Next.js 16 และความแตกต่างระหว่าง App Router กับ Pages Router
- เข้าใจแนวคิด React Server Components, Client Components และ Streaming ได้อย่างถูกต้อง
- ออกแบบระบบ Routing, Layout, Loading State และ Error Handling ด้วย App Router ได้
- สร้าง API Layer ด้วย Route Handlers และ Server Actions ได้อย่างถูกต้องและปลอดภัย
- ออกแบบ Database Schema ด้วย Prisma Schema Language ได้อย่างเป็นระบบ
- ติดตั้งและใช้งาน Prisma ORM 7 ร่วมกับ Driver Adapter สำหรับ PostgreSQL และ MariaDB ได้
- บริหารจัดการโครงสร้างฐานข้อมูลด้วย Prisma Migrate ได้ตลอดวงจร Development และ Production
- เขียน Query ที่ซับซ้อนด้วย Prisma Client ทั้ง Relation, Aggregation, Transaction และ Raw SQL ได้
- เข้าใจความแตกต่างเชิงปฏิบัติระหว่าง PostgreSQL และ MariaDB เมื่อใช้งานผ่าน Prisma พร้อมข้อควรระวัง
- ตรวจสอบและปรับปรุงประสิทธิภาพของ Query รวมถึงแก้ปัญหา N+1 และการจัดการ Connection Pool ได้
- พัฒนาระบบ Authentication และ Authorization แบบ RBAC ที่ปลอดภัยด้วย Better Auth ได้
- เขียน Test ครอบคลุมทั้ง Unit, Integration และ End-to-End ด้วย Vitest และ Playwright ได้
- ใช้งานระบบแคชของ Next.js 16 ทั้ง Cache Components, use cache, cacheTag และ Partial Prerendering ได้
- Deploy แอปพลิเคชันแบบ Self-hosted ด้วย Docker และวางกระบวนการ CI/CD ได้
- นำความรู้ทั้งหมดมาประกอบเป็นระบบงานจริงได้ครบวงจรผ่าน Capstone Project

02 กลุ่มเป้าหมาย Target Audience

- นักพัฒนา Frontend ที่มีพื้นฐาน React และต้องการก้าวสู่การพัฒนาแบบ Full-Stack
- นักพัฒนา Full-Stack ที่ต้องการอัปเดตความรู้ Next.js ให้ทันเวอร์ชัน 16 และ Prisma 7
- นักพัฒนา Backend ที่ต้องการเข้าใจการออกแบบ API Layer บน Next.js
- Technical Lead และ Solution Architect ที่ต้องการวางมาตรฐานโครงสร้างโปรเจกต์ Next.js ในองค์กร
- นักศึกษาและผู้สนใจที่มีพื้นฐาน React และต้องการพัฒนาเว็บแอปพลิเคชันระดับ Production

03 ความรู้พื้นฐานที่ผู้เข้าอบรมควรมี Prerequisites

- **สำคัญ** มีพื้นฐานการพัฒนาด้วย React (Component, Props, State, Hooks) มาก่อน
- มีพื้นฐาน JavaScript สมัยใหม่ (ES2020+) ได้แก่ async/await, Destructuring, Module และ Array Method
- มีพื้นฐาน TypeScript ระดับใช้งานได้ (Type, Interface, Generic เบื้องต้น)
- เข้าใจการทำงานของ HTTP, REST API และ JSON
- เข้าใจฐานข้อมูลเชิงสัมพันธ์และคำสั่ง SQL เบื้องต้น (SELECT, INSERT, UPDATE, DELETE, JOIN)
- สามารถใช้งาน Command Line, Git และ npm/pnpm เบื้องต้นได้
- เคยใช้งาน Docker เบื้องต้น (จะช่วยให้เรียนวันที่ 5 ได้ราบรื่นขึ้น แต่ไม่บังคับ)

04 เครื่องมือและเทคโนโลยีที่ใช้ Tools & Technologies

- **Next.js 16.3** เฟรมเวิร์กหลักของหลักสูตร (App Router, Turbopack, Cache Components)
- **React 19** ผ่าน React Canary ที่มาพร้อม Next.js 16 (Server Components, Actions, use hook)
- **TypeScript 5.9+** ภาษาหลักที่ใช้ตลอดหลักสูตร
- **Node.js 22 LTS** Runtime (Next.js 16 ต้องการ Node.js เวอร์ชัน 20.9.0 ขึ้นไป)
- **Prisma ORM 7.9** ORM สำหรับจัดการฐานข้อมูลและ Migration
- **PostgreSQL 17** ฐานข้อมูลหลักของหลักสูตร (ผ่าน @prisma/adapters-pg)
- **MariaDB 11** ฐานข้อมูลสำหรับหัวข้อเปรียบเทียบและ Lab เสริม (ผ่าน @prisma/adapters-mariadb)
- **Tailwind CSS 4 + shadcn/ui** สำหรับงานส่วนติดต่อผู้ใช้
- **Zod** สำหรับ Schema Validation ทั้งฝั่ง Server Action และ Route Handler
- **Better Auth** สำหรับระบบ Authentication, Session, 2FA และ Access Control
- **Vitest + Testing Library + Playwright** สำหรับ Unit, Integration และ E2E Testing
- **Docker และ Docker Compose** สำหรับสภาพแวดล้อมฐานข้อมูลและการ Deploy
- **GitHub Actions** สำหรับกระบวนการ CI/CD
- **Prisma Studio และ pgAdmin / DBeaver** สำหรับตรวจสอบข้อมูลในฐานข้อมูล

05 ภาพรวมหลักสูตร 5 วัน Course at a Glance

DAY 1	ปูพื้นฐาน Next.js 16, App Router, React Server Components และระบบ Routing
DAY 2	Data Fetching, Route Handlers (REST API), Server Actions และการเชื่อมต่อ Prisma กับ PostgreSQL ครั้งแรก
DAY 3	Prisma เชิงลึก: Schema Design, Migrate, Relation, Transaction, Performance และเปรียบเทียบ MariaDB
DAY 4	API ระดับ Production: Validation, Error Handling, Authentication, RBAC, File Upload และ Testing
DAY 5	Caching เชิงลึก, Performance Optimization, Docker Deployment, CI/CD และ Capstone Project



DAY
1

ปูพื้นฐาน Next.js 16, App Router และ React Server Components

เนื้อหาการอบรม · Module 1.1 – 1.5

รวม
6 ชั่วโมง

เริ่มจากศูนย์สู่ความเข้าใจสถาปัตยกรรมของ Next.js 16 สร้างโปรเจกต์แรก วางโครงสร้าง Routing และเข้าใจความต่างระหว่าง Server และ Client Component อย่างถ่องแท้

1.1 ทำความรู้จัก Next.js และภาพรวมเวอร์ชัน 16

- Next.js คืออะไร และแก้ปัญหอะไรให้กับนักพัฒนา React
- ความแตกต่างระหว่าง CSR, SSR, SSG, ISR และ Streaming SSR
- ภาพรวมพัฒนาการของ Next.js: จาก Pages Router สู่ App Router
- สิ่งใหม่ที่สำคัญใน Next.js 16: Turbopack เป็นค่ามาตรฐาน, Cache Components, proxy.ts, Async Request APIs
- ข้อกำหนดของระบบ: Node.js 20.9+, TypeScript 5.1+ และเบราว์เซอร์ที่รองรับ
- สิ่งที่ถูกถอดออกใน Next.js 16 และผลกระทบต่อโปรเจกต์เดิม (AMP, next lint, serverRuntimeConfig, Sync Params)

1.2 สร้างโปรเจกต์และทำความเข้าใจโครงสร้าง

- การสร้างโปรเจกต์ด้วย create-next-app และตัวเลือกที่ควรรู้
- โครงสร้างไดเรกทอรีมาตรฐาน: app/, public/, next.config.ts
- การตั้งค่า TypeScript, Path Alias และ ESLint Flat Config
- Turbopack: ความเร็วที่เปลี่ยนไป และการ Opt-out ด้วย --webpack
- การใช้งาน Next.js DevTools และ Dev Server แบบใหม่

• **Workshop:** สร้างโปรเจกต์ nextjs-training ที่จะใช้พัฒนาต่อเนื่องตลอดหลักสูตร

1.3 ระบบ Routing ของ App Router

- แนวคิด File-system Routing และไฟล์พิเศษ: page.tsx, layout.tsx, template.tsx
- Nested Routes และการซ้อน Layout หลายชั้น
- Dynamic Routes: [id], Catch-all [...slug] และ Optional Catch-all [...slug]
- **สำคัญ** params และ searchParams เป็น Promise แล้วใน Next.js 16 (ต้อง await)
- Route Groups (group) และ Private Folders _folder
- Parallel Routes และ Intercepting Routes พร้อมข้อบังคับใหม่เรื่องไฟล์ default.js
- การนำทางด้วย <Link>, useRouter, redirect() และ notFound()



DAY
1

ปูพื้นฐาน Next.js 16, App Router และ React Server Components

เนื้อหาการอบรม · Module 1.1 – 1.5 (ต่อ)

รวม

6 ชั่วโมง

1.4 React Server Components และ Client Components

- แนวคิด React Server Components (RSC) และเหตุผลที่เปลี่ยนวิธีการพัฒนา
- Server Component เป็นค่ามาตรฐาน: ประโยชน์ด้าน Bundle Size และความปลอดภัย
- เมื่อไรควรใช้ 'use client' และเมื่อไรไม่ควร
- ขอบเขตของ Server และ Client (Component Boundary) และการส่ง Props ข้ามฝั่ง
- รูปแบบที่ใช้บ่อย: การส่ง Server Component เป็น children เข้า Client Component
- ข้อจำกัดที่ต้องรู้: Hooks, Event Handler, Browser API และการ Serialize ข้อมูล
- การใช้ import 'server-only' และ 'client-only' เพื่อป้องกันโค้ดหลุดฝั่ง

1.5 Loading, Error, Streaming และ Metadata

- loading.tsx และการทำ Instant Loading State
- การใช้ <Suspense> เพื่อทำ Streaming แบบละเอียดในระดับ Component
- error.tsx, global-error.tsx และ not-found.tsx
- Custom Error Boundary ด้วย catchError จาก next/error (Next.js 16.3) และการใช้ retry()
- Static และ Dynamic Metadata API เพื่อทำ SEO
- การจัดการ Font ด้วย next/font และรูปภาพด้วย next/image พร้อมค่ามาตรฐานใหม่ใน v16
- **Workshop:** สร้างโครงสร้างหน้าเว็บของระบบงานตัวอย่าง พร้อม Layout, Loading, Error และ Metadata ครบทุกชั้น

DAY
2

Data Fetching, Route Handlers, Server Actions และก้าวแรกกับ Prisma + PostgreSQL

เนื้อหาการอบรม · Module 2.1 – 2.5

รวม

6

ชั่วโมง

เรียนรู้วิธีดึงและเขียนข้อมูลบน Next.js ทุกรูปแบบ สร้าง REST API ด้วย Route Handlers และเชื่อมต่อฐานข้อมูล PostgreSQL ด้วย Prisma 7 เป็นครั้งแรก

2.1 การดึงข้อมูล (Data Fetching) ใน App Router

- การ Fetch ข้อมูลใน Server Component ด้วย async/await
- พฤติกรรมของ fetch ใน Next.js 16 และการเปลี่ยนค่ามาตรฐานเรื่องแคช
- Request Memoization ด้วย React.cache()
- รูปแบบการโหลดข้อมูล: Sequential vs Parallel Data Fetching
- การจัดการ Loading และ Error ระหว่าง Fetch
- การดึงข้อมูลฝั่ง Client เมื่อจำเป็น และการเลือกใช้ SWR หรือ TanStack Query



DAY
2

Data Fetching, Route Handlers, Server Actions และท้าวแรกกับ Prisma + PostgreSQL

เนื้อหาการอบรม · Module 2.1 – 2.5 (ต่อ)

รวม

6

ชั่วโมง

2.2 Route Handlers: สร้าง REST API ใน Next.js

- ไฟล์ route.ts และการ Export ตาม HTTP Method (GET, POST, PUT, PATCH, DELETE)
- NextRequest และ NextResponse กับความสามารถที่เพิ่มจาก Web Standard
- การอ่าน Query String, Path Params (เป็น Promise แล้วใน v16) และ Request Body
- การจัดการ Header, Cookie ด้วย next/headers และการตั้งค่า CORS
- Route Segment Config: dynamic, revalidate, runtime, maxDuration
- Runtime: Node.js (ค่ามาตรฐาน) กับ Edge และเหตุผลที่ Edge Runtime ถูกลดบทบาทลง
- การทำ Streaming Response ด้วย ReadableStream
- แนวปฏิบัติที่ถูกต้อง: ไม่ควรเรียก Route Handler ของตัวเองจาก Server Component

2.3 Server Actions และการเขียนข้อมูล

- แนวคิด Server Actions และ 'use server'
- การใช้ Server Action กับ <form> แบบ Progressive Enhancement
- Hooks ที่เกี่ยวข้อง: useActionState, useFormStatus, useOptimistic
- การส่งผลลัพธ์และข้อความ Error กลับไปยัง Client
- การทำ Revalidation ด้วย revalidatePath และ revalidateTag
- **ความปลอดภัยที่ต้องรู้** Server Action เป็น Endpoint สาธารณะ ต้องตรวจสอบสิทธิ์ภายในทุกครั้ง
- เปรียบเทียบว่าเมื่อไรควรใช้ Server Action และเมื่อไรควรใช้ Route Handler

2.4 รู้จัก Prisma ORM 7 และการเปลี่ยนแปลงครั้งใหญ่

- ORM คืออะไร และทำไมต้องใช้ Prisma ในโปรเจกต์ Next.js
- องค์ประกอบของ Prisma: Schema, Client, Migrate, Studio
- **สิ่งที่เปลี่ยนใน Prisma 7** ยกเลิก Rust Engine เปลี่ยนมาใช้ Query Compiler แบบ WebAssembly
- Generator ตัวใหม่ prisma-client (แทน prisma-client-js ที่ถูก Deprecate) และการกำหนด output ที่บังคับ
- **Driver Adapter เป็นข้อบังคับแล้ว** ไม่สามารถเรียก new PrismaClient() เปล่าได้อีกต่อไป
- ไฟล์ prisma.config.ts และการย้าย url / shadowDatabaseUrl ออกจาก schema.prisma
- ข้อกำหนดใหม่: Node.js 20.19+, TypeScript 5.4+ และแนวทาง ESM
- ภาพรวม Prisma 8 (Early Access) และเหตุผลที่หลักสูตรนี้ยึด Prisma 7 เป็นหลัก



DAY
2

Data Fetching, Route Handlers, Server Actions และก้าวแรกกับ Prisma + PostgreSQL

เนื้อหาการอบรม · Module 2.1 – 2.5 (ต่อ)

รวม

6

ชั่วโมง

2.5 ติดตั้ง PostgreSQL และเชื่อมต่อ Prisma ครั้งแรก

- การเตรียม PostgreSQL 17 ด้วย Docker Compose สำหรับ Development
- การติดตั้ง Prisma และ @prisma/adapters-pg
- prisma init และการตั้งค่า prisma.config.ts ให้ถูกต้อง
- การเขียน Model แรกใน schema.prisma และการรัน prisma migrate dev
- รูปแบบ Singleton PrismaClient สำหรับ Next.js และเหตุผลที่ต้องใช้ globalThis เฉพาะตอน Development
- การตั้งค่า postinstall: prisma generate และการจัดการไพลเดอร์ Client ที่ถูก Generate
- การใช้งาน Prisma Studio เพื่อตรวจสอบข้อมูล
- **Workshop:** เชื่อมต่อโปรเจกต์เข้ากับ PostgreSQL สร้างตารางแรก และทำ CRUD API ครบชุดด้วย Route Handlers + Server Actions

DAY
3

Prisma เชิงลึก · Schema, Migration, Query, Transaction และ MariaDB

เนื้อหาการอบรม · Module 3.1 – 3.6

รวม

6 ชั่วโมง

วันที่เข้มข้นที่สุดของหลักสูตร เราจะศึกษาการออกแบบฐานข้อมูลด้วย Prisma ตั้งแต่ Schema จนถึงการใช้งานประสิทธิภาพ พร้อมทดลองใช้งานจริงกับ MariaDB และเรียนรู้ข้อควรระวังที่พบได้จริง

3.1 ออกแบบ Schema ด้วย Prisma Schema Language

- โครงสร้างไฟล์ schema.prisma และการแยกไฟล์แบบ Multi-file Schema
- Scalar Type และการ Map ไปเป็น Type จริงในฐานข้อมูล
- Attribute ที่ใช้บ่อย: @id, @unique, @default, @updatedAt, @map, @@map, @@index
- ฟังก์ชันค่าเริ่มต้น: autoincrement(), uuid(7), cuid(2), ulid(), nanoid(), dbgenerated()
- ข้อควรรู้ uuid() และ cuid() ทำงานฝั่ง Client เท่านั้น มองไม่เห็นจาก Raw SQL
- การใช้งาน Enum และข้อจำกัดของ Enum บนแต่ละฐานข้อมูล
- Native Type ของ PostgreSQL: @db.Uuid, @db.Citext, @db.JsonB, @db.TimestampTz, @db.Inet
- การออกแบบ Index ให้เหมาะกับรูปแบบการ Query

DAY
3

Prisma เชิงลึก · Schema, Migration, Query, Transaction และ MariaDB

เนื้อหาการอบรม · Module 3.1 – 3.6 (ต่อ)

รวม
6 ชั่วโมง

3.2 ความสัมพันธ์ระหว่างตาราง (Relations)

- One-to-One, One-to-Many และ Many-to-Many (ทั้งแบบ Implicit และ Explicit Join Table)
 - Self-relation และการออกแบบโครงสร้างแบบต้นไม้ (Category, Organization Chart)
 - onDelete และ onUpdate: Cascade, Restrict, SetNull, NoAction
 - Composite Key และ Composite Unique Constraint
 - relationMode: foreignKeys กับ prisma และผลกระทบเมื่อฐานข้อมูลไม่รองรับ FK
 - แนวคิด multiSchema สำหรับ PostgreSQL และเหตุผลที่ MySQL/MariaDB ใช้ไม่ได้
- **Workshop:** ออกแบบ Schema ระบบงานตัวอย่าง (User, Role, Product, Category, Order, OrderItem, Payment) แบบครบวงจร

3.3 Prisma Migrate ตลอดจนวงจรการพัฒนา

- ความแตกต่างระหว่าง migrate dev, migrate deploy, db push และควรใช้เมื่อไร
- Shadow Database คืออะไร ทำงานอย่างไร และสิทธิ์ที่ฐานข้อมูลต้องมี
- การแก้ไข Migration ด้วยมือด้วย migrate dev --create-only
- การจัดการ Drift และคำสั่ง migrate status, migrate resolve, migrate diff
- การทำ Baselining เพื่อนำ Prisma เข้าไปคุณฐานข้อมูลที่มีอยู่แล้ว
- การทำ Seed Data ด้วย prisma db seed (Prisma 7 ต้องสั่งเองเท่านั้น ไม่ทำงานอัตโนมัติแล้ว)
- Error Code ที่พบบ่อยและวิธีแก้: P3005, P3006, P3009, P3014, P3018
- แนวทางการทำ Migration บน Production อย่างปลอดภัย และการ Rollback

3.4 Prisma Client Query เชิงลึก

- CRUD คุบชุด: findUnique, findFirst, findMany, create, update, upsert, delete
- Batch Operations: createMany, updateMany, deleteMany, createManyAndReturn
- Filtering ขั้นสูง: AND, OR, NOT, in, contains, startsWith, mode: 'insensitive'
- การโหลดข้อมูลสัมพันธ์: include, select, Nested Select และการทำ Nested Write
- Pagination: Offset-based (skip/take) กับ Cursor-based และการเลือกใช้ให้เหมาะสม
- Aggregation และ Grouping: count, aggregate, groupBy, _sum, _avg
- การทำงานกับ JSON Field และข้อจำกัดของการ Filter บน JSON แต่ละฐานข้อมูล
- Full-Text Search: สถานะ Preview บน PostgreSQL กับ GA บน MySQL/MariaDB
- Raw Query: \$queryRaw, \$executeRaw และการป้องกัน SQL Injection
- TypedSQL (Preview) สำหรับ Query ที่ซับซ้อนเกินความสามารถของ Prisma Client



DAY
3

Prisma เชิงลึก · Schema, Migration, Query, Transaction และ MariaDB

เนื้อหาการอบรม · Module 3.1 – 3.6 (ต่อ)

รวม
6 ชั่วโมง

3.5 Transaction, Performance และ Connection Pool

- Sequential Transaction `$transaction([...])` กับ Interactive Transaction `$transaction(async tx => ...)`
- ตัวเลือกสำคัญ: `maxWait`, `timeout` และ `Isolation Level`
- การจัดการ Write Conflict (P2034) และรูปแบบการ Retry
- Nested Transaction ด้วย Savepoint (Prisma 7.5+)
- ปัญหา N+1 เกิดขึ้นได้อย่างไรบน Prisma และวิธีตรวจจับ
- `relationLoadStrategy` (Preview): join กับ query และผลต่อ SQL ที่ถูกสร้าง
- การเปิด Query Logging เพื่อดู SQL จริงที่ถูกส่งไปยังฐานข้อมูล
- Connection Pool ใน Prisma 7: ตั้งค่าที่ Driver Adapter แล้ว ไม่ใช่ที่ Connection String
- การใช้งาน PgBouncer, Supabase Supavisor และข้อควรระวังตอนทำ Migration ผ่าน Pooler
- Query Plan Cache และ `queryPlanCacheMaxSize` (Prisma 7.8+)
- Prisma Client Extensions สำหรับสร้าง Method และ Logic ที่ใช้ซ้ำ (แทน `$use` ที่ถูกถอดออก)

3.6 MariaDB กับ Prisma: การใช้งานจริงและข้อควรระวัง

- MariaDB ใช้ Provider `mysql` เดียวกับ MySQL และผลที่ตามมา
- การติดตั้งและตั้งค่า `@prisma/adapters-mariadb`
- ตารางเปรียบเทียบ Type Mapping PostgreSQL กับ MariaDB (`text` vs `VARCHAR(191)`, `jsonb` vs `JSON`, `boolean` vs `TINYINT(1)`)
- ฟังก์ชันที่ไม่ได้บน MariaDB: `Scalar List`, `multiSchema`, `mode: 'insensitive'`, `Partial Index`, `createManyAndReturn`
- ฟังก์ชันที่ MariaDB/MySQL ได้เปรียบ: `Full-Text Index (@@fulltext)` และ `Index length`
- ข้อควรระวังสำคัญ ปัญหาที่ยังเปิดค้างของ Prisma 7 กับ MariaDB โดยเฉพาะเรื่อง `JSON Column`, `Migration Drift` และ `Timezone`
- ข้อจำกัดของ Migration: DDL ไม่เป็น Transaction, สิทธิ์ `Shadow DB` ระดับ `Global`, `migrate reset` คือการ Drop Database
- ปัญหา `Collation Drift` และวิธีจัดการ
- เกณฑ์การเลือกฐานข้อมูลสำหรับโปรเจกต์จริง
- **Workshop เสริม:** ย้าย Schema เดิมมารันบน MariaDB สังเกตความแตกต่างของ SQL ที่ถูก Generate และแก้ปัญหาคือพบจริง

DAY
4

API ระดับ Production · Validation, Security, Authentication, RBAC และ Testing

เนื้อหาการอบรม · Module 4.1 – 4.6

รวม

6

ชั่วโมง

ยกระดับ API จากที่ใช้งานได้ให้กลายเป็นระบบที่ปลอดภัย ตรวจสอบได้ และมีการทดสอบครอบคลุม

4.1 Validation และ Error Handling อย่างเป็นระบบ

- การใช้ Zod ตรวจสอบข้อมูลขาเข้าทั้งใน Route Handler และ Server Action
- การสร้าง Schema ที่ใช้ร่วมกันได้ระหว่าง Client Form และ Server
- การออกแบบรูปแบบ Response ของ API ให้เป็นมาตรฐานเดียวกันทั้งระบบ
- การจัดการ Error ของ Prisma: PrismaClientKnownRequestError และรหัสที่พบบ่อย (P2002, P2003, P2025)
- การแปลง Error ของฐานข้อมูลให้เป็นข้อความที่ผู้ใช้เข้าใจ
- การทำ Centralized Error Handler และการเก็บ Log
- instrumentation.ts และ onRequestError สำหรับดักจับ Error ระดับ Server

4.2 Data Access Layer และความปลอดภัยของข้อมูล

- แนวคิด Data Access Layer (DAL) และการแยก Business Logic ออกจาก Route
- รูปแบบ DTO เพื่อควบคุมว่าข้อมูลใดบ้างที่ถูกส่งออกไปยัง Client
- การใช้ server-only เพื่อป้องกัน Secret รั่วไหลไปยัง Client Bundle
- การจัดการ Environment Variable อย่างปลอดภัย และความต่างระหว่าง Build Time กับ Runtime
- ช่องโหว่ที่พบบ่อยและวิธีป้องกัน: Mass Assignment, IDOR, การเปิดเผยข้อมูลเกินจำเป็น
- ภาพรวม CVE ที่เคยเกิดกับ Next.js และเหตุผลที่ต้องอัปเดตเวอร์ชันเสมอ

4.3 Authentication ด้วย Better Auth

- ภาพรวมกลไก Authentication บนเว็บ: Session-based กับ Token-based (JWT)
- ทำไมหลักสูตรนี้เลือก Better Auth เป็นไลบรารีหลักสำหรับ Next.js 16
- การติดตั้งและตั้งค่า Better Auth พร้อมไฟล์ lib/auth.ts และ lib/auth-client.ts
- การเชื่อม Prisma Adapter และการ Generate Schema ด้วย @better-auth/cli generate
- Email & Password: การสมัครสมาชิก การเข้าสู่ระบบ การยืนยันอีเมล และการรีเซ็ตรหัสผ่าน
- Social Provider: Google, GitHub และการทำ Account Linking
- Plugin nextCookies() ที่ต้องอยู่ท้ายสุดของ Array เสมอ เพื่อให้ Server Action ตั้ง Cookie ได้
- การอ่าน Session ใน Server Component, Route Handler และ Server Action ด้วย auth.api.getSession()
- การใช้ Client Hook useSession, signIn, signUp และ signOut ฝั่ง Client Component
- การตั้งค่า Cookie ให้ปลอดภัย: HttpOnly, Secure, SameSite และ Cookie Cache
- ความปลอดภัยเพิ่มเติม: Rate Limiting และการจัดการ Session แบบเพิกถอนได้ทันที
- ภาพรวม Plugin ที่ใช้บ่อย: Two-Factor Authentication (2FA), Passkey (WebAuthn) และ Magic Link



DAY
4

API ระดับ Production · Validation, Security, Authentication, RBAC และ Testing

เนื้อหาการอบรม · Module 4.1 – 4.6 (ต่อ)

รวม
6
ชั่วโมง

4.4 Authorization และ Role-Based Access Control (RBAC)

- การออกแบบ Schema สำหรับ Role และ Permission ด้วย Prisma
 - Access Control API ของ Better Auth** การประกาศ Statement, Role และ Permission
 - Admin Plugin และ Organization Plugin สำหรับระบบหลายบทบาทและหลายองค์กร
 - การตรวจสอบสิทธิ์ในระดับต่างๆ: Route, Component, Server Action และ Data Layer
 - proxy.ts** แทน **middleware.ts** ใน Next.js 16 และการทำ Route Protection
 - ข้อควรระวัง: proxy.ts ทำงานบน Node.js Runtime เท่านั้นและกำหนด runtime ไม่ได้
 - การทำ Row-Level Security ในระดับแอปพลิเคชันด้วย Prisma Client Extensions
 - การเขียนฟังก์ชัน requireAuth และ requireRole ที่ใช้ซ้ำได้ทั้งระบบ
- **Workshop:** เพิ่มระบบ Login และ RBAC 3 ระดับ (Admin, Staff, Customer) เข้ากับระบบงานตัวอย่าง

4.5 File Upload และงานเบื้องหลัง

- การรับไฟล์อัปโหลดผ่าน Server Action และ Route Handler
- การตรวจสอบชนิดและขนาดไฟล์ และการเก็บ Metadata ลงฐานข้อมูล
- แนวทางเก็บไฟล์: Local Storage, S3-compatible Storage และการทำ Presigned URL
- การใช้ after() เพื่อทำงานเบื้องหลังหลังส่ง Response แล้ว
- แนวทางการทำ Background Job และ Scheduled Task ร่วมกับ Next.js

4.6 Testing: Vitest และ Playwright

- กลยุทธ์การทดสอบสำหรับแอป Next.js (Testing Pyramid ในบริบท Full-Stack)
- การตั้งค่า Vitest ให้ทำงานร่วมกับ Next.js และ TypeScript
- Unit Test สำหรับ Business Logic และฟังก์ชันใน Data Access Layer
- Component Test ด้วย Testing Library
- Integration Test สำหรับ API ที่ต่อฐานข้อมูลจริง** การใช้ Test Database และการ Reset ข้อมูลระหว่างเทสต์
- การ Mock Prisma Client และเมื่อไรที่ไม่ควร Mock
- E2E Test ด้วย Playwright: การเขียน Scenario, การจัดการ Authentication State
- การรัน Test อัตโนมัติในกระบวนการ CI

DAY
5

Caching, Performance, Deployment, CI/CD และ Capstone Project

เนื้อหาการอบรม · Module 5.1 – 5.6

รวม
6 ชั่วโมง

ปรับจูนแอปให้เร็วขึ้นด้วยระบบแคชแบบใหม่ของ Next.js 16 นำขึ้น Production ด้วย Docker และปิดท้ายด้วยการสร้างระบบงานจริงด้วยตัวเอง

5.1 ระบบแคชของ Next.js 16 และ Cache Components

- ภาพรวมขั้นแคชทั้งหมดของ Next.js และการเปลี่ยนแปลงใน v16
- Cache Components** เปิดใช้ด้วย cacheComponents: true และแนวคิด Dynamic by Default
- คำสั่ง 'use cache' และ Scope ที่ใช้ได้ (Page, Component, Function)
- 'use cache: private' และ 'use cache: remote' สำหรับสถานการณ์ที่ต่างกัน
- การกำหนดอายุแคชด้วย cacheLife และการติด Tag ด้วย cacheTag
- การล้างแคช: revalidateTag(tag, profile) ที่ต้องระบุ Profile แล้ว, updateTag() และ refresh()
- ข้อจำกัดที่ต้องรู้** ภายใน use cache เรียก cookies() / headers() ไม่ได้ และส่ง Class Instance เข้าไปไม่ได้
- รูปแบบการลดภาระฐานข้อมูลด้วยการแคชผลลัพธ์ Query ของ Prisma
- ISR และการ Revalidate หน้าเพจแบบตามเวลาและตามเหตุการณ์

5.2 Partial Prerendering และ Performance Optimization

- แนวคิด Partial Prerendering (PPR) และการทำงานร่วมกับ Cache Components
- Partial Prefetching และ Instant Navigations ใน Next.js 16.3
- การวิเคราะห์ Bundle Size และการลดขนาด JavaScript ฝั่ง Client
- Dynamic Import, Code Splitting และการเลื่อนโหลด Component ที่ไม่จำเป็น
- การปรับแต่งรูปภาพ: next/image, remotePatterns, localPatterns และค่ามาตรฐานใหม่ของ v16
- React Compiler: เปิดใช้อย่างไร ได้อะไร และแลกกับอะไร
- การวัดผลด้วย Core Web Vitals และ Lighthouse
- การปรับจูนฝั่งฐานข้อมูล: การอ่าน EXPLAIN, การเพิ่ม Index ที่ถูกจุด และการลดจำนวน Query

5.3 Deployment แบบ Self-hosted ด้วย Docker

- ตัวเลือกในการ Deploy: Node.js Server, Docker, Static Export และ Adapter API
- การตั้งค่า output: 'standalone' และการเขียน Dockerfile แบบ Multi-stage
- การจัดการ Environment Variable ให้ใช้ Image เดียวได้หลายสภาพแวดล้อม
- การรัน prisma migrate deploy ในขั้นตอน Deploy อย่างปลอดภัย
- Docker Compose สำหรับ Stack ครบชุด: Next.js + PostgreSQL + Reverse Proxy
- การตั้งค่า Nginx เป็น Reverse Proxy และการปิด Buffering เพื่อให้ Streaming ทำงาน
- การ Deploy หลาย Instance: generateBuildId, NEXT_SERVER_ACTIONS_ENCRYPTION_KEY และ deploymentId
- Custom Cache Handler ด้วย Redis สำหรับระบบที่มีหลาย Instance
- การ Deploy บนแพลตฟอร์มอื่น: Vercel และภาพรวม Adapter API

DAY
5

Caching, Performance, Deployment, CI/CD และ Capstone Project

เนื้อหาการอบรม · Module 5.1 – 5.6 (ต่อ)

รวม

6 ชั่วโมง

5.4 CI/CD และการดูแลระบบหลัง Deploy

- การออกแบบ Pipeline ด้วย GitHub Actions: Lint, Type Check, Test, Build, Deploy
- การจัดการ Migration ใน Pipeline และการแยก Database ของแต่ละ Environment
- การจัดการ Secret และ Credential อย่างปลอดภัย
- การทำ Health Check และ Graceful Shutdown
- แนวทางการติดตามระบบ (Monitoring) และการเก็บ Log ในระดับ Production
- การวางแผนอัปเดตเวอร์ชัน Next.js และ Prisma อย่างมีระบบ พร้อมการใช้ Codemod

5.5 Capstone Project

- โจทย์ พัฒนาระบบจัดการสินค้าและคำสั่งซื้อ (Product & Order Management) ให้สมบูรณ์
- ออกแบบ Schema ด้วย Prisma และทำ Migration บน PostgreSQL
- พัฒนา REST API ด้วย Route Handlers พร้อม Validation และ Error Handling ครบถ้วน
- พัฒนาส่วนติดต่อผู้ใช้ด้วย Server Components และ Server Actions
- ใส่ระบบ Authentication และ RBAC ให้ครบทุกชั้นการเข้าถึง
- เพิ่มการแคชด้วย use cache และ cacheTag ในจุดที่เหมาะสม
- เขียน Test ครอบคลุมทั้ง Unit, Integration และ E2E
- สร้าง Dockerfile และ Deploy ขึ้น Environment ทดสอบ

• นำเสนอผลงานและร่วมรีวิวกั้ดกับวิทยากรและเพื่อนร่วมชั้น:

5.6 สรุปและแนวทางต่อยอด

- สรุปภาพรวมสถาปัตยกรรมที่ได้เรียนรู้ตลอด 5 วัน
- Checklist ก่อนนำระบบขึ้น Production
- แนวทางต่อยอด: Monorepo (Turborepo), tRPC, GraphQL, Microservices
- แนวทางต่อยอดฝั่งฐานข้อมูล: Read Replica, Sharding, Caching Layer ด้วย Redis
- การติดตามความเปลี่ยนแปลงของ Next.js และ Prisma รวมถึงแหล่งข้อมูลที่ต้องติดตาม
- ถาม-ตอบและให้คำแนะนำเฉพาะรายโปรเจกต์

06 สิ่งที่คุณจะเข้าอบรมจะได้รับ What you will get

- เอกสารประกอบการอบรมฉบับเต็มภาษาไทย
- Source Code ตัวอย่างและ Starter Project ทุกวัน
- ไฟล์ Docker Compose สำหรับ PostgreSQL และ MariaDB พร้อมใช้งาน
- Checklist และ Best Practices สำหรับนำไปใช้ในโปรเจกต์จริง
- วุฒิบัตรรับรองการอบรมจากสถาบันไอทีจีเนียส เอ็นจิเนียริ่ง
- ไฟล์บันทึกการอบรม (ตามเงื่อนไขของแต่ละรอบ)



07 การเตรียมตัวก่อนเข้าอบรม Before you start

- **เครื่องคอมพิวเตอร์** Windows 10/11, macOS หรือ Linux หน่วยความจำ 16 GB ขึ้นไป (ขั้นต่ำ 8 GB)
- **ซอฟต์แวร์ที่ต้องติดตั้งล่วงหน้า** Node.js 22 LTS, Git, Visual Studio Code, Docker Desktop
- **บัญชีที่ควรเตรียม** GitHub Account (สำหรับหัวข้อ OAuth และ CI/CD)
- **อินเทอร์เน็ต** ความเร็วเสถียรเพียงพอสำหรับ Zoom พร้อมเปิดกล้องและแชร์หน้าจอ
- **แนะนำอย่างยิ่ง** ใช้จอภาพ 2 จอ หรือแท็บเล็ตเสริม เพื่อดูสไลด์และเขียนโค้ดไปพร้อมกัน
- วิทยากรจะส่งคู่มือเตรียมเครื่องให้ล่วงหน้าก่อนวันอบรม และมีช่วงตรวจสอบความพร้อมก่อนเริ่มวันแรก

สนใจสมัครอบรม · ขอใบเสนอราคา

ขอใบเสนอราคา

อบรมออนไลน์ผ่าน Zoom แบบสด · จัดอบรมแบบ In-house ภายในองค์กรได้ทั้งออนไลน์และ On-site ·
ราคายังไม่รวมภาษีมูลค่าเพิ่ม (VAT 7%)

☎ 02-570-8449

LINE @itgenius

☎ 088-807-9770, 092-841-7931

✉ contact@itgenius.co.th



เพิ่มเพื่อน
LINE