

Project: Library Room Booking System

ระบบจองห้องสมุด (Library Room Booking System) เป็นเว็บแอปพลิเคชันเต็มรูปแบบ (Full-Stack Web Application) โดยมีวัตถุประสงค์เพื่อจำลองและสาธิตกระบวนการจองห้องสำหรับอ่านหนังสือหรือทำกิจกรรมต่างๆ ภายในห้องสมุด ระบบถูกออกแบบมาเพื่ออำนวยความสะดวกให้กับผู้ใช้งานทั่วไปในการค้นหาและส่งคำขอจองห้อง ในขณะเดียวกันก็มีระบบหลังบ้านที่มีประสิทธิภาพสำหรับผู้ดูแลในการจัดการคำขอและทรัพยากรห้องสมุด

โครงการนี้แสดงให้เห็นถึงความสามารถในการออกแบบและพัฒนาระบบ ตั้งแต่ต้นจนจบ ตั้งแต่การวางโครงสร้างฐานข้อมูล, การสร้าง API Server, ไปจนถึงการพัฒนาส่วนติดต่อผู้ใช้ (User Interface) ที่ทันสมัยและตอบสนองต่อทุกขนาดหน้าจอ (Responsive Design) โดยเน้นหลักการเขียนโค้ดที่สะอาด (Clean Code) และสถาปัตยกรรมที่ง่ายต่อการบำรุงรักษา

ฟีเจอร์ของระบบ

ระบบถูกแบ่งการทำงานออกเป็น 2 ส่วนตามบทบาทของผู้ใช้งาน:

1. ส่วนของผู้ใช้งานทั่วไป (User Role)

- **ระบบการจองห้อง (Booking System):** ผู้ใช้สามารถจองห้องได้ผ่านกระบวนการ 4 ขั้นตอนที่ชัดเจน คือ 1.เลือกสถานที่ 2.เลือกห้อง 3.เลือกวันและเวลา 4.กรอกข้อมูลส่วนตัวเพื่อส่งคำขอ
- **หน้าตารางเวลา (Schedule Timetable):** สามารถดูภาพรวมการจองทั้งหมดของทุกห้องในแต่ละวันได้ โดยมี Filter สำหรับกรองข้อมูลตามสถานที่ และแสดงสถานะของห้องแบบ Real-time (ว่าง, รออนุมัติ, จองแล้ว)
- **หน้าประวัติการจอง (Booking History):** ผู้ใช้สามารถตรวจสอบประวัติและสถานะคำขอจองของตนเองได้ (รออนุมัติ, อนุมัติแล้ว, ไม่อนุมัติ) พร้อมฟังก์ชันการค้นหาและกรองข้อมูล

2. ส่วนของผู้ดูแลระบบ (Administrator Role)

- **การจัดการคำขออนุมัติ (Request Management):** ผู้ดูแลสามารถดูรายการคำขออนุมัติที่รอการอนุมัติทั้งหมด และทำการ "อนุมัติ" หรือ "ไม่อนุมัติ" คำขอได้
- **การจัดการห้องและสถานที่ (Room & Location Management):** ระบบ CRUD (Create, Read, Update, Delete) ที่สมบูรณ์แบบสำหรับจัดการข้อมูลห้องและสถานที่ เช่น การเพิ่มอาคารใหม่ หรือการแก้ไขรายละเอียดของห้อง (ความจุ, สิ่งอำนวยความสะดวก)
- **หน้าประวัติการอนุมัติ (Approval History):** สามารถดูประวัติการตัดสินใจอนุมัติ/ไม่อนุมัติทั้งหมดที่ผ่านมาได้ พร้อมระบบค้นหาและจัดเรียงข้อมูล (Sorting)

เทคโนโลยีที่ใช้ (Technology Stack)

- **Frontend:**
 - **Framework:** Angular
 - **Styling:** Tailwind CSS
 - **Libraries:** SweetAlert2
- **Backend:**
 - **Framework:** Node.js, Express.js
 - **ORM:** Prisma
- **Database:**
 - **Type:** PostgreSQL

หลักการออกแบบและพัฒนา (Design & Development Principles)

- **Clean Code & SOLID Principles:** โครงสร้างโปรเจกต์ถูกแบ่งอย่างชัดเจนระหว่าง Frontend และ Backend ในส่วนของ Backend

มีการแบ่งแยกหน้าที่ตามหลัก Separation of Concerns (Routes, Controllers, Services) เพื่อให้ง่ายต่อการทำความเข้าใจและบำรุงรักษา

- **RESTful API Design:** การออกแบบ API Server เป็นไปตามมาตรฐาน RESTful เพื่อให้การสื่อสารระหว่าง Frontend และ Backend มีประสิทธิภาพและเป็นระบบ
- **Responsive Design:** ส่วนติดต่อผู้ใช้ถูกออกแบบด้วย Tailwind CSS ให้สามารถแสดงผลได้อย่างเหมาะสมในทุกขนาดหน้าจอ ตั้งแต่เดสก์ท็อปไปจนถึงโทรศัพท์มือถือ
- **User-Centric Interface:** เน้นการออกแบบ UI/UX ที่ทันสมัยและใช้งานง่าย เพื่อสร้างประสบการณ์ที่ดีให้กับผู้ใช้งานทั้งสองบทบาท