

3 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง LMS

อธิบายถึงทฤษฎี / เครื่องมือ / งานวิจัยที่เกี่ยวข้อง

● ทฤษฎีที่เกี่ยวข้อง

- Client–Server Architecture ระบบ LMS ถูกออกแบบตามสถาปัตยกรรม Client–Server
 - Frontend (Next.js, TypeScript, React Context, Tailwind CSS) แสดงผลและรับอินพุตจากผู้ใช้
 - Backend (Django, Django Rest Framework) ทำหน้าที่ประมวลผลข้อมูลและสื่อสารผ่าน REST API (JSON over HTTPS) โดยการใช้ PostgreSQL เป็นฐานข้อมูลหลักทำให้สามารถรองรับการจัดเก็บข้อมูลขนาดใหญ่และซับซ้อน เช่น เนื้อหาบทเรียนและผลการเรียนของผู้ใช้ได้อย่างมีประสิทธิภาพ
- Software Requirement Specification (SRS)
 - การกำหนดความต้องการซอฟต์แวร์ในระบบ LMS อ้างอิงจากมาตรฐาน IEEE 830-1998 เพื่อทำให้การออกแบบและการพัฒนาระบบเป็นไปอย่างเป็นระบบ สามารถตรวจสอบและทดสอบความถูกต้องได้ง่าย โดยในกรณีนี้การออกแบบ SRS จะกำหนดคุณสมบัติ เช่น การเข้าสู่ระบบด้วย JWT Authentication การจัดการคอร์สเรียน การเก็บผลการเรียน และการรองรับผู้ใช้หลายบทบาท
- Software Testing Theory
ทฤษฎีการทดสอบซอฟต์แวร์ถูกนำมาใช้ในการพัฒนาระบบ LMS เพื่อให้มั่นใจว่าระบบทำงานได้ถูกต้องและมีเสถียรภาพ โดยมีการทดสอบหลายระดับ ได้แก่
 - Unit Test: ทดสอบฟังก์ชัน
 - Integration Test: ทดสอบการเชื่อมต่อ Frontend กับ Backend ผ่าน API
 - User Acceptance Test (UAT): ให้ผู้ใช้ทดลองใช้งานระบบจริง
 - Performance Test: ใช้เครื่องมือเช่น Google Lighthouse เพื่อตรวจสอบประสิทธิภาพ และ SEO

● เครื่องมือที่เกี่ยวข้อง

- Next.js 14+ และ TypeScript: รองรับการสร้าง UI ที่มีประสิทธิภาพสูง ใช้หลักการ Component-based ทำงานร่วมกับ React Context สำหรับการจัดการ State
- Tailwind CSS: ช่วยออกแบบ UI แบบ Responsive รองรับหลายอุปกรณ์
- Django 5 + Django Rest Framework: ใช้พัฒนา Backend และ API
- PostgreSQL: ฐานข้อมูล
- drf-spectacular (Swagger UI): สำหรับสร้างเอกสาร API อัตโนมัติ
- Docker / Docker Compose: ใช้ในการ Containerize ระบบเพื่อให้ง่ายต่อการ Deploy และ Scaling
- GitHub Actions (CI/CD), Postman, Google Lighthouse: ใช้ทดสอบและประเมินคุณภาพของระบบ

● งานวิจัยที่เกี่ยวข้อง

- งานวิจัยด้าน Web Application Development ที่เน้นการใช้สถาปัตยกรรม Client–Server และ REST API ช่วยเพิ่มความสามารถในการขยายระบบ LMS และทำให้การพัฒนาแบบแยกส่วน (Frontend/Backend) มีความยืดหยุ่นมากขึ้น
- งานวิจัยด้าน Software Quality Assurance ที่แนะนำการวัดคุณภาพผ่านตัวชี้วัด เช่น MTBF, MTTR, Error Rate, Response Time เพื่อประเมินประสิทธิภาพและความน่าเชื่อถือของระบบ