เอกสารประกอบการสอบหัวข้อโปรเจกต์

โครงการพัฒนาระบบจัดการหอพักบ้านกาญจน์ เรสซิเดนซ์

(Baankarn Residence — Dormitory Management System)

เอกสารฉบับนี้จัดทำเพื่อใช้ประกอบการนำเสนอและสอบหัวข้อโครงงาน
สรุปจากการวิเคราะห์ซอร์สโค้ดจริงของระบบ (server.js, services 31 โมดูล, route modules, ฐานข้อมูล 32 ตาราง และชุดทดสอบ 64 ไฟล์) ร่วมกับเอกสารออกแบบภายในโครงการ

บทที่ 1 บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ธุรกิจหอพักและอพาร์ตเมนต์ขนาดเล็กถึงกลางในประเทศไทยจำนวนมาก ยังคงบริหารจัดการด้วยวิธีการแบบดั้งเดิม เช่น การจดบันทึกลงสมุด การใช้ไฟล์ Microsoft Excel หรือการสื่อสารกับผู้เช่าผ่านแอปพลิเคชันแชตทั่วไป ซึ่งก่อให้เกิดปัญหาในการดำเนินงานหลายประการ ได้แก่

1. ความผิดพลาดด้านการเงิน — การคำนวณค่าเช่า ค่าน้ำ ค่าไฟ ค่าปรับชำระล่าช้า และการหักเงินมัดจำเมื่อย้ายออก ทำด้วยมือ ทำให้เกิดความผิดพลาดได้ง่าย และตรวจสอบย้อนหลังได้ยาก
2. การออกบิลล่าช้าหรือตกหล่น — เมื่อไม่มีระบบอัตโนมัติ ผู้ดูแลต้องจดจำรอบบิลของผู้เช่าแต่ละรายเอง ทำให้เกิดการลืมออกบิลและสูญเสียรายได้โดยไม่รู้ตัว
3. การจองห้องซ้ำซ้อน — เมื่อไม่มีระบบกลางที่ล็อกสถานะห้อง อาจเกิดการรับจองห้องเดียวกันให้ผู้เช่าหลายราย
4. การติดตามเงินมัดจำและสัญญายาก — เอกสารกระจัดกระจาย ไม่มีการแจ้งเตือนเมื่อสัญญาใกล้หมดอายุ
5. การสื่อสารและแจ้งเตือนไม่เป็นระบบ — การแจ้งบิล แจ้งซ่อม หรือยืนยันการชำระเงินทำผ่านการพิมพ์ข้อความด้วยมือทีละราย
6. ขาดการบันทึกร่องรอยการทำงาน (Audit Trail) — เมื่อข้อมูลถูกแก้ไข ไม่สามารถตรวจสอบได้ว่าใครแก้ เมื่อใด

จากปัญหาข้างต้น ผู้จัดทำจึงพัฒนา ระบบจัดการหอพักบ้านกาญจน์ เรสซิเดนซ์ ขึ้น เพื่อเป็นระบบสารสนเทศกลางที่รวมทุกกระบวนการบริหารหอพักไว้ในที่เดียว ตั้งแต่การแสดงสถานะห้องว่าง การจองห้องออนไลน์ การจัดการสัญญาเช่า การออกบิลและรับชำระเงินผ่าน PromptPay การตรวจสอบสลิปอัตโนมัติ การบันทึกค่ามิเตอร์ ไปจนถึงการแจ้งเตือนผ่าน LINE โดยออกแบบให้ **กระบวนการที่เกี่ยวข้องกับเงินทุกก้อน (มัดจำ ค่าจอง ค่าเช่า ค่าปรับ) ไหลผ่านระบบบิลเดียวกัน** เพื่อความถูกต้องและตรวจสอบได้

โครงการนี้เริ่มต้นจากการพัฒนาเว็บจำลองแสดงสถานะห้องของหอพัก 5 ชั้น และได้รับการพัฒนาต่อยอดอย่างต่อเนื่องจนกลายเป็นระบบบริหารจัดการที่สามารถใช้งานได้จริง (Production-ready) และถูกนำขึ้นใช้งานบนระบบคลาวด์ (Railway) แล้ว

1.2 วัตถุประสงค์ของโครงการ

1. เพื่อพัฒนาระบบแสดงสถานะห้องพักของหอพัก 5 ชั้น จำนวน 40 ห้อง ที่ผู้สนใจสามารถดูสถานะห้องว่างและรายละเอียดค่าใช้จ่ายได้โดยไม่ต้องเข้าสู่ระบบ
2. เพื่อพัฒนาระบบจองห้องและแจ้งซ่อมออนไลน์สำหรับผู้เช่าและผู้สนใจ

3. เพื่อพัฒนาระบบหลังบ้าน (Admin) สำหรับผู้ดูแลในการจัดการห้อง ผู้เช่า สัญญาเช่า บิล การเงิน มิเตอร์ และการตั้งค่าระบบ
4. เพื่อพัฒนาระบบออกบิลและรับชำระเงินกึ่งอัตโนมัติ โดยออกบิลเป็นไฟล์ PDF ภาษาไทยพร้อมรหัส PromptPay QR และตรวจสอบสลิปการโอนเงินอัตโนมัติ
5. เพื่อลดภาระงานซ้ำซ้อนของผู้ดูแล ด้วยระบบงานเบื้องหลังอัตโนมัติ เช่น การออกบิลรายเดือน การคิดค่าปรับชำระล่าช้า และการสำรองข้อมูล
6. เพื่อพัฒนาระบบแจ้งเตือนหลายช่องทาง (LINE / อีเมล / SMS) ให้ผู้เช่าและผู้ดูแลรับทราบเหตุการณ์สำคัญแบบทันเวลา
7. เพื่อออกแบบระบบให้มีความปลอดภัยและความถูกต้องของข้อมูล ทั้งการเข้ารหัสข้อมูลส่วนบุคคล การควบคุมสิทธิ์การเข้าถึง และการบันทึกร่องรอยการใช้งาน

1.3 ขอบเขตของโครงการ

1.3.1 ขอบเขตด้านผู้ใช้งาน (4 บทบาท)

บทบาท	สิทธิ์การใช้งาน
ผู้ใช้ทั่วไป / สาธารณะ	ดูสถานะห้องว่าง จองห้อง และแจ้งซ่อม โดยไม่ต้องเข้าสู่ระบบ
ผู้เช่า (Tenant)	เข้าสู่ระบบผ่านพอร์ทัลผู้เช่า ดูบิล อัปโหลดสลิป แจ้งซ่อม ดูสัญญา และจัดการโปรไฟล์
ผู้ดูแลระบบ (Admin)	จัดการระบบทั้งหมด แบ่งสิทธิ์เป็น 4 ระดับ คือ เจ้าของ (owner) > ผู้จัดการ (manager) > พนักงาน (staff) > อ่านอย่างเดียว (readonly)
ระบบภายนอก	LINE Webhook และอุปกรณ์ควบคุมการเข้า-ออก (RFID/QR)

1.3.2 ขอบเขตด้านฟังก์ชันการทำงาน

ฝั่งผู้เช่าและสาธารณะ

- แดชบอร์ดแสดงสถานะห้องทั้ง 5 ชั้น เลือกดูรายชั้น/รายห้องพร้อมรายละเอียดและรูปภาพ
- ฟอร์มจองห้องออนไลน์ (ไม่ต้องเข้าสู่ระบบ)
- ระบบแจ้งซ่อมและติดตามสถานะงานซ่อม
- พอร์ทัลผู้เช่า: ดูบิล อัปโหลดสลิป ดูสัญญา จัดการโปรไฟล์ รองรับโหมดมืดและสองภาษา (ไทย/อังกฤษ)

ฝั่งผู้ดูแล

- จัดการห้องพักและผู้เช่า
- จัดการการจอง (ขั้นตอน pending → reviewing → approved/rejected)
- จัดการสัญญาเช่า การเชิญเซ็นสัญญา และการสร้างสัญญาเป็นไฟล์ PDF
- จัดการบิลและการเงิน (รองรับภาษีมูลค่าเพิ่ม ค่าปรับ ค่าใช้จ่ายประจำ การคิดค่าเช่าตามสัดส่วน และเงินมัดจำ)
- คิวตรวจสอบสลิปการชำระเงิน (อนุมัติ/ปฏิเสธ) พร้อมการตรวจสลิปอัตโนมัติ
- บันทึกค่ามิเตอร์น้ำ/ไฟ พร้อมกราฟและการตรวจจับค่าผิดปกติทางสถิติ (3 ซิกมา)
- กระดาน Kanban ติดตามงานแจ้งซ่อม
- ระบบจัดการพัสดุ (parcels) และการควบคุมการเข้า-ออก (access control)
- รายงานรายได้ ลูกหนี้ค้างชำระ และสถานะห้อง
- ระบบแจ้งเตือนหลายช่องทางพร้อมคิวการส่ง
- ระบบเปิด/ปิดฟีเจอร์ (Feature Flags) และการตรวจสอบความพร้อมของระบบ (Health/Readiness)

1.3.3 ขอบเขตด้านเทคโนโลยี

ส่วน	เทคโนโลยีที่ใช้

ส่วนหน้า (Frontend)	React 18 (โหลดผ่าน CDN) + Babel แปลงในเบราว์เซอร์
ส่วนหลัง (Backend)	Node.js 18+ / Express 4
ฐานข้อมูล	PostgreSQL (32 ตาราง relational + JSONB key/value)
การยืนยันตัวตน	bcrypt + express-session (แอดมิน) และตาราง session แยกสำหรับผู้เช่า
เอกสาร/การเงิน	PDFKit + ฟอนต์ Sarabun, promptpay-qr (มาตรฐาน EMV QR), Zod validation
การแจ้งเตือน	LINE Messaging API + SMTP (nodemailer) + SMS
ระบบโฮสติ้ง	Railway (Cloud)

1.3.4 ข้อจำกัดของโครงการ (สิ่งที่อยู่นอกขอบเขต)

1. ยังไม่ได้ปรับไปใช้ระบบ build แบบ Vite + TypeScript ทำให้การโหลดหน้าครั้งแรกมีการแปลง Babel ในเบราว์เซอร์ เพิ่มเวลาประมาณ 2 วินาที
2. ยังไม่รองรับการเชื่อมต่อมิเตอร์ IoT จริงผ่าน MQTT รองรับเฉพาะการกรอกค่าด้วยมือ
3. ระบบยังเก็บข้อมูลห้อง/ผู้เช่าไว้สองแหล่ง (JSONB blob และตาราง relational) ซึ่งเป็นหนี้ทางเทคนิคที่อยู่ระหว่างการปรับปรุงให้เหลือแหล่งเดียว
4. การเข้าสู่ระบบของผู้เช่าใช้ปัจจัยเดียว (เบอร์โทรศัพท์) ยังไม่มีการยืนยัน OTP/PIN
5. การจำกัดอัตราการเรียก (Rate limit) ทำงานในหน่วยความจำต่อโพรเซส ยังไม่รองรับการทำงานข้ามหลายเครื่องแม่ข่าย

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ผู้สนใจสามารถดูสถานะห้องว่างและจองห้องได้สะดวกตลอด 24 ชั่วโมง
2. ผู้ดูแลลดภาระงานเอกสารและการคำนวณด้วยมือ ลดความผิดพลาดด้านการเงิน
3. ลดการสูญเสียรายได้จากการลืมออกบิลหรือการคิดค่าปรับผิดพลาด
4. ผู้เช่าได้รับความสะดวกในการดูบิล ชำระเงิน และแจ้งซ่อมผ่านช่องทางออนไลน์
5. ข้อมูลการดำเนินงานถูกจัดเก็บอย่างเป็นระบบ ปลอดภัย และตรวจสอบย้อนหลังได้
6. เป็นต้นแบบระบบสารสนเทศที่สามารถนำไปประยุกต์ใช้กับหอพักหรืออพาร์ตเมนต์อื่นได้ด้วยกลไกเปิด/ปิดฟีเจอร์ตามความต้องการ

1.5 ขนาดและความซับซ้อนของระบบ (โดยสังเขป)

รายการ	ปริมาณ
โค้ดส่วนหลัง (Backend)	ประมาณ 52,000 บรรทัด
โค้ดส่วนหน้า (Frontend)	ประมาณ 42,000 บรรทัด
ชุดทดสอบ (Tests)	ประมาณ 21,500 บรรทัด / 64 ไฟล์
ตารางฐานข้อมูล	32 ตาราง
โมดูลบริการ (Services)	31 โมดูล
หน้าจอผู้ดูแล (Admin Pages)	33 หน้า
ชุดตรวจสอบข้อมูลเข้า (Zod schemas)	104 ชุด

บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง (โดยสรุป)

- สถาปัตยกรรมแบบ Client-Server และ Single Page Application (SPA) ด้วย React
- ฐานข้อมูลเชิงสัมพันธ์ (Relational Database) ร่วมกับการเก็บข้อมูลแบบ JSONB
- REST API สำหรับการสื่อสารระหว่างส่วนหน้าและส่วนหลัง
- การยืนยันตัวตนและการจัดการเซสชัน (Authentication & Session) ด้วย bcrypt และ cookie
- การควบคุมสิทธิ์ตามบทบาท (Role-Based Access Control) 4 ระดับ
- มาตรฐาน PromptPay / EMVCo QR Code สำหรับการชำระเงิน
- LINE Messaging API สำหรับการแจ้งเตือน
- การเข้ารหัสข้อมูล AES-256-GCM สำหรับข้อมูลส่วนบุคคล (เลขบัตรประชาชน)
- การตรวจจับค่าผิดปกติทางสถิติ (3-Sigma Anomaly Detection) สำหรับค่ามิเตอร์

บทที่ 3 หลักการออกแบบระบบ — 3 เสาหลัก

ระบบออกแบบโดยยึดหลัก 3 ด้าน เพื่อให้ทุกฟีเจอร์ (1) ทำงานถูกต้อง (2) มีการป้องกันหลายชั้น และ (3) แจ้งเตือนได้ชัดเจน

เสาที่ 1 — ความถูกต้อง (Correctness)

- จัดเก็บเงินเป็นชนิด NUMERIC(10,2) หลีกเลี่ยงการใช้เลขทศนิยมลอยตัว (float)
- รักษาความสัมพันธ์คงที่ เช่น ยอดรวม = ยอดย่อย + ภาษี + ค่าปรับ ทั้งในระดับโปรแกรมและฐานข้อมูล (DB CHECK)
- ออกแบบให้ทำซ้ำได้ผลเดิม (Idempotent) ด้วย unique key และ ON CONFLICT

เสาที่ 2 — การป้องกันหลายชั้น (Defense in Depth) 7 ชั้น

1. ตรวจจับและบล็อกการสอดส่อง (probe) ระดับเครือข่าย
2. จำกัดอัตราการเรียก (Rate limit)
3. ป้องกัน CSRF (sameOrigin + double-submit cookie)
4. ยืนยันตัวตน (Authentication)
5. ควบคุมสิทธิ์ตามบทบาท (Authorization / RBAC)
6. ตรวจสอบข้อมูลเข้าด้วย Zod schema + จำกัดความยาว + whitelist
7. รักษาความถูกต้องของข้อมูลด้วย row-lock + transaction + DB CHECK + unique index

เสาที่ 3 — การสังเกตได้และการแจ้งเตือน (Observability) 4 ระดับ

1. บันทึกร่องรอยการทำงาน (Audit log)
2. เหตุการณ์ด้านความปลอดภัย (Security event) แจ้งเจ้าของอัตโนมัติ
3. การแจ้งเตือนเจ้าของ (Owner alert) ผ่าน LINE → อีเมล → log
4. การตรวจสอบสุขภาพระบบ (Health/Readiness)

ภาคผนวก ก — รายการตารางฐานข้อมูลหลัก (32 ตาราง)

กลุ่มหลัก: app_data, auth_users, user_sessions, audit_logs, maintenance_tickets

กลุ่มผู้เช่าและสัญญา: tenants, tenant_sessions, contracts, contract_invitations, contract_templates, rooms_v, bookings, booking_deposit_orphans

กลุ่มการเงิน: bills, payments, recurring_charges

กลุ่มมิเตอร์และการเข้า-ออก: meter_readings, access_logs, access_cards, access_devices

กลุ่มการแจ้งเตือนและ LINE: notifications_log, notifications_queue, line_bindings, line_oas, admin_line_recipients, admin_recipient_tokens

กลุ่มระบบและความปลอดภัย: secrets, system_settings, file_uploads, login_lockouts, owner_claim_tokens, parcels

---

ภาคผนวก ข — แนวคำถาม-คำตอบสำหรับการสอบ

ถาม: ระบบนี้ต่างจากการใช้ Excel หรือ Google Form อย่างไร
ตอบ: ระบบทำงานเงินอัตโนมัติ (ออกบิล คิดค่าปรับ หักมัดจำ) ป้องกันการจองห้องซ้ำในระดับฐานข้อมูล แจ้งเตือนผ่าน LINE และบันทึกร่องรอยการแก้ไขทุกครั้ง ซึ่งเครื่องมือทั่วไปทำไม่ได้

ถาม: PromptPay QR และการตรวจสลิปทำงานอย่างไร
ตอบ: ระบบสร้าง QR ตามมาตรฐาน EMV ที่ระบุยอดเงิน เมื่อผู้เช่าอัปโหลดสลิป ถ้ายอดและบัญชีตรงกัน ระบบจะอนุมัติอัตโนมัติ หากไม่ตรงจะเข้าคิวให้ผู้ดูแลตรวจสอบ

ถาม: ดูแลความปลอดภัยของข้อมูลส่วนบุคคลอย่างไร
ตอบ: เลขบัตรประชาชนถูกเข้ารหัสด้วย AES-256-GCM ขณะจัดเก็บ และต้องร้องขอเป็นกรณีจึงจะถอดรหัสได้

ถาม: จุดที่ยังต้องพัฒนาต่อคืออะไร
ตอบ: การยุบการเก็บข้อมูลสองแหล่งให้เหลือแหล่งเดียว การปรับไปใช้ Vite + TypeScript การเชื่อมต่อมิเตอร์ IoT จริง และการเพิ่ม OTP/PIN ให้พอร์ทัลผู้เช่า

จัดทำเพื่อประกอบการสอบหัวข้อโครงงาน — สรุปจากการวิเคราะห์ซอร์สโค้ดจริงของระบบ