



รายงาน

เรื่อง ระบบฐานข้อมูลของห้องพัก

จัดทำโดย

นายธนภฤติ อักษรเจริญสุข	รหัสนักศึกษา 66050172
นายภูริเดช สนิธิ	รหัสนักศึกษา 66050336
นางสาวปรียานุช เชื้อน้อย	รหัสนักศึกษา 66050851
นายพลัฏฐ์ มะบุตร	รหัสนักศึกษา 66050907
นางสาวพัชรลักษณ์ รักษ์ณรงค์	รหัสนักศึกษา 66050913
นางสาวพิจิตรา ไทรแก้วดวง	รหัสนักศึกษา 66050917
นางสาวพุทธรชาด เทพสุภรณ์กุล	รหัสนักศึกษา 66050938
นางสาวรุจิรา บุญชมภู	รหัสนักศึกษา 66051013

เสนอ

ผู้ช่วยศาสตราจารย์ ดร.พรพิมล ชัยวุฒิศักดิ์

รายงานฉบับนี้เป็นส่วนหนึ่งของรายวิชา 05406145 ระบบจัดการฐานข้อมูล(Database Management)

หลักสูตรวิทยาศาสตรบัณฑิต สาขาสถิติประยุกต์และการวิเคราะห์ข้อมูล ปีการศึกษา 2567

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

คำนำ

รายงานฉบับนี้จัดทำขึ้นเพื่อเป็นส่วนหนึ่งของรายวิชาระบบฐานข้อมูลเป็นการออกแบบฐานข้อมูลการบริหารจัดการฐานข้อมูล รวมถึงภาษาสอบถามเชิงโครงสร้างหรือ SQL ซึ่งมีความสำคัญอย่างยิ่งในการบริหารฐานข้อมูลให้มีประสิทธิภาพ และสามารถนำความรู้มาประยุกต์ใช้ในการทำระบบฐานข้อมูลของห้องพัก

เนื้อหา รายงานฉบับนี้จัดทำขึ้นเพื่อศึกษาและนำเสนอแนวทางในการพัฒนาและปรับปรุงระบบการจัดการฐานข้อมูลการจองห้องพักให้มีประสิทธิภาพมากยิ่งขึ้น โดยมุ่งเน้นการใช้เทคโนโลยีสารสนเทศในการบริหารข้อมูลการจองห้องพัก รวมไปถึงการจัดการใบเสร็จค่าน้ำ ค่าไฟ เพื่อให้เกิดความสะดวกสบาย รวดเร็ว แม่นยำ และลดปัญหาข้อผิดพลาดที่อาจเกิดขึ้นจากการบริหารจัดการด้วยระบบแบบเดิมได้ นอกจากนี้ รายงานฉบับนี้ยังเป็นประโยชน์ต่อผู้ดูแลห้องพักในการติดตามสถานการณ์การจองห้องพัก การบริหารทรัพยากรให้เกิดประโยชน์สูงสุด และช่วยให้ผู้ใช้บริการสามารถเข้าถึงข้อมูลได้อย่างสะดวกและรวดเร็วมากขึ้น การศึกษานี้ครอบคลุมแนวทางการออกแบบและพัฒนาฐานข้อมูลที่เหมาะสมกับการใช้งานในสถานการณ์จริง ซึ่งสามารถนำไปประยุกต์ใช้ในระบบห้องพักนักศึกษาของสถานศึกษา ห้องพักเอกชน และหน่วยงานที่ให้บริการที่พักในรูปแบบต่าง ๆ ได้อย่างมีประสิทธิภาพ

คณะผู้จัดทำรายงานหวังเป็นอย่างยิ่งว่า รายงานฉบับนี้จะเป็นประโยชน์สำหรับนักศึกษาและผู้ที่เกี่ยวข้องทั่วไปหากมีข้อเสนอแนะอื่น ๆ ผู้ทำรายงานเล็งเห็นดีรับไปปรับปรุงและแก้ไข ให้รายงานฉบับนี้มีความสมบูรณ์ยิ่งขึ้น

คณะผู้จัดทำ

สารบัญ

เรื่อง	หน้า
คำนำ	
บทที่ 1 บทนำ	
1.1 ที่มาและความสำคัญ	1
1.2 วัตถุประสงค์	2
1.3 ประโยชน์ที่คาดว่าจะได้รับ	2
1.4 ขอบเขตการศึกษา	2
บทที่ 2 ทฤษฎีพื้นฐานและเนื้อหาที่เกี่ยวข้อง	
2.1 ทฤษฎีพื้นฐานที่เกี่ยวข้อง	3
2.2 โมเดลทางธุรกิจ	5
บทที่ 3 วิธีดำเนินงานทางธุรกิจ	
3.1 ER-Diagram	6
บทที่ 4 Normalization	
4.1 Relation Schema	10
4.2 Dependency Diagram	11
4.3 Schema Report For Database	12
4.4 Normalization	20
ภาคผนวก	21
บรรณานุกรม	33

บทนำ

บทที่ 1

1.1 ที่มาและความสำคัญ

ในปัจจุบันภาพจำของการจองห้องพักของคนไทยยังคงเป็นกระบวนการที่เต็มไปด้วยความยุ่งยากและข้อผิดพลาด เจ้าของห้องพักจำนวนมากยังใช้วิธีจองแบบดั้งเดิม เช่น การรับจองผ่านโทรศัพท์หรือแอปพลิเคชันแชท เช่น ไลน์ ซึ่งพนักงานต้องเช็คตารางห้องว่างด้วยตัวเอง นอกจากนี้ การใช้ไฟล์ Excel หรือการจดลงสมุดเพื่อบันทึกข้อมูลการจอง มีความเสี่ยงสูงต่อความผิดพลาดและข้อมูลล่าช้า ส่งผลให้พนักงานต้องใช้เวลามากขึ้นในการตอบคำถามซ้ำ ๆ เกี่ยวกับราคา สิ่งอำนวยความสะดวก หรือโปรโมชั่น กระบวนการจองที่ซับซ้อนนี้ทำให้ลูกค้ายุ่งยาก และอาจพลาดโอกาสในการขาย คำถามสำคัญที่หลาย ๆ คนสงสัยคือ เราจะสามารถรับประกันความถูกต้องของข้อมูลได้อย่างไร หากยังต้องใช้แรงงานคนในการตรวจสอบปริมาณข้อมูลจำนวนมาก? การจ้างพนักงานเพื่อเช็คข้อมูลเหล่านี้อาจช่วยแก้ปัญหาได้บางส่วน แต่ก็ไม่สามารถมั่นใจได้ว่าทุกอย่างจะถูกต้อง 100% ตลอดเวลา ปัญหาดังกล่าวไม่เพียงส่งผลกระทบต่อประสิทธิภาพของธุรกิจห้องพัก แต่ยังส่งผลกระทบต่อความพึงพอใจของลูกค้าอีกด้วย ความล่าช้าในการตรวจสอบห้องว่าง การจองซ้ำซ้อน หรือข้อมูลที่ไม่ถูกต้อง ล้วนเป็นปัจจัยที่ลดความน่าเชื่อถือของธุรกิจห้องพัก และอาจส่งผลให้ลูกค้าเลือกใช้บริการจากคู่แข่งแทน นอกจากนี้การพึ่งพาการทำงานของพนักงานเพียงอย่างเดียว โดยไม่มีระบบที่เป็นมาตรฐาน ยังเพิ่มโอกาสในการเกิดข้อผิดพลาดและเป็นภาระที่ไม่จำเป็นต่อองค์กรหากเราสามารถพัฒนาระบบจองห้องพักของเราเอง ที่ช่วยให้ลูกค้าสามารถเลือกห้อง ตรวจสอบห้องว่าง ทำการจอง ชำระเงิน ได้ด้วยตัวเอง เราจะสามารถลดภาระของพนักงาน และทำให้พวกเขาสามารถจดจ่อกับงานที่ต้องใช้ทักษะได้มากขึ้น ในขณะเดียวกันก็ช่วยสร้างประสบการณ์การจองที่สะดวกและรวดเร็วให้กับลูกค้า ดังนั้นจากปัญหาดังกล่าว ผู้นำเสนอโครงการนี้จึงมีความประสงค์นำความรู้วิชา MySQL เก็บฐานข้อมูล เพื่อสร้างระบบจองอัจฉริยะ มาพัฒนาระบบจองห้องพักให้มีความแม่นยำและทำงานได้อัตโนมัติ การจองห้องพักจะกลายเป็นเรื่องง่าย ลดภาระงานของพนักงาน และเพิ่มประสิทธิภาพการดำเนินการ นอกจากนี้ ยังช่วยให้ธุรกิจสามารถแข่งขันได้ดีขึ้นในยุคดิจิทัล ทำให้การให้บริการลูกค้าเป็นไปอย่างราบรื่นและน่าเชื่อถือมากขึ้น

1.2 วัตถุประสงค์

1. เพื่อพัฒนาระบบฐานข้อมูลที่สามารถจัดเก็บและบริหารจัดการข้อมูลการจองห้องพักได้อย่างเป็นระบบ มีความถูกต้อง ปลอดภัย
2. เพื่อเพิ่มประสิทธิภาพในการดำเนินงาน ลดข้อผิดพลาดในการบริหารข้อมูล และช่วยให้ผู้ดูแลห้องพักสามารถบริหารจัดการห้องพักได้อย่างมีประสิทธิภาพ
3. เพื่ออำนวยความสะดวกให้กับผู้ใช้บริการในการตรวจสอบข้อมูลการจอง การเลือกห้องพัก การติดตามสถานะการจองได้อย่างสะดวกและรวดเร็ว
4. เพื่อพัฒนาระบบฐานข้อมูลที่สามารถจัดการใบเสร็จค่าน้ำค่าไฟได้อย่างเป็นระบบ ช่วยลดข้อผิดพลาดในการคำนวณค่าใช้จ่าย ความสะดวกให้กับทั้งผู้ดูแลและผู้ให้บริการ
5. เพื่อเป็นแนวทางในการพัฒนา ปรับปรุงระบบฐานข้อมูลการจองห้องพักให้สามารถนำไปใช้งานได้จริง สอดคล้องกับความต้องการของผู้ดูแลห้องพักและผู้ให้บริการในอนาคต

1.3 ประโยชน์ที่คาดว่าจะได้รับ

1. ลดข้อผิดพลาดในการจอง ระบบจะปรับข้อมูลห้องพักให้เป็นปัจจุบันทันที ลดปัญหาการจองซ้ำซ้อนและข้อมูลผิดพลาด
2. เพิ่มประสิทธิภาพในการดำเนินงาน ลดภาระของพนักงาน ลดเวลาการตอบคำถามซ้ำ ๆ ทำให้ธุรกิจดำเนินไปอย่างมีประสิทธิภาพมากขึ้น
3. ยกระดับประสบการณ์ของลูกค้า ลูกค้าสามารถจอง ตรวจสอบห้องพัก และชำระเงินได้อย่างสะดวก รวดเร็ว

1.4 ขอบเขตการศึกษา

1. การออกแบบและพัฒนาโครงสร้างฐานข้อมูล ใช้ ER Diagram และ SQL เพื่อสร้างฐานข้อมูลที่จัดเก็บข้อมูลห้องพัก ลูกค้า และการจองอย่างเป็นระบบ
2. การพัฒนาและทดสอบระบบต้นแบบ สร้างระบบที่รองรับการจอง ตรวจสอบห้องพัก การอัปเดตสถานะห้องพักแบบเรียลไทม์ พร้อมทดสอบเพื่อปรับปรุงประสิทธิภาพ
3. ข้อจำกัดของโครงการ ระบบพัฒนาสำหรับที่พักเพียงแห่งเดียว และยังไม่รองรับการเชื่อมต่อกับแพลตฟอร์ม OTA หรือระบบการจองจากหลายช่องทาง

บทที่ 2

ทฤษฎีพื้นฐานและเนื้อหาที่เกี่ยวข้อง

2.1 ทฤษฎีพื้นฐานที่เกี่ยวข้อง

2.1.1 ระบบฐานข้อมูล(Database System)

ระบบฐานข้อมูล(Database) คือ กลุ่มของข้อมูลที่มีความสัมพันธ์กันหรือเกี่ยวข้องกันอย่างมีระบบ ความสัมพันธ์ข้อมูลต่าง ๆ ที่ชัดเจน เช่น กลุ่มข้อมูลที่เกี่ยวข้องกับห้องพักและผู้เช่า เช่น ชื่อ นามสกุล อายุ ที่อยู่ เบอร์โทรศัพท์ และข้อมูลต่าง ๆ ที่เกี่ยวข้องกับห้องพัก เป็นต้น ข้อมูลดังกล่าวถูกเก็บอยู่รวมกันหลาย ๆ กลุ่ม ซึ่งอาจจะเก็บอยู่ใน รูปแบบแฟ้มเอกสารหรือแฟ้มข้อมูลในคอมพิวเตอร์

ประโยชน์ของระบบฐานข้อมูล ฐานข้อมูลจะช่วยสร้างระบบการจัดเก็บข้อมูลขององค์กรให้เป็นระเบียบ แยกแยะข้อมูลตามประเภททำให้ข้อมูลประเภทเดียวกันจัดเก็บอยู่ด้วยกัน สามารถค้นหาและ เรียกใช้ ได้ง่าย ซึ่งทั้งนี้ขึ้นอยู่กับการนำมาใช้ประโยชน์ขององค์กรนั้น ๆ จากประโยชน์ของระบบฐานข้อมูลด้วย คอมพิวเตอร์ ข้างต้น อาจกล่าวได้ว่าระบบฐานข้อมูลในคอมพิวเตอร์มีข้อดีมากกว่าการเก็บข้อมูลใน รูปแบบแฟ้มเอกสาร ดังนี้

1. หลีกเลี่ยงความขัดแย้งของข้อมูลได้
2. สามารถใช้ข้อมูลร่วมกัน
3. สามารถลดความซ้ำซ้อนของข้อมูล
4. สามารถกำหนดความเป็นมาตรฐานเดียวกัน
5. สามารถกำหนดระบบรักษาความปลอดภัยของข้อมูลได้
6. การรักษาความถูกต้องน่าเชื่อถือของข้อมูลได้
7. ความเป็นอิสระของข้อมูล

2.1.2 ER-Diagram

ER Diagram คือแบบจำลองที่ใช้อธิบายโครงสร้างของฐานข้อมูลซึ่งเขียนออกมาในลักษณะ ของรูปแบบ การอธิบายโครงสร้างและความสัมพันธ์ของข้อมูล (Relationship) ประกอบด้วย

1. เอนทิตี (Entity) เป็นวัตถุหรือสิ่งของที่เราสงสัยในระบบงานนั้นๆ
2. แอททริบิวต์ (Attribute) เป็นคุณสมบัติของวัตถุที่เราสงสัย
3. ความสัมพันธ์ (Relationship) คือ ความสัมพันธ์ระหว่างเอนทิตีต่าง ๆ
4. ER Diagram มีความสำคัญต่อการพัฒนาระบบงานฐานข้อมูล Application ต่าง ๆ ที่ต้องการเก็บข้อมูลอย่างมีระบบ มีโครงสร้าง ดังนั้น ER Diagram จึงใช้เป็นเอกสารในการสื่อสารระหว่างนักออกแบบระบบและนักพัฒนาระบบ เพื่อให้สื่อสารได้ตรงกันและเป็นสากล

2.1.3 SQL (Structured Query Language)

SQL คือภาษาที่ใช้ในการจัดการฐานข้อมูล ซึ่งได้รับการยอมรับมากภาษาหนึ่ง และได้รับการยอมรับในมาตรฐาน American National Standards Institute (ANSI) SQL สามารถใช้งานร่วมกับเว็บไซต์ ระบบฐานข้อมูล SQL Server ไปจนถึงการสร้างระบบวิเคราะห์ข้อมูลด้วยตนเอง

2.1.4 XAMPP

XAMPP คือโปรแกรมสำหรับจำลองเครื่องคอมพิวเตอร์ส่วนบุคคลของเรา ให้ทำงานในลักษณะของ WEBSERVER นั่นคือเครื่องคอมพิวเตอร์จะเป็นเครื่องแม่และเครื่องลูกในเครื่องเดียวกัน ทำให้ไม่ต้องเชื่อมกับ Internet ก็สามารถตรวจสอบเว็บไซต์ที่สร้างขึ้นได้ทุกที่ทุกเวลา

โปรแกรม Xampp จะมาพร้อมกับ

- PHP ภาษาสำหรับพัฒนาเว็บแอปพลิเคชันที่เป็นที่นิยม
- MySQL ฐานข้อมูล - Apache ทำหน้าที่เป็นเว็บเซิร์ฟเวอร์
- Perl มาพร้อมกับ OpenSSL
- phpMyadmin ระบบบริหารฐานข้อมูลที่พัฒนาโดย PHP เพื่อใช้เชื่อมต่อไปยังฐานข้อมูล สนับสนุนฐานข้อมูล MySQL และ SQLite

2.1.5 Visual Studio Code

Visual Studio Code หรือ VSCode เป็นโปรแกรม Code Editor ที่ใช้ในการแก้ไขและปรับแต่งโค้ด จากค่ายไมโครซอฟท์ มีการพัฒนาออกมาในรูปแบบของ OpenSource จึงสามารถนำมาใช้งานได้แบบฟรี ๆ ที่ต้องการความเป็นมืออาชีพ ซึ่ง Visual Studio Code นั้นเหมาะสำหรับนักพัฒนาโปรแกรมที่ต้องการใช้งานข้ามแพลตฟอร์ม รองรับการใช้งานทั้งบน Windows, macOS และ Linux สนับสนุนทั้งภาษา JavaScript, TypeScript และ Node.js สามารถเชื่อมต่อกับ Git ได้ นำมาใช้งานได้ง่ายไม่ซับซ้อน

2.2 โมเดลทางธุรกิจ

Key Partners	Key Activities	Value Proposition	Customer Relationships	Customer Segments
<ul style="list-style-type: none"> แพลตฟอร์มให้เช่าห้องพัก บริษัทอินเทอร์เน็ตและสาธารณูปโภค ธนาคารและสถาบันการเงิน บริษัทรับเหมาก่อสร้างและซ่อมบำรุง บริษัทรักษาความปลอดภัย บริษัทประกันภัย 	<ul style="list-style-type: none"> การบริหารจัดการอพาร์ทเมนต์ : การจัดการห้องพัก การจองห้องพัก การดูแลบำรุงรักษาอาคาร : ซ่อมแซม ทำความสะอาด ปรับปรุงพื้นที่ การทำสัญญาและต่อสัญญาเช่า : กระบวนการทำสัญญา ตรวจสอบประวัติผู้เช่า การเก็บค่าเช่าและค่าใช้จ่ายอื่น ๆ : ระบบการชำระเงิน การติดตามค่าเช่าค้างชำระ การให้บริการลูกค้า : การแก้ไขปัญหา การรับข้อร้องเรียน การตลาดและโปรโมชั่นห้องพัก 	<ul style="list-style-type: none"> ราคาคุ้มค่า กับการเช่าคอนโดหรืออพาร์ทเมนต์ เงื่อนไขสัญญาที่ยืดหยุ่น สิ่งอำนวยความสะดวก ความปลอดภัย ทำเลดี ไม่มีค่าใช้จ่ายแอบแฝง บริการที่ดี พนักงานให้บริการที่ดี การซ่อมแซมรวดเร็ว 	<ul style="list-style-type: none"> ระบบจองห้องออนไลน์ การให้บริการส่วนบุคคล โปรแกรมสะสมแต้ม หรือส่วนลด สำหรับลูกค้าที่ต่อสัญญา กิจกรรมสำหรับลูกค้า เช่น งานเลี้ยง, Workshop, Group Chat สำหรับผู้เช่า บริการหลังการขายที่ดี แก้ไขปัญหาไว มีทีมช่างซ่อมพร้อมให้บริการ สัญญาเช่า ชัดเจน เงื่อนไขเป็นธรรม 	<ul style="list-style-type: none"> นักศึกษา คนทำงาน คู่รักหรือครอบครัวเล็ก คนโสด คนที่ทำงานคนเดียว ต้องการห้องพัก ชาวต่างชาติ
		Key Resources	Channels	
		<ul style="list-style-type: none"> อาคารและห้องพักที่มีคุณภาพ ทีมบริหารและพนักงานดูแลห้องพัก ระบบจองและบริหารจัดการห้องพัก ระบบรักษาความปลอดภัย พันธมิตร เช่น ร้านค้า, ผู้ให้บริการอินเทอร์เน็ต เงินทุน เงินทุนหมุนเวียนสำหรับการบำรุงรักษาและปรับปรุง ใบอนุญาตต่าง ๆ 	<ul style="list-style-type: none"> เว็บไซต์ของห้องพักเอง Facebook Page และโซเชียลออนไลน์ แพลตฟอร์มให้เช่าห้องพัก โบรชัวร์และป้ายประกาศในมหาวิทยาลัยหรือสำนักงานใกล้เคียง การบอกต่อจากผู้เช่าปัจจุบัน โปรแกรมแนะนำเพื่อน ส่วนลดสำหรับผู้เช่าที่แนะนำลูกค้าใหม่ 	
Cost Structure		Revenue Streams		
<ul style="list-style-type: none"> ต้นทุนที่ดินและอาคาร ค่าบำรุงรักษาและซ่อมแซมห้องพัก ค่าใช้จ่ายด้านการตลาดและโฆษณา ค่าพนักงาน (แอดมิน, ไรป., แม่บ้าน) 		<ul style="list-style-type: none"> ค่าเช่ารายเดือน/รายปี ค่ามัดจำห้อง (คืนเมื่อออกจากห้องพัก) รายได้จากค่าส่วนกลาง ค่าทำความสะอาดพื้นที่ส่วนรวม รายได้จากบริการเสริม ค่าซักผ้า ค่าทำความสะอาดห้อง รายได้จากพื้นที่เชิงพาณิชย์ เครื่องซักผ้าหยอดเหรียญ, ตู้กดน้ำ, ร้านค้าในห้องพัก 		

ความต้องการของผู้ใช้งาน

- สามารถเก็บข้อมูลของผู้เช่าและข้อมูลสัญญาเช่า
- สามารถเก็บข้อมูลการติดต่อและการแจ้งปัญหาของผู้เช่า
- สามารถแสดงรายละเอียดข้อมูลและสถานะของห้องพัก
- สามารถแสดงรายละเอียดประวัติการเข้าพัก
- สามารถคำนวณค่าเช่าและค่าน้ำค่าไฟ

บทที่ 3

วิธีดำเนินงานทางธุรกิจ

3.1 ER-Diagram

3.1.1 กำหนดเอนทิตี(Entity)

Entity “Tenant”

Entity “Booking”

Entity “Room”

Entity “MaintenanceRequest”

Entity “Payment”

Entity “UtilityUsage”

Entity “Lease Agreement”

Entity “Staff”

3.1.2 Business Process

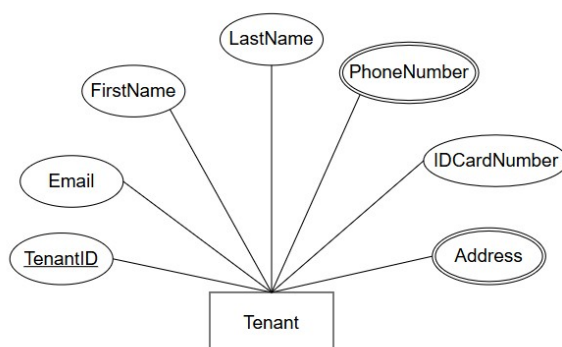
ระบบการจองห้องพัก

3.1.3 Requirement

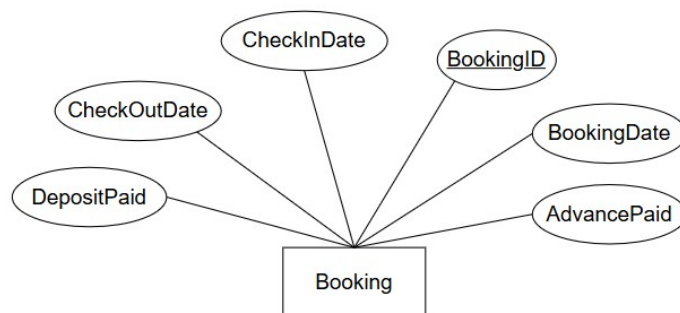
ระบบฐานข้อมูลที่ต้องการเก็บข้อมูล

3.1.4 การกำหนดแอททริบิวต์(Attribute)

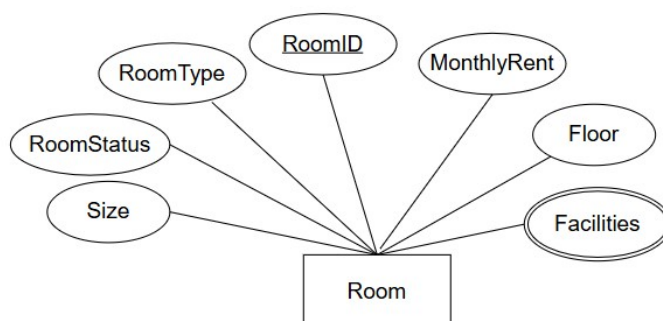
1.กำหนด Attribute ของ Entity “Tenant”



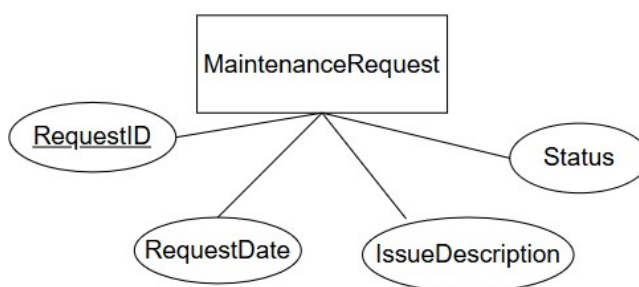
2.กำหนด Attribute ของ Entity “Booking”



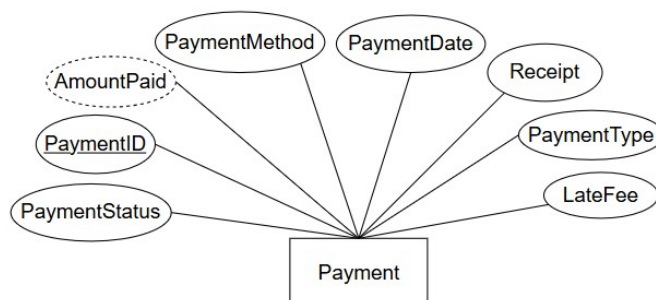
3.กำหนด Attribute ของ Entity “Room”



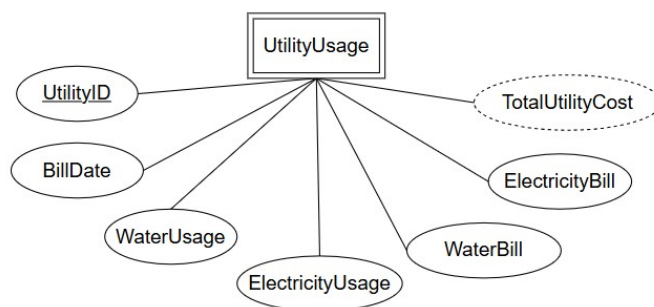
4.กำหนด Attribute ของ Entity “MaintenanceRequest”



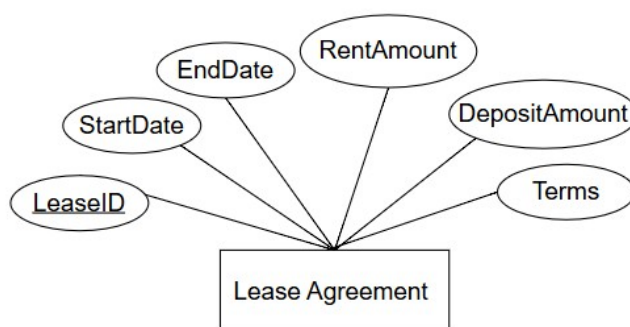
5.กำหนด Attribute ของ Entity “Payment”



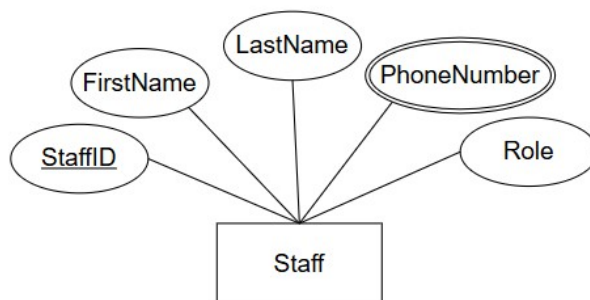
6.กำหนด Attribute ของ Entity “UtilityUsage”



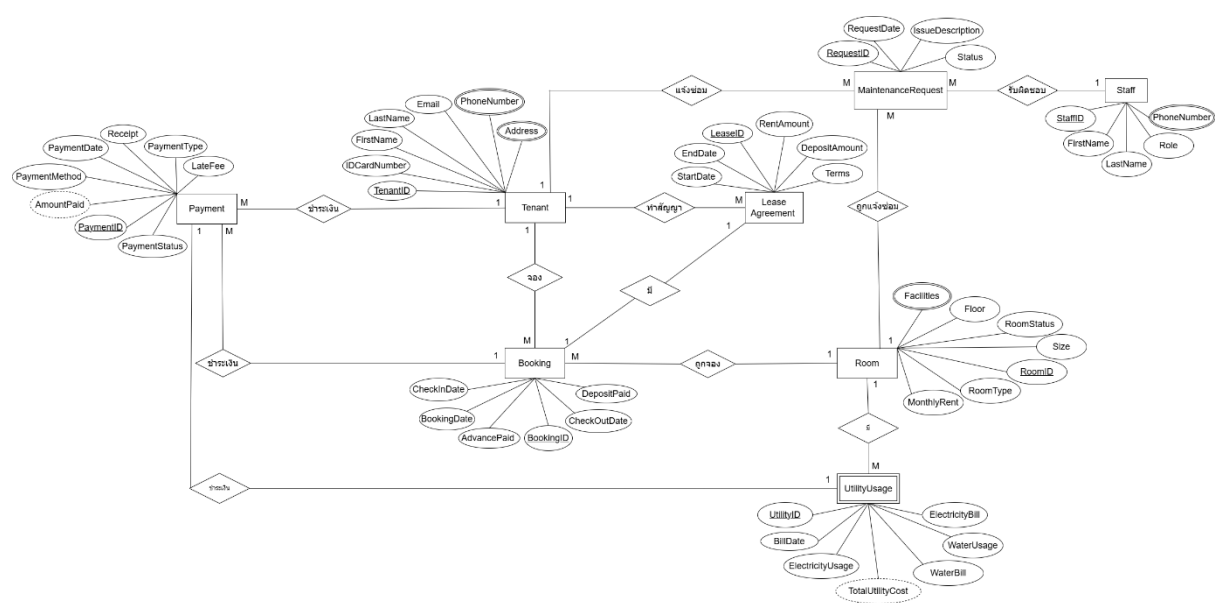
7.กำหนด Attribute ของ Entity “Lease Agreement”



8.กำหนด Attribute ของ Entity “Staff”



3.1.5 ER-Diagram

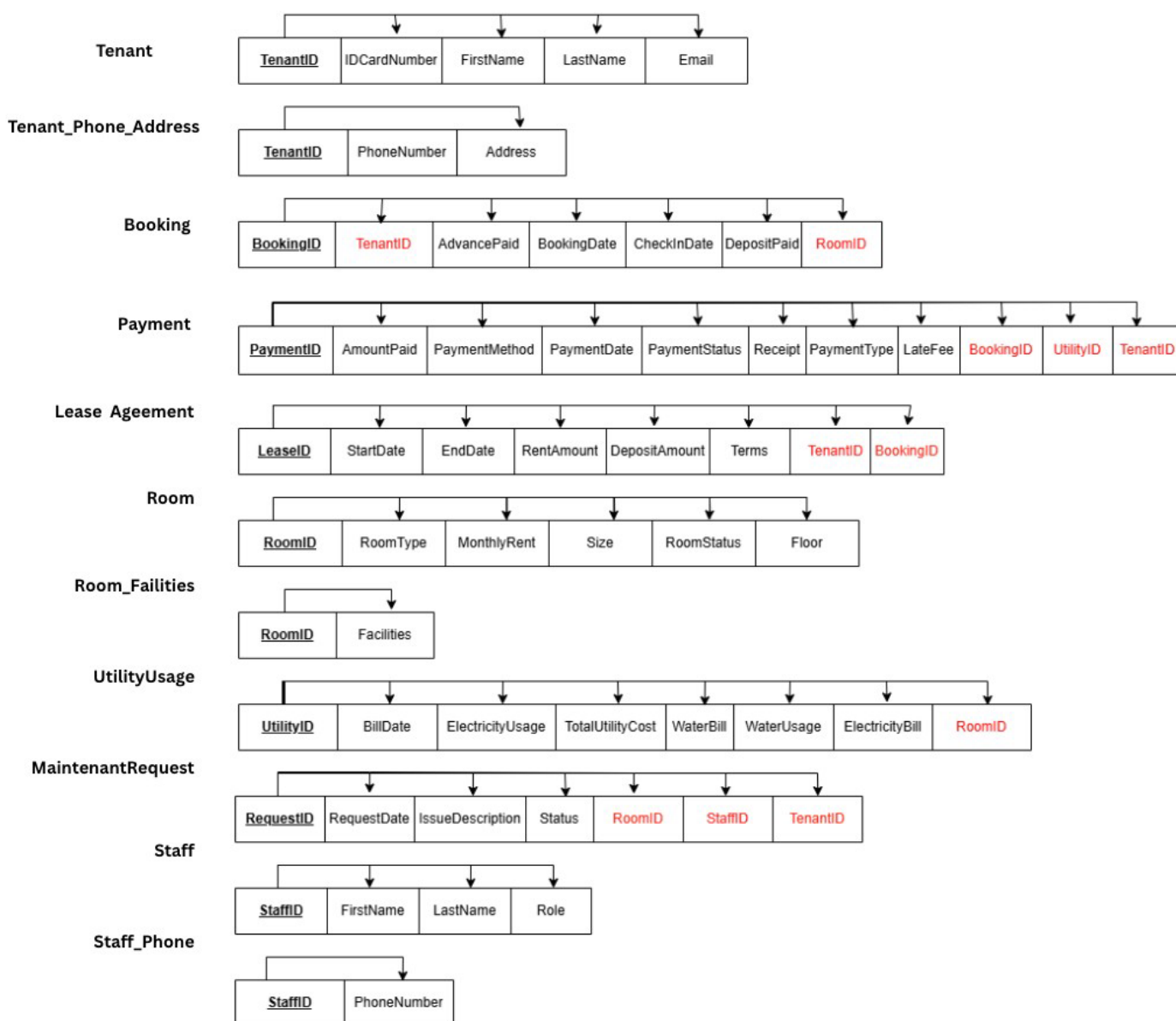


บทที่ 4

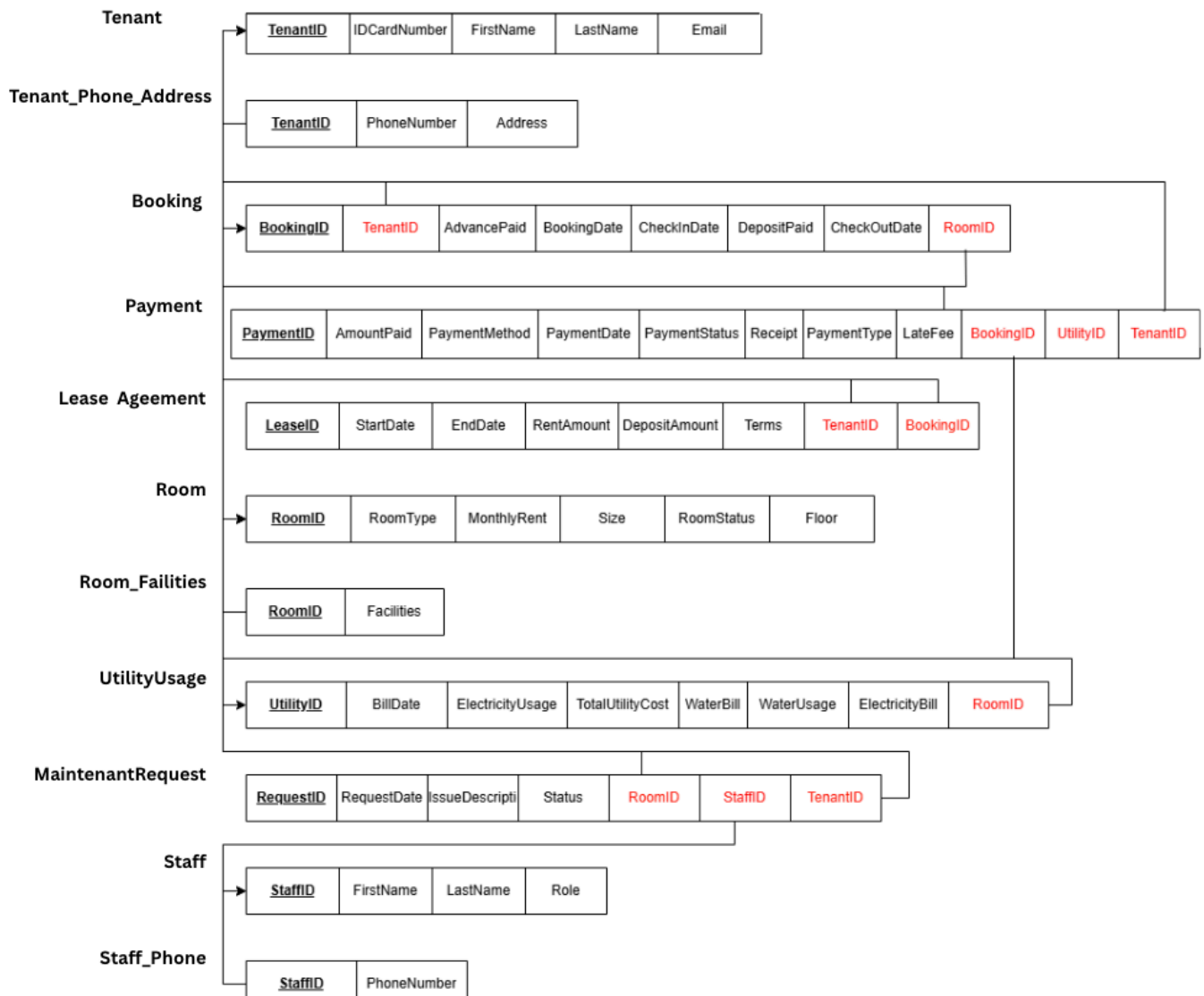
Normalization

Normalization คือการใช้ในการตรวจสอบและแก้ไขปัญหาค่าความซ้ำซ้อนของข้อมูล โดยดำเนินการให้แต่ละข้อมูลใน Relation อยู่ในรูปที่เป็นหน่วยเล็กที่สุดไม่สามารถแตกเป็นหน่วยย่อยได้ โดยยังคงความสัมพันธ์ระหว่างข้อมูลใน Relation ต่าง ๆ ไว้ในหลักการที่กำหนดใน Relation Mode

4.1 Relation Schema



4.2 Dependency Diagram



4.3 Schema Report For Database

Table : Tenant

Name	Data Type	Key	Default	Null	Description
TenantID	Int(11)	PK	None	No	รหัสผู้เช่า
IDCardNumber	Varchar(13)		None	No	เลขประจำตัวประชาชน
FirstName	Varchar(15)		None	No	ชื่อ
LastName	Varchar(15)		None	No	นามสกุล
Email	Varchar(25)		None	No	อีเมลล์
Address	Varchar(150)		None	No	ที่อยู่
PhoneNumber	Varchar(10)		Null	Yes	เบอร์โทรศัพท์

ตารางที่ 1 แสดงข้อมูล “ ผู้เช่า(Tenant) ”

Table: Booking

Name	Data Type	Key	Default	Null	Description
BookingID	Int(11)	PK	None	No	รหัสการจองห้องพัก
TenantID	Int(11)	FK	None	No	รหัสผู้เช่า
RoomID	Int(11)	FK	None	No	รหัสห้องพัก
DepositPaid	Decimal(7,2)		None	No	เงินมัดจำ
AdvancePaid	Decimal(7,2)		None	No	ค่าเช่าล่วงหน้า
BookingDate	Date		Curdate()	No	วันที่จองห้องพัก
CheckInDate	Date		None	No	วันที่เข้าพัก
CheckOutDate	Date		None	No	วันที่ย้ายออก

ตารางที่ 2 แสดงข้อมูล “ การจองห้องพัก(Booking) ”

Table : Room

Name	Data type	Key	Default	Null	Description
RoomID	Int(11)	PK	None	No	รหัสห้องพัก
RoomType	Varchar(15)		None	No	ประเภทห้อง
Size	Decimal(5,2)		None	No	ขนาดห้อง
Floor	Int(2)		None	No	ชั้น
RoomStatus	Varchar(15)		None	No	สถานะห้องพัก
Facilities	Varchar(50)		None	No	สิ่งอำนวยความสะดวก
MonthlyRent	Decimal(10,2)		None	No	ค่าเช่ารายเดือน

ตารางที่ 3 แสดงข้อมูล “ห้องพัก(Room)”

Table : Payment

Name	Data Type	Key	Default	Null	Description
PaymentID	Int(11)	PK	None	No	รหัสการชำระเงิน
TenantID	Int(11)	FK	None	No	รหัสผู้เช่า
BookingID	Int(11)	FK	None	No	รหัสการจอง
UtilityID	int11	FK	Null	Yes	รหัสบิลค่าน้ำ-ค่าไฟ
PaymentType	Varchar(15)		None	No	ประเภทการชำระเงิน
PaymentMethod	Varchar(15)		None	No	วิธีการชำระเงิน
PaymentDate	Date		None	No	วันที่ชำระ
AmountPaid	Decimal(7,2)		None	No	จำนวนเงินที่ชำระ
LateFee	decimal(7,2)		0	No	ค่าปรับชำระเงินล่าช้า
Receipt	Varchar(50)		Null	Yes	ใบเสร็จ
PaymentStatus	Varchar(15)		pending	No	สถานะการชำระเงิน

ตารางที่ 4 แสดงข้อมูล “ การชำระเงิน(Payment) ”

Table : Lease Agreement

Name	Data Type	Key	Default	Null	Description
LeaseID	Int(11)	PK	None	No	รหัสสัญญาเช่า
TenantID	Int(11)	FK	None	No	รหัสผู้เช่า
BookingID	Int(11)	FK	None	No	รหัสการจอง
StartDate	date		None	No	วันที่เริ่มทำสัญญา
EndDate	date		None	No	วันที่สิ้นสุดสัญญา
RentAmount	decimal(7,2)		None	No	ค่าเช่ารายเดือน
DepositAmount	decimal(7,2)		None	No	ค่านัดจำ
Terms	text		None	No	เงื่อนไข

ตารางที่ 5 แสดงข้อมูล “ สัญญาเช่า(Lease Agreement) ”

Table : UtilityUsage

Name	Data Type	Key	Default	Null	Description
UtilityID	Int(11)	PK	None	No	รหัสบิล
RoomID	Int(11)	FK	None	No	รหัสห้องพัก
WaterUsage	Decimal(7,2)		0	No	ปริมาณที่ใช้น้ำ(หน่วย)
WaterBill	Decimal(7,2)		0	No	ค่าน้ำ
ElectricityUsage	Decimal(7,2)		0	No	ปริมาณที่ใช้ไฟ(หน่วย)
ElectricityBill	Decimal(7,2)		0	No	ค่าไฟ
BillDate	date		None	No	วันที่ที่เรียกเก็บ
TotalUtilityCost	Decimal(7,2)		None	No	ค่าใช้จ่ายรวม

ตารางที่ 6 แสดงข้อมูล “ ค่าน้ำ-ค่าไฟ(Utility Usage)”

Table : MaintenanceRequest

Name	Data Type	Key	Default	Null	Description
RequestID	Int(11)	PK	None	No	รหัสการแจ้งซ่อม
TenantID	Int(11)	FK	None	No	รหัสผู้เช่า
RoomID	Int(11)	FK	None	No	รหัสห้อง
StaffID	Int(11)	FK	None	No	รหัสพนักงาน
RequestDate	Date		None	No	วันที่แจ้งซ่อม
IssueDescription	Text		None	No	รายละเอียดปัญหา
Status	Char(30)		pending	No	สถานะ

ตารางที่ 7 แสดงข้อมูล “ การแจ้งซ่อม(Maintenance Request) ”

Table : Staff

Name	Data Type	Key	Default	Null	Description
StaffID	Int(11)	PK	None	No	รหัสพนักงาน
FirstName	Varchar(15)		None	No	ชื่อ
LastName	Varchar(15)		None	No	นามสกุล
PhoneNumber	Varchar(10)		Null	yes	เบอร์โทร
Role	Varchar(15)		None	No	ตำแหน่ง

ตารางที่ 8 แสดงข้อมูล “ พนักงาน(Staff)”

4.4 Normalization

First Normalization(1NF)

เนื่องจากไม่มีกลุ่มข้อมูลซ้ำและทุกแอททริบิวต์ขึ้นกับแอททริบิวต์ที่เป็นคีย์หลักอย่างสมบูรณ์ จึงจัดอยู่ในรูปแบบ First Normal Form หรือ (1NF) เรียบร้อยแล้ว

Second Normalization(2NF)

รีเลชั่นอยู่ในรูปแบบ 1NF ไม่มีการขึ้นต่อกันเพียงบางส่วน(Partial dependencies) คือไม่มีแอททริบิวต์ที่ไม่ใช่ คีย์หลัก จึงจัดอยู่ในรูปแบบ Second Normal Form หรือ (2NF) เรียบร้อยแล้ว

Third Normalization(3NF)

รีเลชั่นอยู่ในรูปแบบ 2NF ไม่มีการขึ้นแบบทรานซิทีฟ(Transitive dependencies) คือไม่มีแอททริบิวต์ที่ไม่ใช่คีย์ใดขึ้นกับแอททริบิวต์อื่น ซึ่งเป็นแอททริบิวต์ที่ไม่ใช่คีย์เดียวกัน จึงจัดอยู่ในรูปแบบ Third Normal Form หรือ (3NF) เรียบร้อยแล้ว

ภาคผนวก


```

1 <?php
2 $host = 'localhost';
3 $dbname = 'dormitory_app';
4 $username = 'root';
5 $password = '';
6
7 try {
8     $pdo = new PDO("mysql:host=$host;dbname=$dbname", $username, $password);
9     $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
10 } catch (PDOException $e) {
11     echo "Connection failed: " . $e->getMessage();
12 }
13
14

```

The Explorer panel on the left shows the project structure for 'DORMITORY_APP', including folders like 'images', 'receipts', 'tenant', 'uploads', and various PHP files like 'contact.php', 'db.php', 'dormitory_app.sql', 'header.php', 'index.php', 'login_admin.php', 'login_user.php', 'login.php', 'loginuser_or_admin.php', 'register_admin.php', 'register_user.php', 'reset_password.php', 'scriptต่างๆ', 'sql_nologin.sql', 'style.css', and 'update_cod.sql'. The status bar at the bottom indicates 'Ln 14, Col 1', 'Spaces: 4', 'UTF-8', 'CRLF', and 'PHP'.

เชื่อมต่อฐานข้อมูล

```

1 <?php
2 session_start();
3 require_once('db.php');
4
5 if ($_SERVER["REQUEST_METHOD"] == "POST") {
6     try {
7         $username = $_POST['username'] ?? '';
8         $password = $_POST['password'] ?? '';
9         $email = $_POST['email'] ?? '';
10        $phone = $_POST['phone'] ?? '';
11        $role = 'tenant';
12        if (!isset($_POST['confirmPassword']) || $_POST['confirmPassword'] !== $password) {
13            throw new Exception("Passwords do not match!");
14        }
15        $username = htmlspecialchars($username, ENT_QUOTES, 'UTF-8');
16        $email = htmlspecialchars($email, ENT_QUOTES, 'UTF-8');
17        $phone = htmlspecialchars($phone, ENT_QUOTES, 'UTF-8');
18
19        $pdo->beginTransaction();
20
21        $stmt = $pdo->prepare("SELECT COUNT(*) FROM Users WHERE Username = ? OR Email = ?");
22        $stmt->execute([$username, $email]);
23        if ($stmt->fetchColumn() > 0) {
24            throw new Exception("Username or Email already exists!");
25        }
26
27        $passwordHash = password_hash($password, PASSWORD_BCRYPT);
28
29        $stmt = $pdo->prepare("INSERT INTO Users (Username, PasswordHash, Email, Phone) VALUES (?, ?, ?, ?)");
30

```

The Explorer panel on the left shows the project structure for 'DORMITORY_APP', including folders like 'tenant', 'utility_bills.php', 'uploads', and various PHP files like 'contact.php', 'db.php', 'dormitory_app.sql', 'header.php', 'index.php', 'login_admin.php', 'login_user.php', 'login.php', 'register_admin.php', 'register_user.php', 'reset_password.php', 'scriptต่างๆ', 'sql_nologin.sql', 'style.css', 'update_cod.sql', 'update_tenant_users.sql', and 'update.sql'. The status bar at the bottom indicates 'Ln 3, Col 25', 'Spaces: 4', 'UTF-8', 'LF', and 'PHP'.

```

1  <?php
2  session_start();
3  require_once('db.php');
4
5
6  if ($_SERVER["REQUEST_METHOD"] == "POST") {
7      $username = $_POST['username'];
8      $password = $_POST['password'];
9      $confirmPassword = $_POST['confirmPassword'];
10     $email = $_POST['email'];
11     $phone = $_POST['phone'];
12
13     if ($password !== $confirmPassword) {
14         $errorMessage = "Passwords do not match!";
15     } else {
16         $passwordHash = password_hash($password, PASSWORD_BCRYPT);
17         $username = htmlspecialchars($username, ENT_QUOTES, 'UTF-8');
18         $email = htmlspecialchars($email, ENT_QUOTES, 'UTF-8');
19         $phone = htmlspecialchars($phone, ENT_QUOTES, 'UTF-8');
20         $stmt = $pdo->prepare("INSERT INTO Admin (Username, PasswordHash, Email, Phone, Status) VALUES (?, ?, ?, ?, ?)");
21         $stmt->execute([$username, $passwordHash, $email, $phone]);
22         $successMessage = "Admin Registration successful!";
23     }
24 }
25
26 <DOCTYPE html>
27 <html lang="en">
28 <head>

```

```

4
5  if ($_SERVER["REQUEST_METHOD"] == "POST") {
6      $username = $_POST['username'];
7      $password = $_POST['password'];
8
9      $username = htmlspecialchars($username, ENT_QUOTES, 'UTF-8');
10
11     $stmt = $pdo->prepare("SELECT * FROM Users WHERE Username = ?");
12     $stmt->execute([$username]);
13     $user = $stmt->fetch();
14
15     if ($user && password_verify($password, $user['PasswordHash'])) {
16         $_SESSION['userID'] = $user['UserID'];
17         $_SESSION['username'] = $user['Username'];
18
19         header('Location: tenant/room_booking.php');
20         exit;
21     } else {
22         $errorMessage = "Invalid username or password!";
23     }
24 }
25
26 <DOCTYPE html>
27 <html lang="en">
28 <head>
29
30     <meta charset="UTF-8">
31     <meta name="viewport" content="width=device-width, initial-scale=1.0">
32     <link rel="stylesheet" href="style.css">

```

```

4
5 if ($_SERVER["REQUEST_METHOD"] == "POST") {
6     $username = $_POST['username'];
7     $password = $_POST['password'];
8
9     $username = htmlspecialchars($username, ENT_QUOTES, 'UTF-8');
10
11     $stmt = $pdo->prepare("SELECT * FROM Users WHERE Username = ?");
12     $stmt->execute([$username]);
13     $user = $stmt->fetch();
14
15     if ($user && password_verify($password, $user['PasswordHash'])) {
16         $_SESSION['userID'] = $user['UserID'];
17         $_SESSION['username'] = $user['Username'];
18
19         header('Location: tenant/room_booking.php');
20         exit;
21     } else {
22         $errorMessage = "Invalid username or password!";
23     }
24 }
25
26
27 <!DOCTYPE html>
28 <html lang="en">
29 <head>
30     <meta charset="UTF-8">
31     <meta name="viewport" content="width=device-width, initial-scale=1.0">
32     <link rel="stylesheet" href="style.css">

```

ส่วนของหน้าแอดมิน(จัดการฐานข้อมูล)

```

75 <body>
76     <div class="content">
77         <table border="1" cellpadding="10" cellspacing="0" style="width: 100%; margin-top: 20px;">
78             <thead>
79                 <tr>
80                     <th>BookingID</th>
81                     <th>RoomID</th>
82                     <th>TenantID</th>
83                     <th>BookingDate</th>
84                     <th>CheckInDate</th>
85                     <th>CheckOutDate</th>
86                     <th>DepositPaid</th>
87                     <th>AdvancePaid</th>
88                     <th>TotalAmount</th>
89                     <th>PaymentStatus</th>
90                 </tr>
91             </thead>
92             <tbody>
93                 <?php foreach ($bookings as $booking): ?>
94                     <tr>
95                         <td><?php echo htmlspecialchars($booking['BookingID']); ?></td>
96                         <td><?php echo htmlspecialchars($booking['RoomID']); ?></td>
97                         <td><?php echo htmlspecialchars($booking['TenantID']); ?></td>
98                         <td><?php echo date('d/m/Y', strtotime($booking['BookingDate'])); ?></td>
99                         <td><?php echo date('d/m/Y', strtotime($booking['CheckInDate'])); ?></td>
100                        <td><?php echo date('d/m/Y', strtotime($booking['CheckOutDate'])); ?></td>
101                        <td><?php echo number_format($booking['DepositPaid'], 2); ?></td>
102                        <td><?php echo number_format($booking['AdvancePaid'], 2); ?></td>
103                        <td><?php echo number_format($booking['TotalAmount'], 2); ?></td>
104                        <td>
105                            <span class="status-badge <?php echo strtolower($booking['PaymentStatus']); ?>">
106                                <?php echo $booking['PaymentStatus']; ?>
107                            </span>
108                            <!-- ลบไปยังการชำระเงิน -->
109                        </td>
110                    </tr>
111                    <!-- Added Actions buttons -->
112                    <tr>
113                        <td colspan="9">
114                            <a href="edit_booking.php?id=<?php echo $booking['BookingID']; ?>" class="action-b
115                            <a href="booking_manage.php?id=<?php echo $booking['BookingID']; ?>"
116                                class="action-btn"
117                                onclick="return confirm('Are you sure you want to delete this booking?');">Delete
118                        </td>
119                    </tr>
120                </tr>
121            </tbody>
122        </table>
123    </div>
124
125
126
127
128
129
130
131
132
133
134
135

```



```

1  if (!isset($_GET['id'])) {
2      exit();
3  }
4
5  $stmt = $pdo->prepare("SELECT BookingID FROM lease_agreement WHERE LeaseID = ?");
6  $stmt->execute([$_GET['id']]);
7  $lease = $stmt->fetch();
8
9  if ($lease) {
10     $stmt = $pdo->prepare("DELETE FROM lease_agreement WHERE LeaseID = ?");
11     $stmt->execute([$_GET['id']]);
12 } else {
13     throw new Exception('Lease agreement not found.');
```

```

1  if (!isset($_SESSION['adminID'])) {
2      exit();
3  }
4
5  $stmt = $pdo->prepare("
6  SELECT
7      mr.RequestID,
8      mr.TenantID,
9      mr.RoomID,
10     mr.RequestDate,
11     mr.IssueDescription,
12     mr.Status,
13     mr.StaffID
14 FROM MaintenanceRequest mr
15 ORDER BY mr.RequestID DESC
16 ");
17 $stmt->execute();
18 $requests = $stmt->fetchAll();
19
20 <?php
21 <doctype html>
22 <html lang="en">
23 <head>
24     <meta charset="UTF-8">
25     <meta name="viewport" content="width=device-width, initial-scale=1.0">
26     <title>Maintenance Requests Management</title>
27     <link rel="stylesheet" href="style.css">
28 </head>
```

```

4  if (!isset($_SESSION['adminID'])) {
5  }
6  }
7  if (isset($_GET['delete'])) {
8      $deleteID = $_GET['delete'];
9      $stmt = $pdo->prepare("DELETE FROM Payment WHERE PaymentID = ?");
10     $stmt->execute([$deleteID]);
11     echo "<script>alert('Payment record has been deleted successfully!'); window.location.href='payment_ma";
12 }
13
14
15 $stmt = $pdo->prepare("
16     SELECT p.PaymentID, p.TenantID, p.BookingID, p.UtilityID,
17           p.PaymentType, p.AmountPaid, p.LateFee,
18           p.PaymentDate, p.PaymentMethod, p.PaymentStatus, p.Receipt,
19           DATE_FORMAT(p.PaymentDate, '%d/%m/%Y') as FormattedDate
20 FROM Payment p
21 LEFT JOIN Booking b ON p.BookingID = b.BookingID
22 LEFT JOIN utilityusage u ON p.UtilityID = u.UtilityID
23 ORDER BY p.PaymentDate DESC");
24
25 $stmt->execute();
26 $payments = $stmt->fetchAll();
27 ?>
28
29 <!DOCTYPE html>
30 <html lang="en">
31 <head>
32     <meta charset="UTF-8">
33     <meta name="viewport" content="width=device-width, initial-scale=1.0">
34     <title>Admin - Payment Management</title>
35     <link rel="stylesheet" href="style.css">

```

```

58 <body>
59     <div class="content">
60         <table>
61             <thead>
62                 <tr>
63                     <th>StaffID</th>
64                     <th>Phone</th>
65                     <th>Role</th>
66                     <th>Actions</th>
67                 </tr>
68             </thead>
69             <tbody>
70                 <?php foreach ($staffMembers as $staff): ?>
71                     <tr>
72                         <td><?php echo htmlspecialchars($staff['StaffID']); ?></td>
73                         <td><?php echo htmlspecialchars($staff['FirstName']); ?></td>
74                         <td><?php echo htmlspecialchars($staff['LastName']); ?></td>
75                         <td><?php echo htmlspecialchars($staff['Phone']); ?></td>
76                         <td><?php echo htmlspecialchars($staff['Role']); ?></td>
77                         <td>
78                             <a href="edit_staff.php?id=<?php echo $staff['StaffID']; ?>"
79                               class="action-btn edit-btn">Edit</a>
80                             <a href="staff_manage.php?delete=<?php echo $staff['StaffID']; ?>"
81                               class="action-btn delete-btn"
82                               onclick="return confirm('Are you sure you want to delete this staff member?')">Delete</a>
83                         </td>
84                     </tr>
85                 <?php endforeach; ?>
86             </tbody>
87         </table>
88     </div>
89 </body>
90 </html>

```

```

10 if (isset($_GET['delete'])) {
11     try {
12         $stmt = $pdo->prepare("DELETE FROM Tenant WHERE TenantID = ?");
13         $stmt->execute([$tenantID]);
14
15         $pdo->commit();
16         $_SESSION['success'] = 'Tenant deleted successfully';
17         header('Location: tenant_manage.php');
18         exit();
19     } catch (Exception $e) {
20         $pdo->rollBack();
21         $_SESSION['error'] = 'Error deleting tenant: ' . $e->getMessage();
22         header('Location: tenant_manage.php');
23         exit();
24     }
25 }
26
27 $stmt = $pdo->prepare("SELECT t.* FROM Tenant t ORDER BY t.TenantID DESC");
28 $stmt->execute();
29 $tenants = $stmt->fetchAll();
30
31 <?php
32 <doctype html>
33 <html lang="en">
34 <head>
35     <meta charset="UTF-8">
36     <meta name="viewport" content="width=device-width, initial-scale=1.0">
37     <title>Admin - Manage Tenants</title>

```

```

1 <?php
2 require_once('../db.php');
3 session_start();
4 if (isset($_SESSION['adminID'])) {
5     header('Location: ../login_admin.php');
6     exit();
7 }
8 //add
9 if (isset($_GET['delete'])) {
10     $deleteID = $_GET['delete'];
11     $stmt = $pdo->prepare("DELETE FROM UtilityUsage WHERE UtilityID = ?");
12     $stmt->execute([$deleteID]);
13     echo "<script>alert('Utility record has been deleted successfully!'); window.location.href='utility_ma";
14 }
15
16 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
17     // Set fixed rates
18     $waterRate = 20; // 20 baht per unit
19     $electricityRate = 8; // 8 baht per unit
20
21     $waterTotal = $_POST['water_usage'] * $waterRate;
22     $electricityTotal = $_POST['electricity_usage'] * $electricityRate;
23     $totalCost = $waterTotal + $electricityTotal;
24
25     if (isset($_POST['add'])) {
26         $stmt = $pdo->prepare("INSERT INTO UtilityUsage (RoomID, BillDate, WaterUsage, ElectricityUsage, W
27         $stmt->execute([$_POST['room_id'], $_POST['bill_date'], $_POST['water_usage'], $_POST['electricity
28         echo "<script>alert('Utility record added successfully!'); window.location.href='utility_manage.ph";
29     }

```



```

1 <?php
2 require_once('../db.php');
3 session_start();
4 if (isset($_SESSION['adminID'])) {
5     header('Location: ../login_admin.php');
6     exit();
7 }
8 //add
9 if (isset($_GET['delete'])) {
10     $deleteID = $_GET['delete'];
11     $stmt = $pdo->prepare("DELETE FROM UtilityUsage WHERE UtilityID = ?");
12     $stmt->execute([$deleteID]);
13     echo "<script>alert('Utility record has been deleted successfully!'); window.location.href='utility_ma";
14 }
15
16 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
17     // Set fixed rates
18     $waterRate = 20; // 20 baht per unit
19     $electricityRate = 8; // 8 baht per unit
20
21     $waterTotal = $_POST['water_usage'] * $waterRate;
22     $electricityTotal = $_POST['electricity_usage'] * $electricityRate;
23     $totalCost = $waterTotal + $electricityTotal;
24
25     if (isset($_POST['add'])) {
26         $stmt = $pdo->prepare("INSERT INTO UtilityUsage (RoomID, BillDate, WaterUsage, ElectricityUsage, W
27         $stmt->execute([$_POST['room_id'], $_POST['bill_date'], $_POST['water_usage'], $_POST['electricity
28         echo "<script>alert('Utility record added successfully!'); window.location.href='utility_manage.ph";
29     }

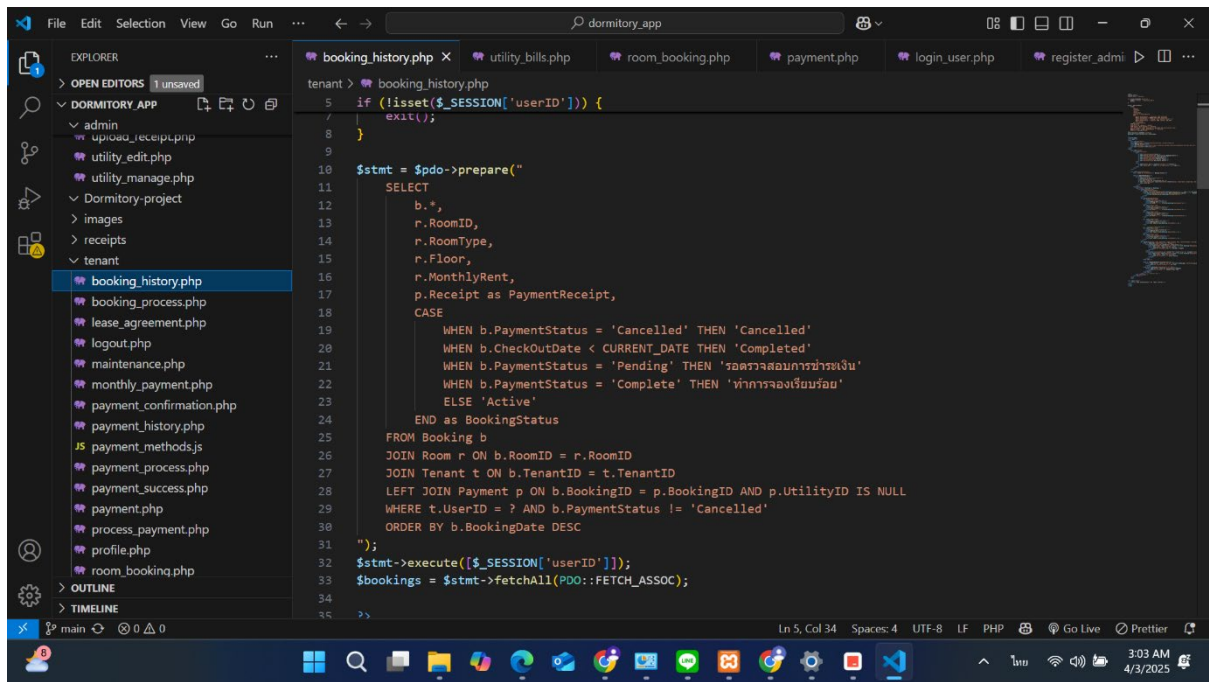
```

ส่วนของหน้าจอห้องและชำระเงินค่าเช่า

```

1 <?php
2 session_start();
3 require_once('../db.php');
4
5 if (isset($_SESSION['userID'])) {
6     header('Location: ../login_user.php');
7     exit();
8 }
9
10 $stmt = $pdo->prepare("
11     SELECT r.RoomID, r.RoomType, r.Floor, r.Size, r.Facilities,
12     'Available' as RoomStatus
13 FROM Room r
14 WHERE NOT EXISTS (
15     SELECT 1
16     FROM Booking b
17     WHERE b.RoomID = r.RoomID
18     AND (b.PaymentStatus = 'Paid' OR b.PaymentStatus = 'Pending' OR b.PaymentStatus = 'Completed')
19     AND b.CheckOutDate >= CURRENT_DATE
20 )
21 ORDER BY r.RoomID
22 ");
23 $stmt->execute();
24 $rooms = $stmt->fetchAll();
25
26 <!DOCTYPE html>
27 <html lang="en">
28 <head>
29     <meta charset="UTF-8">

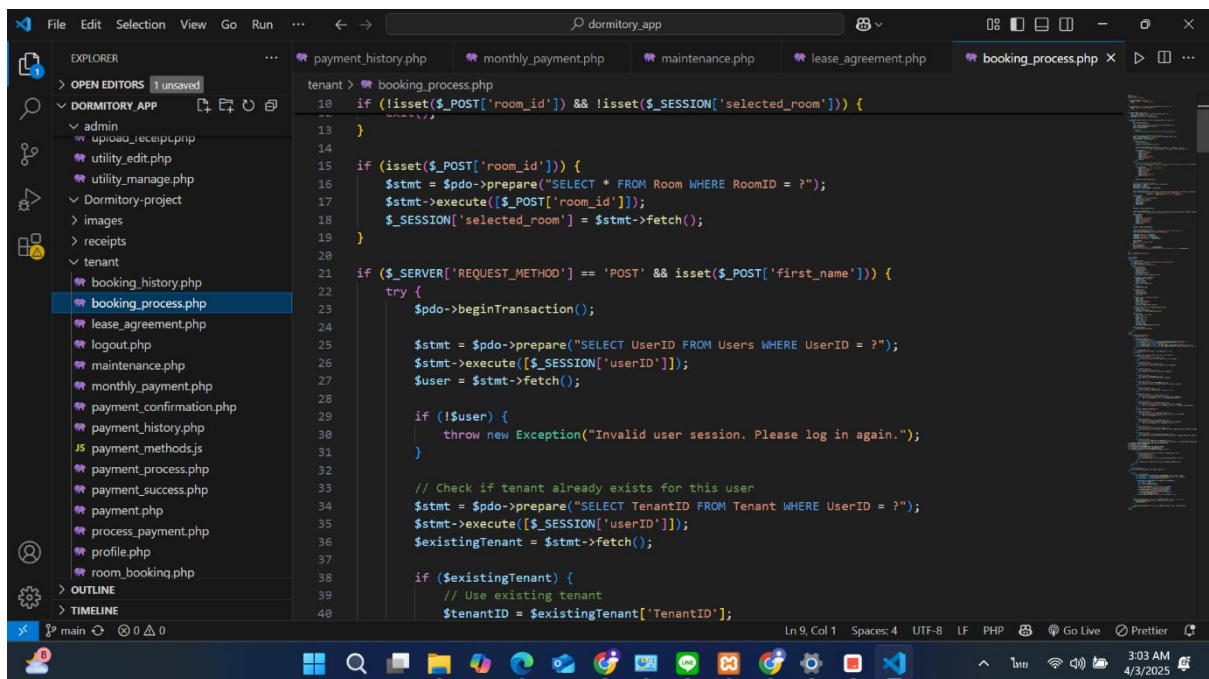
```



```

tenant > booking_history.php
5  if (!isset($_SESSION['userID'])) {
6      exit();
7  }
8
9
10 $stmt = $pdo->prepare("
11     SELECT
12         b.*,
13         r.RoomID,
14         r.RoomType,
15         r.Floor,
16         r.MonthlyRent,
17         p.Receipt as PaymentReceipt,
18         CASE
19             WHEN b.PaymentStatus = 'Cancelled' THEN 'Cancelled'
20             WHEN b.CheckOutDate < CURRENT_DATE THEN 'Completed'
21             WHEN b.PaymentStatus = 'Pending' THEN 'รอตรวจสอบการชำระเงิน'
22             WHEN b.PaymentStatus = 'Complete' THEN 'ทำการจองเรียบร้อยแล้ว'
23             ELSE 'Active'
24         END as BookingStatus
25     FROM Booking b
26     JOIN Room r ON b.RoomID = r.RoomID
27     JOIN Tenant t ON b.TenantID = t.TenantID
28     LEFT JOIN Payment p ON b.BookingID = p.BookingID AND p.UtilityID IS NULL
29     WHERE t.UserID = ? AND b.PaymentStatus != 'Cancelled'
30     ORDER BY b.BookingDate DESC
31 ");
32 $stmt->execute([$_SESSION['userID']]);
33 $bookings = $stmt->fetchAll(PDO::FETCH_ASSOC);
34
35 >>

```



```

tenant > booking_process.php
10 if (!isset($_POST['room_id']) && !isset($_SESSION['selected_room'])) {
11     // ...
12 }
13
14
15 if (isset($_POST['room_id'])) {
16     $stmt = $pdo->prepare("SELECT * FROM Room WHERE RoomID = ?");
17     $stmt->execute([$_POST['room_id']]);
18     $_SESSION['selected_room'] = $stmt->fetch();
19 }
20
21 if ($_SERVER['REQUEST_METHOD'] == 'POST' && isset($_POST['first_name'])) {
22     try {
23         $pdo->beginTransaction();
24
25         $stmt = $pdo->prepare("SELECT UserID FROM Users WHERE UserID = ?");
26         $stmt->execute([$_SESSION['userID']]);
27         $user = $stmt->fetch();
28
29         if (!$user) {
30             throw new Exception("Invalid user session. Please log in again.");
31         }
32
33         // Check if tenant already exists for this user
34         $stmt = $pdo->prepare("SELECT TenantID FROM Tenant WHERE UserID = ?");
35         $stmt->execute([$_SESSION['userID']]);
36         $existingTenant = $stmt->fetch();
37
38         if ($existingTenant) {
39             // Use existing tenant
40             $tenantID = $existingTenant['TenantID'];

```



```

tenant > lease_agreement.php
5  if (!isset($_SESSION['userID'])) {
6      exit();
7  }
8
9  // ดึงข้อมูลสัญญาเช่า
10 $sql = "SELECT l.*, l.Terms as LeaseTerms, t.FirstName, t.LastName, t.Email, t.Phone,
11         t.IDCardNumber, t.Address, r.RoomType, r.Floor, r.RoomID, b.BookingDate
12         FROM lease_agreement l
13         JOIN tenant t ON l.TenantID = t.TenantID
14         JOIN booking b ON l.BookingID = b.BookingID
15         JOIN Room r ON b.RoomID = r.RoomID
16         WHERE t.UserID = ?
17         AND b.CheckOutDate >= CURRENT_DATE
18         ORDER BY l.CreatedAt DESC";
19
20 $stmt = $pdo->prepare($sql);
21 $stmt->execute([$_SESSION['userID']]);
22 $leases = $stmt->fetchAll(PDO::FETCH_ASSOC);
23
24 <?php
25
26 <!DOCTYPE html>
27 <html lang="th">
28 <head>
29     <meta charset="UTF-8">
30     <meta name="viewport" content="width=device-width, initial-scale=1.0">
31     <title>หน้าแสดงสัญญาเช่า</title>
32     <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/css/bootstrap.min.css">
33     <link rel="stylesheet" href="style.css">
34 </head>
35 <body>

```

```

tenant > payment_history.php
3  require_once('../db.php');
4
5  if (!isset($_SESSION['userID'])) {
6      header('Location: ../login_user.php');
7      exit();
8  }
9
10 $userID = $_SESSION['userID'];
11
12 $stmt = $pdo->prepare("
13     SELECT p.PaymentID, p.AmountPaid, p.PaymentDate, p.PaymentMethod, p.Receipt, p.PaymentType,
14            b.RoomID, b.CheckInDate, b.CheckOutDate,
15            t.TenantID
16     FROM Payment p
17     LEFT JOIN Booking b ON p.BookingID = b.BookingID
18     LEFT JOIN Tenant t ON b.TenantID = t.TenantID
19     WHERE t.UserID = ?
20     ORDER BY p.PaymentDate DESC");
21
22 try {
23     $stmt->execute([$_SESSION['userID']]);
24     $payments = $stmt->fetchAll();
25 } catch (PDOException $e) {
26     echo "Error: " . $e->getMessage();
27 }
28
29 <?php
30
31 <!DOCTYPE html>

```

```

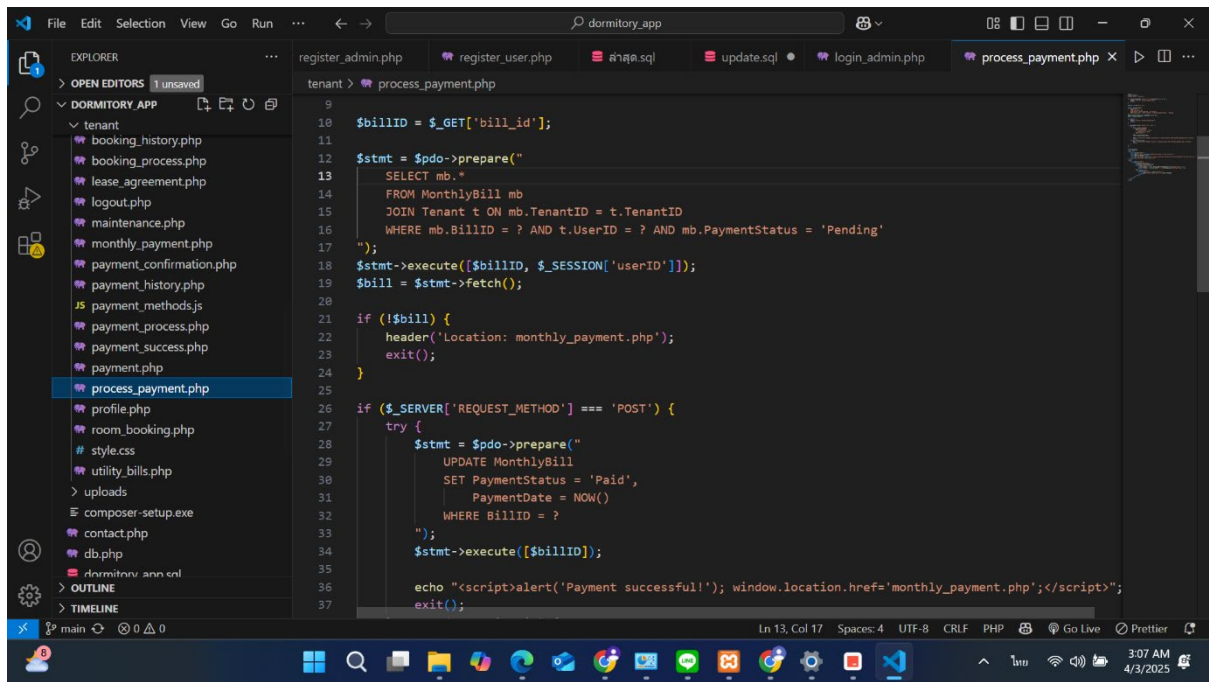
tenant > payment_process.php
5  if (isset($_SESSION['userID'])) {
6      exit();
7  }
8
9
10 $stmt = $pdo->prepare("SELECT TenantID FROM tenant WHERE UserID = ?");
11 $stmt->execute([$_SESSION['userID']]);
12 $tenant = $stmt->fetch();
13
14 if (isset($_GET['booking_id']) || isset($_GET['utility_id'])) {
15     header('Location: monthly_payment.php');
16     exit();
17 }
18
19 $stmt = $pdo->prepare("
20     SELECT
21         b.BookingID,
22         r.RoomType,
23         r.MonthlyRent,
24         u.UtilityID,
25         u.WaterUsage,
26         u.ElectricityUsage,
27         u.WaterBill,
28         u.ElectricityBill,
29         u.TotalUtilityCost,
30         u.BillDate
31     FROM booking b
32     JOIN room r ON b.RoomID = r.RoomID
33     LEFT JOIN utilityusage u ON r.RoomID = u.RoomID
34     WHERE b.BookingID = ? AND (u.UtilityID = ? OR ? IS NULL)
35     AND b.TenantID = ?

```

```

tenant > payment.php
1  <?php
2  session_start();
3  require_once('../db.php');
4
5  if (isset($_SESSION['tenant_id']) || isset($_SESSION['booking_id'])) {
6      header('Location: room_booking.php');
7      exit();
8  }
9
10 $stmt = $pdo->prepare("SELECT b.*, r.RoomType, r.MonthlyRent
11     FROM Booking b
12     JOIN Room r ON b.RoomID = r.RoomID
13     WHERE b.BookingID = ?");
14 $stmt->execute([$_SESSION['booking_id']]);
15 $booking = $stmt->fetch();
16
17 $depositPaid = 2000; // ค่ามัดจำคงที่ 2,000 บาท
18 $advancePaid = $booking['MonthlyRent']; // ค่าเช่าล่วงหน้าเท่ากับค่าเช่า 1 เดือน
19 $totalAmount = $depositPaid + $advancePaid;
20 $booking['DepositPaid'] = $depositPaid;
21 $booking['AdvancePaid'] = $advancePaid;
22 $booking['TotalAmount'] = $totalAmount;
23
24 if ($_SERVER['REQUEST_METHOD'] == 'POST') {
25     try {
26         $pdo->beginTransaction();
27
28         // Generate receipt number
29         $receipt = 'RCPT' . time() . '_' . rand(1000, 9999);

```



The screenshot shows a code editor with the following content:

```
File Edit Selection View Go Run ... < -> dormitory_app
EXPLORER
> OPEN EDITORS 1 unsaved
DORMITORY_APP
  tenant
    booking_history.php
    booking_process.php
    lease_agreement.php
    logout.php
    maintenance.php
    monthly_payment.php
    payment_confirmation.php
    payment_history.php
    payment_methods.js
    payment_process.php
    payment_success.php
    payment.php
    process_payment.php
    profile.php
    room_booking.php
    style.css
    utility_bills.php
  uploads
  composer-setup.exe
  contact.php
  db.php
  dormitory_app.sql
  OUTLINE
  TIMELINE

register_admin.php register_user.php สำนัด.sql update.sql login_admin.php process_payment.php x
tenant > process_payment.php
9
10 $billID = $_GET['bill_id'];
11
12 $stmt = $pdo->prepare("
13     SELECT mb.*
14     FROM MonthlyBill mb
15     JOIN Tenant t ON mb.TenantID = t.TenantID
16     WHERE mb.BillID = ? AND t.UserID = ? AND mb.PaymentStatus = 'Pending'
17 ");
18 $stmt->execute([$billID, $_SESSION['userID']]);
19 $bill = $stmt->fetch();
20
21 if (!$bill) {
22     header('Location: monthly_payment.php');
23     exit();
24 }
25
26 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
27     try {
28         $stmt = $pdo->prepare("
29             UPDATE MonthlyBill
30             SET PaymentStatus = 'Paid',
31                 PaymentDate = NOW()
32             WHERE BillID = ?
33         ");
34         $stmt->execute([$billID]);
35
36         echo "<script>alert('Payment successful!'); window.location.href='monthly_payment.php';</script>";
37         exit();
38     } catch (Exception $e) {
39         // Handle exception
40     }
41 }
```

Ln 13, Col 17 Spaces: 4 UTF-8 CRLF PHP Go Live Prettier

บรรณานุกรม

โปรแกรม Xampp / <https://fastclick.co.th/howto-install-xampp-windows/>

โปรแกรม Visual Studio / <https://code.visualstudio.com/>

โปรแกรมSQL/<https://www.9experttraining.com/articles/microsoft-sql-server-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3>