

<https://youtu.be/HryARB6wmL0>



## รายงานโครงการ เรื่อง เว็บจัดการห้องพักรายเดือน

### จัดทำโดย

- |                              |                       |
|------------------------------|-----------------------|
| 1. นาย วิทยศิลป์ ห้วยหงษ์ทอง | รหัสนักศึกษา 66070183 |
| 2. นาย ศุภณัฐ งามเกตสุข      | รหัสนักศึกษา 66070195 |
| 3. นาย กิตติพศ มงคลรัตน์     | รหัสนักศึกษา 66070239 |
| 4. นาย ชนกชันม์ แฝ่นคำ       | รหัสนักศึกษา 66070247 |
| 5. นาย ศรัณย์ มั่นประพันธ์   | รหัสนักศึกษา 66070309 |

เสนอ

ผศ.ดร. มนพ พันธ์โคกกรวด

รายงานนี้เป็นส่วนหนึ่งของรายวิชา

06066302 Fundamental Web Programming

คณะ เทคโนโลยีสารสนเทศ สาขา เทคโนโลยีสารสนเทศ  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

## คำนำ

รายงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาค้นคว้าและพัฒนาระบบ HavenHub – ระบบจัดการห้องพักรายเดือน ซึ่งมีวัตถุประสงค์เพื่ออำนวยความสะดวกให้กับเจ้าของที่พักและผู้เช่าในการจัดการข้อมูลห้องพัก

ผู้จัดทำหวังเป็นอย่างยิ่งว่ารายงานฉบับนี้จะเป็นประโยชน์ต่อผู้ที่สนใจศึกษาเกี่ยวกับระบบบริหารจัดการห้องพักรายเดือน รวมถึงเป็นแนวทางในการพัฒนาโครงการที่เกี่ยวข้องต่อไป หากมีข้อผิดพลาดประการใด ผู้จัดทำขออภัยรับคำแนะนำเพื่อการปรับปรุงให้ดียิ่งขึ้น

คณะผู้จัดทำ

# สารบัญ

บทที่ 1 บทนำ .....	1
ที่มาและความสำคัญ .....	1
วัตถุประสงค์ .....	1
ประโยชน์ที่คาดว่าจะได้รับ .....	1
ขอบเขตของโครงงาน .....	2
บทที่ 2 แนวคิดทฤษฎี และเอกสารที่เกี่ยวข้อง .....	3
เทคโนโลยีที่ใช้ในการพัฒนาเว็บไซต์ .....	3
2.1 HTML .....	3
2.2 CSS .....	5
2.3 JavaScript .....	7
2.4 Bootstrap .....	8
2.5 Node.js .....	10
2.6 SQLite .....	13
2.7 Google Fonts .....	15
2.8 EJS .....	16
2.9 Chart.js .....	17
2.10 Express.js - Section .....	19
2.11 ความต้องการใช้งานของเว็บไซต์ .....	21
Use-case Descriptions .....	22
2.12 องค์ประกอบเชิงสถาปัตยกรรมของเว็บไซต์ .....	31
3 – Tier Architecture .....	32
การออกแบบฐานข้อมูล .....	34
พจนานุกรมข้อมูล (Data Dictionary) .....	35
ตาราง users (ผู้ใช้งาน) .....	35
ตาราง rooms (ห้องพัก) .....	36
ตาราง meters (มิเตอร์น้ำและไฟ) .....	37
ตาราง bookings (การจองห้อง) .....	38
ตาราง tenants (ผู้เช่า) .....	39

ตาราง bills (ใบแจ้งหนี้).....	40
ตาราง payments (การชำระเงิน).....	41
ตาราง maintenance (การแจ้งซ่อม).....	42
ตาราง parcels (พัสดุ) .....	43
<b>บทที่ 3 วิธีการดำเนินงาน.....</b>	<b>44</b>
หน้าเริ่มต้นของผู้เช่า(/home) .....	44
หน้าเข้าสู่ระบบผู้ใช้งาน(/login).....	47
หน้าตรวจสอบบันทึกมิเตอร์ไฟฟ้า&น้ำ(/meter).....	50
หน้าส่งเรื่องแจ้งซ่อม(/repair) .....	52
หน้าแจ้งเตือนพัสดุ(/parcel) .....	55
หน้าเริ่มต้นของแอดมิน(/admin) .....	58
หน้าคำขอแจ้งซ่อมของแอดมิน(/fixpage).....	64
หน้าแจ้งเตือนพัสดุของแอดมิน(/parcelpage).....	67
หน้าจัดการห้องพัก(/manageroom).....	70
หน้าจัดการผู้เช่า(/manageuser) .....	73
หน้าจองห้องพัก(/bookroom).....	75
หน้าจัดการมิเตอร์ไฟ/น้ำ(/managemeter).....	78
หน้าใบแจ้งหนี้/ใบเสร็จ .....	80
<b>บทที่ 4 ผลการดำเนินงาน .....</b>	<b>83</b>
หน้าเข้าสู่ระบบผู้ใช้งาน .....	83
หน้าหลักผู้เช่า.....	84
หน้าตรวจสอบบันทึก มิเตอร์ .....	85
หน้าส่งเรื่องแจ้งซ่อม.....	86
หน้าแจ้งเตือนพัสดุ .....	87
หน้าแรกของเจ้าของหอพัก.....	88
หน้าคำขอแจ้งซ่อม.....	89

หน้าแจ้งเตือนพัสดุ .....	90
หน้าจัดการห้องพัก.....	91
หน้าจัดการผู้เช่า.....	93
หน้าจองห้องพัก .....	96
หน้าบันทึกมิเตอร์น้ำ/ไฟ .....	97
หน้าใบแจ้งหนี้/ใบเสร็จ .....	98
<b>บทที่ 5 คู่มือการใช้งาน.....</b>	<b>102</b>
สำหรับเจ้าของห้องพัก .....	102
หน้าเข้าสู่ระบบ.....	102
เมื่อเข้าสู่ระบบสำเร็จ .....	102
เมนูการใช้งาน .....	103
หน้าแรก .....	104
หน้าคำขอแจ้งซ่อม .....	105
หน้าแจ้งเตือนพัสดุ .....	107
หน้าจัดการห้องพัก .....	110
หน้าจัดการผู้เช่า.....	114
หน้าการจองห้องพัก .....	117
หน้าจัดการมิเตอร์ไฟ/น้ำ .....	119
หน้าใบแจ้งหนี้และใบเสร็จ .....	120
สำหรับผู้เช่า .....	127
หน้าเข้าสู่ระบบ.....	127
เมื่อเข้าสู่ระบบสำเร็จ .....	127
เมนูการใช้งาน .....	128
หน้าแรก .....	128
ตรวจสอบบันทึกมิเตอร์ไฟ/น้ำ .....	129
ส่งเรื่องแจ้งซ่อม.....	129
หน้าแจ้งพัสดุ.....	131

## บทที่ 1 บทนำ

### ที่มาและความสำคัญ

ปัจจุบันธุรกิจห้องพักรายเดือนมีการเติบโตอย่างต่อเนื่อง เนื่องจากมีความต้องการที่พักระยะยาวในกลุ่มนักศึกษา พนักงานบริษัท และบุคคลทั่วไปที่ต้องการที่พักอาศัยในราคาประหยัด อย่างไรก็ตาม การบริหารจัดการห้องพักยังคงเป็นปัญหาสำคัญ ไม่ว่าจะเป็นการรับจองห้อง การติดตามสถานะการเช่า การชำระค่าเช่า และการแจ้งซ่อมบำรุง

ระบบจัดการห้องพักรายเดือนถูกพัฒนาขึ้นเพื่อช่วยให้เจ้าของที่พักสามารถบริหารจัดการห้องพักได้ง่ายขึ้น ลดข้อผิดพลาดที่เกิดจากการจัดการแบบเดิม เช่น การจดบันทึกด้วยกระดาษหรือไฟล์เอกสารที่อาจเกิดการสูญหาย รวมถึงช่วยให้ผู้เช่าสามารถตรวจสอบข้อมูลห้องพักและทำธุกรรมต่าง ๆ ได้สะดวกมากขึ้น

### วัตถุประสงค์

- เพื่อให้ผู้เช่าสามารถค้นหาห้องพัก วางแผนจอง และตรวจสอบสถานะการเช่าได้ง่ายผ่านระบบออนไลน์
- เพื่อช่วยให้เจ้าของหอพักสามารถบริหารจัดการข้อมูลผู้เช่า การชำระค่าเช่า และการแจ้งซ่อมบำรุงได้อย่างเป็นระบบ
- เพื่อเพิ่มความสะดวกในการแจ้งเตือนวันครบกำหนดชำระค่าเช่าและการต่อสัญญาเช่า
- เพื่อช่วยลดข้อผิดพลาดและเพิ่มประสิทธิภาพในการจัดการข้อมูลห้องพัก

### ประโยชน์ที่คาดว่าจะได้รับ

- ลดภาระของเจ้าของหอพักในการติดตามสถานะการเช่าและการชำระเงิน
- ผู้เช่าสามารถค้นหาห้องพักและดำเนินการจองได้สะดวกผ่านระบบออนไลน์
- ลดข้อผิดพลาดที่เกิดจากการบันทึกข้อมูลด้วยวิธีเดิม เช่น การจดบันทึกด้วยกระดาษ
- เพิ่มความสะดวกในการติดต่อระหว่างเจ้าของหอพักและผู้เช่า เช่น การแจ้งซ่อมบำรุงผ่านระบบ

## ขอบเขตของโครงการ

ระบบจัดการห้องพักรายเดือนนี้ถูกออกแบบมาเพื่อช่วยเจ้าของหอพักและผู้เช่าให้สามารถจัดการข้อมูลและธุกรรมที่เกี่ยวข้องได้อย่างมีประสิทธิภาพ โดยขอบเขตของระบบจะครอบคลุมฟังก์ชันหลักดังต่อไปนี้:

1. **บันทึกมิเตอร์น้ำและไฟฟ้า** – ระบบจะช่วยบันทึกค่ามิเตอร์น้ำและไฟฟ้าเพื่อใช้ในการคำนวณค่าใช้จ่าย
2. **ระบบแจ้งซ่อม** – ระบบสามารถให้ผู้เช่าแจ้งสิ่งที่ต้องการซ่อมได้ เพื่อให้เจ้าของหอพักรับทราบและสามารถติดตามการทำงานได้
3. **ระบบแจ้งพัสดุ** – ระบบจะแจ้งไปยังผู้เช่าว่ามีพัสดุถึงผู้เช่าจากการบันทึกของเจ้าของหอพัก
4. **สร้างใบแจ้งหนี้** – ระบบสามารถสร้างใบแจ้งหนี้ยื้อต่อไปตามรอบการชำระเงินโดยคำนวณจากการบันทึก มิเตอร์น้ำและไฟฟ้ารวมไปถึงราคาค่าเช่าห้องต่อเดือน
5. **ระบบจองห้อง** – เจ้าของหอพักสามารถระบุได้ว่าผู้เช่าได้เช่าห้องจนถึงเมื่อไหร่และใครคือผู้เช่าห้อง
6. **ระบบรายงานค่าใช้จ่าย** – เจ้าของหอพักสามารถดูรายงานค่าใช้จ่ายและยอดการชำระเงินจากผู้เช่า
7. **ระบบจัดการห้องพัก** – สามารถเพิ่ม ลบ หรือแก้ไขข้อมูลห้องพักและรายการเช่าห้องต่อเดือน
8. **ระบบจัดการผู้เช่า** – จัดเก็บข้อมูลของผู้เช่า ติดตามสถานะการเช่าและเพื่อเข้าใช้งานระบบ

## บทที่ 2 แนวคิดทฤษฎี และเอกสารที่เกี่ยวข้อง

### เทคโนโลยีที่ใช้ในการพัฒนาเว็บไซต์

#### 2.1 HTML

HTML, which stands for Hyper Text Markup Language, is the standard markup language used to create and structure web pages. It provides the basic framework for how content is displayed on the World Wide Web. HTML is not a programming language but a markup language, meaning it is used to define the structure and presentation of content on a web page, rather than to perform computations or functions. Here's an overview of the architecture of HTML:

##### Elements

HTML documents consist of a collection of elements. An element is a fundamental building block of an HTML page and can represent various types of content, such as text, images, links, forms, and more. Each element is defined by a specific tag, enclosed in angle brackets.

##### Tags

Tags are used to mark up content within an HTML document. They come in pairs: an opening tag and a closing tag, with the content enclosed between them. The opening tag denotes the beginning of an element, and the closing tag signifies its end.

##### Attributes

HTML tags can also have attributes that provide additional information about the element or modify its behavior. Attributes are added within the opening tag and are typically in the form of name-value pairs.

##### Document Structure

HTML documents have a specific structure. The basic structure includes an declaration that specifies the HTML version being used, followed by the , , and elements. The element wraps the entire content of the web page. The contains metadata, such as the page title and links to external resources, and the holds the visible content of the page.

## Nesting

HTML elements can be nested within one another, creating a hierarchical structure. Proper nesting is essential to maintain the document's structure and to ensure that browsers render the pages correctly.

## Document Object Model (DOM)

The Document Object Model is a programming interface provided by web browsers that represents the structure of an HTML document as a tree of objects. Each HTML element is represented as a node in this tree, allowing developers to manipulate and interact with the content and structure of a web page using JavaScript.

## Semantic

elements, like `<header>`, `<nav>` , `<article>`, and `<footer>`, provide meaning to the structure of a web page. They help search engines and assistive technologies understand the content and improve accessibility.

## CSS Integration

While HTML defines the structure of a web page, Cascading Style Sheets (CSS) are used to control the presentation and styling of the content. CSS is often linked to HTML documents to define how elements should be visually displayed.

## 2.2 CSS

Cascading Style Sheets (CSS) is a stylesheet language used for describing the presentation and styling of web documents, including HTML documents. CSS defines how elements in a web page are displayed, such as their layout, colors, fonts, and spacing. Here's an overview of the architecture of CSS:

### Selectors CSS

operates on the basis of selectors, which are patterns used to select and target HTML elements that you want to style. Selectors can be based on element types, classes, IDs, attributes, and more.

### Properties

CSS rules consist of selectors and declarations. Declarations are made up of properties and values. Properties define what aspect of the element you want to style, such as color, font size, or margin. Values specify the actual styling for that property. Declarations: Declarations are grouped within curly braces {} and are separated by semicolons.

### Stylesheets

CSS rules can be defined within an HTML document using the , or they can be placed in an external CSS file with a .css extension and linked to the HTML document using the element. External stylesheets are commonly used for separating content and presentation, making it easier to maintain and update the styling across multiple web pages.

### Inheritance

CSS properties can be inherited from parent elements to child elements in the HTML document. This means that if you set a style on a parent element, its child elements can inherit that style unless you explicitly override it.

### Specificity

Specificity is a mechanism that determines which CSS rule is applied when multiple rules target the same element. Specificity is based on the type of selector used, where more specific selectors take precedence.

## Cascading

The "C" in CSS stands for "cascading." This means that multiple CSS rules can be applied to the same element, and their styles cascade down to affect that element. The order of precedence is determined by specificity and the order of the rules in the stylesheet.

## Media Queries

CSS allows for the creation of responsive web designs by using media queries. Media queries enable you to apply different styles based on the characteristics of the user's device, such as screen size, orientation, or resolution.

## CSS Preprocessors

CSS preprocessors like SASS and LESS are tools that extend the capabilities of CSS by introducing variables, nesting, functions, and other features. These preprocessors are compiled into standard CSS for use in web pages.

## CSS Frameworks

CSS frameworks like Bootstrap and Foundation provide pre-designed CSS styles and components, making it easier for developers to create consistent and responsive web designs.

## Browser Rendering

The web browser's rendering engine is responsible for interpreting and rendering the CSS rules. Different browsers may have slight variations in how they interpret and display CSS, which is why web developers often need to write cross-browser-compatible CSS.

## 2.3 JavaScript

JavaScript (often abbreviated as JS) is a programming language that alongside HTML and CSS, is a core technology of the World Wide Web. It's used to make web pages interactive and dynamic. For example, if a website has animations, allows you to enter information in a form, or shows you a map that lets you zoom in and out, that's likely thanks to JavaScript. Here's a breakdown of what JavaScript does:

### Makes web pages interactive

JavaScript can respond to user actions like clicks, scrolls, and form submissions. This allows for features like dropdown menus, image carousels, and more.

### Updates content without reloading

JavaScript can change the content of a web page without having to reload the entire page. This makes things faster and smoother for the user.

### Calculates and validates data

JavaScript can perform calculations and validate user input on the client-side (the user's device) before it's sent to the server. This helps prevent errors and improves security.

## 2.4 Bootstrap

Bootstrap is a popular front-end framework for web development that simplifies the process of designing responsive and visually appealing web pages. It provides a comprehensive set of pre-designed HTML, CSS, and JavaScript components that can be used to create web interfaces quickly and efficiently. Here's an overview of the architecture and key components of Bootstrap:

### HTML Structure

Bootstrap starts with the basic HTML structure that every web page requires. It provides a grid system for laying out content in a responsive manner and includes basic templates for common web page elements like navigation bars, forms, and modals.

### CSS Styles

Bootstrap includes a robust CSS framework that provides a consistent and visually appealing design for your web pages. The CSS is organized into various files, allowing you to selectively include the parts you need for your project. Bootstrap uses a responsive 12 column grid system that adapts to different screen sizes and devices.

### SASS/SCSS

Bootstrap is built using the SASS (Syntactically Awesome Style Sheets) preprocessor, making it highly customizable. You can customize the framework by modifying variables, such as colors, fonts, and spacing, and then recompiling the SASS/SCSS source files to generate a customized CSS file.

### JavaScript Components

Bootstrap provides a collection of JavaScript components that enhance user interactivity and functionality. These include modal dialogs, carousels, dropdown menus, tooltips, popovers, and more. These components are designed to work seamlessly with Bootstrap's CSS and HTML structure.

### Responsive Design

Bootstrap is known for its responsive design capabilities. It uses a mobile-first approach, meaning it's designed for small screens first and then scales up for larger screens. This ensures

that your web pages look good and function well on a wide range of devices, from smartphones to desktops.

### **Customization and Theming**

Bootstrap can be customized to match your specific project requirements. You can modify the framework's variables and create custom themes to reflect your brand's design and style guidelines.

### **Browser Compatibility**

Bootstrap is designed to work consistently across different web browsers, reducing the need for browser-specific CSS hacks. It ensures that your web pages look and function similarly on major browsers.

### **Accessibility**

Bootstrap emphasizes accessibility and follows best practices to make web content accessible to users with disabilities. This includes providing ARIA roles and attributes for assistive technologies.

### **Extensions and Plugins**

The Bootstrap ecosystem includes numerous extensions, themes, and third-party plugins that can be integrated with your Bootstrap-based project to extend functionality, add animations, or implement additional features.

### **Documentation**

Bootstrap offers extensive documentation with examples, code samples, and guidelines for using its components and features effectively. This documentation is an essential resource for developers working with Bootstrap.

### **Community and Support**

Bootstrap has a large and active community of developers, which means you can find help, tutorials, and third-party resources online to assist with your Bootstrap projects.

## 2.5 Node.js

Node.js is an open-source, cross-platform runtime environment that allows developers to execute JavaScript code outside of a web browser. It is built on the **V8 JavaScript engine**, which powers Google Chrome, and is designed for building scalable and high-performance applications, particularly for web servers and networking applications. Below is an overview of Node.js architecture and its key components:

### Event-Driven and Non-Blocking Architecture

One of the defining features of Node.js is its **event-driven, non-blocking I/O model**, which makes it highly efficient for handling concurrent requests. Unlike traditional server-side technologies that create a new thread for each connection, Node.js operates on a **single-threaded event loop**, making it lightweight and capable of handling thousands of simultaneous connections.

### Asynchronous Programming

Node.js is designed to handle multiple requests asynchronously. Instead of waiting for a task to complete before moving to the next one, Node.js uses **callbacks, Promises, and async/await** to handle asynchronous operations efficiently. This design improves performance, especially for applications requiring frequent I/O operations such as database interactions and API requests.

### Modules and NPM (Node Package Manager)

Node.js follows a modular approach, allowing developers to use built-in and third-party modules to enhance functionality.

- Built-in Modules: Node.js comes with several core modules such as fs (file system), http (for handling web servers), path (for file paths), and events (for event handling).
- NPM (Node Package Manager: NPM is the default package manager for Node.js, providing access to thousands of open-source libraries. Developers can easily install, update, and manage dependencies using NPM commands.

## HTTP Server and API Development

Node.js allows developers to create lightweight web servers without relying on third-party software like Apache or Nginx. The http module provides built-in functionality to handle HTTP requests and responses, making it a popular choice for building RESTful APIs and real-time applications.

### Express.js Framework

While Node.js provides a low-level API, frameworks like Express.js simplify the development of web applications. Express.js offers features such as routing, middleware, template engines, and error handling, making it easier to build scalable and maintainable applications.

### File System and Database Interaction

Node.js provides an extensive fs module for handling file operations, such as reading, writing, and updating files asynchronously. It also integrates seamlessly with various databases:

- SQL Databases: MySQL, PostgreSQL, SQLite (via libraries like Sequelize and Knex.js)
- NoSQL Databases: MongoDB (via Mongoose), Firebase, CouchDB
- In-Memory Databases: Redis (for caching and real-time data processing)

### WebSockets and Real-Time Communication

Node.js supports WebSockets using libraries like Socket.io, enabling real-time bidirectional communication between clients and servers. This is particularly useful for chat applications, live notifications, online gaming, and collaborative tools.

### Security Considerations

Although Node.js is powerful, security best practices must be followed to protect applications from vulnerabilities. Key security measures include:

- Validating user input to prevent SQL injection and XSS attacks
- Using HTTPS for secure communication
- Implementing authentication and authorization mechanisms (e.g., JWT, OAuth)
- Keeping dependencies updated to prevent security vulnerabilities

## Scalability and Performance Optimization

Node.js applications can be scaled efficiently using:

- Load Balancing: Distributing traffic across multiple instances of an application.
- Cluster Module: Utilizing multiple CPU cores to run multiple instances of the application.
- Microservices Architecture: Breaking down applications into smaller services to enhance maintainability and scalability.

## Cross-Platform Compatibility

Node.js is cross-platform, meaning it can run on Windows, macOS, and Linux. This makes it an ideal choice for developers working on cloud-based applications, microservices, and cross-platform development.

## Documentation and Community Support

Node.js has extensive official documentation, making it easy to learn and adopt. It also has a large and active community that contributes to its ecosystem, providing support, tutorials, and third-party libraries.

## 2.6 SQLite

SQLite is a lightweight, self-contained, and serverless relational database management system (RDBMS) that is widely used in various applications. Unlike traditional database systems that require a dedicated server, SQLite operates using a single database file, making it an ideal choice for embedded systems, mobile applications, and desktop software. It is known for its simplicity, efficiency, and reliability. Architecture and Key Features of SQLite :

### Database Structure

DB Browser for SQLite offers a clean and intuitive GUI that enables users to interact with SQLite databases effortlessly. Unlike command-line tools, it allows users to manage database structures, execute queries, and browse data visually without writing extensive SQL commands

### Single File Storage

One of SQLite's defining characteristics is that the entire database, including tables, indexes, triggers, and views, is stored in a single file. This makes it highly portable and easy to manage without requiring complex configurations.

### SQL Support

SQLite supports a large subset of SQL92, including:

- CRUD Operations (Create, Read, Update, Delete)
- JOIN Queries for combining multiple tables
- Transactions and ACID Compliance for data integrity
- Indexes and Views to optimize data access

### Serverless Architecture

Unlike traditional relational databases such as MySQL or PostgreSQL, SQLite does not require a separate server process. Applications can directly read from and write to the database file using SQLite's built-in library. This makes SQLite highly efficient for local data storage.

### ACID Compliance

SQLite follows the ACID (Atomicity, Consistency, Isolation, Durability) principles, ensuring that database transactions are reliable and secure, even in cases of system crashes or power failures.

## Cross-Platform Compatibility

SQLite can run on multiple platforms, including Windows, macOS, Linux, Android, and iOS. The database file created on one system can be easily transferred and used on another without modification.

## 2.7 Google Fonts

Google Fonts is a free service that provides a library of fonts you can use for your websites and other projects. Here's a breakdown of its key features:

### Free and Open Source Fonts

Google Fonts offers a collection of fonts that are free to use and come from open source sources. This means you can use them for any personal or commercial project without worrying about licensing fees.

### Easy Integration

Google Fonts provides a user-friendly web interface for browsing fonts and integrating them into your projects. You can use CSS or an API to embed the fonts directly on your website.

### Improved Website Performance

Since Google Fonts are hosted by Google, they load quickly and efficiently, which can help improve the performance of your website.

### Wide Variety of Fonts

Google Fonts offers a vast collection of fonts in various styles, from classic serifs to modern sans-serifs. You can easily find the perfect font to match the look and feel of your project.

### Accessibility

Many Google Fonts offer support for multiple languages and characters, making them ideal for creating websites that cater to a global audience.

## 2.8 EJS

EJS (Embedded JavaScript) is a simple and powerful templating engine for Node.js that allows developers to generate dynamic HTML content by embedding JavaScript code within templates. It is widely used in web applications to render data dynamically while maintaining a clean and structured codebase. Architecture and Key Features of EJS :

### HTML Templating with JavaScript

EJS enables developers to embed JavaScript logic directly into HTML using special syntax (<% %> for logic and <%= %> for output). This allows for dynamic content rendering, such as inserting data from a database or processing user input.

### Template Files (.ejs)

EJS templates are stored as .ejs files, which contain a mix of HTML and embedded JavaScript. These files are processed by the EJS engine, which replaces JavaScript expressions with actual values before sending the final HTML to the client.

### Data Binding

EJS makes it easy to pass data from a server to a template. Developers can define variables and use them within the template to dynamically generate content. This makes EJS particularly useful for rendering lists, tables, or personalized user content.

### Control Structures (Conditionals & Loops)

EJS supports common control structures such as if statements and loops, making it possible to conditionally render elements or iterate over data collections

### Partials and Layouts

EJS supports partial templates, which allow developers to break down web pages into reusable components such as headers, footers, and navigation bars. This promotes modular development and cleaner code organization.

### Integration with Express.js

EJS is commonly used with Express.js, a popular Node.js web framework. Express provides built-in support for EJS, making it easy to set up and render templates

### Client-Side and Server-Side Rendering

While EJS is mainly used for server-side rendering (SSR), it can also be used to generate HTML dynamically on the client-side with JavaScript frameworks or AJAX requests.

## 2.9 Chart.js

Chart.js is a popular open-source JavaScript library used for creating interactive and visually appealing charts on web applications. It simplifies the process of adding data visualizations, such as bar charts, line graphs, and pie charts, using HTML5 <canvas> elements. Chart.js is lightweight, highly customizable, and supports animation, making it an excellent choice for data-driven applications. Architecture and Key Features of Chart.js :

### Canvas-Based Rendering

Chart.js uses the HTML5 <canvas> element to render charts, ensuring smooth performance and high-quality graphics across different devices and screen sizes. Unlike SVG-based charting libraries, canvas rendering allows for faster updates and efficient handling of large datasets.

### Chart Types

Chart.js provides a variety of built-in chart types, including:

- Line Charts
- Bar Charts
- Pie and Doughnut Charts
- Radar Charts
- Polar Area Charts
- Scatter and Bubble Charts

Each chart type can be customized with different styles, colors, labels, and tooltips.

### Data Configuration

Chart.js requires data to be structured in a specific format within JavaScript objects. A basic line chart

### Responsive Design and Animation

Chart.js charts are fully responsive, meaning they automatically adjust to different screen sizes. Additionally, animations can be applied when rendering charts, enhancing user experience.

### Customization and Styling

Chart.js offers extensive customization options, including:

- Changing colors, fonts, and border styles

- Modifying axis labels and grid lines
- Configuring tooltips and legend styles
- Using gradient fills and custom shapes

## Interactivity and User Interaction

Chart.js supports interactive features such as:

- Hover effects and tooltip pop-ups
- Click events for handling user interactions
- Dynamic updates and real-time data changes

## Plugins and Extensions

The Chart.js ecosystem includes various plugins to extend functionality, such as adding annotations, exporting charts as images, or integrating with third-party tools.

## Integration with Frameworks

Chart.js can be easily integrated into web applications using JavaScript frameworks like React, Angular, and Vue.js. It is also commonly used in Node.js-based applications for server-side data visualization.

## 2.10 Express.js - Section

Express.js is a lightweight and flexible web application framework for Node.js that simplifies building web applications and APIs. It provides a set of features for handling HTTP requests, managing routes, and integrating middleware to enhance the functionality of a web server. Below is an overview of the architecture and key components of Express.js:

### Routing

Express.js provides a powerful routing mechanism that allows developers to define routes for handling HTTP requests efficiently. Routes determine how an application responds to a client request based on the HTTP method (GET, POST, PUT, DELETE) and the request URL.

### Middleware

Middleware functions in Express.js process incoming requests before sending a response. Middleware can be used for logging, authentication, error handling, and modifying request and response objects. Express provides built-in middleware, third-party middleware, and custom middleware options.

### Template Engines

Express supports various template engines like **EJS, Pug, and Handlebars** for rendering dynamic web pages. Template engines allow developers to embed JavaScript logic into HTML templates to generate content dynamically.

### Static Files

Express allows serving static files like images, stylesheets, and JavaScript files using the built-in `express.static` middleware.

### Request Handling

Express provides simple ways to handle query parameters, URL parameters, and request bodies.

### Error Handling

Express provides a flexible way to handle errors through middleware functions.

### REST API Development

Express.js is widely used for building RESTful APIs due to its simplicity and efficiency.

## **Security and Performance**

Express can be enhanced with security middleware such as helmet.js to prevent security vulnerabilities and compression to improve performance.

## **Scalability and Modularity**

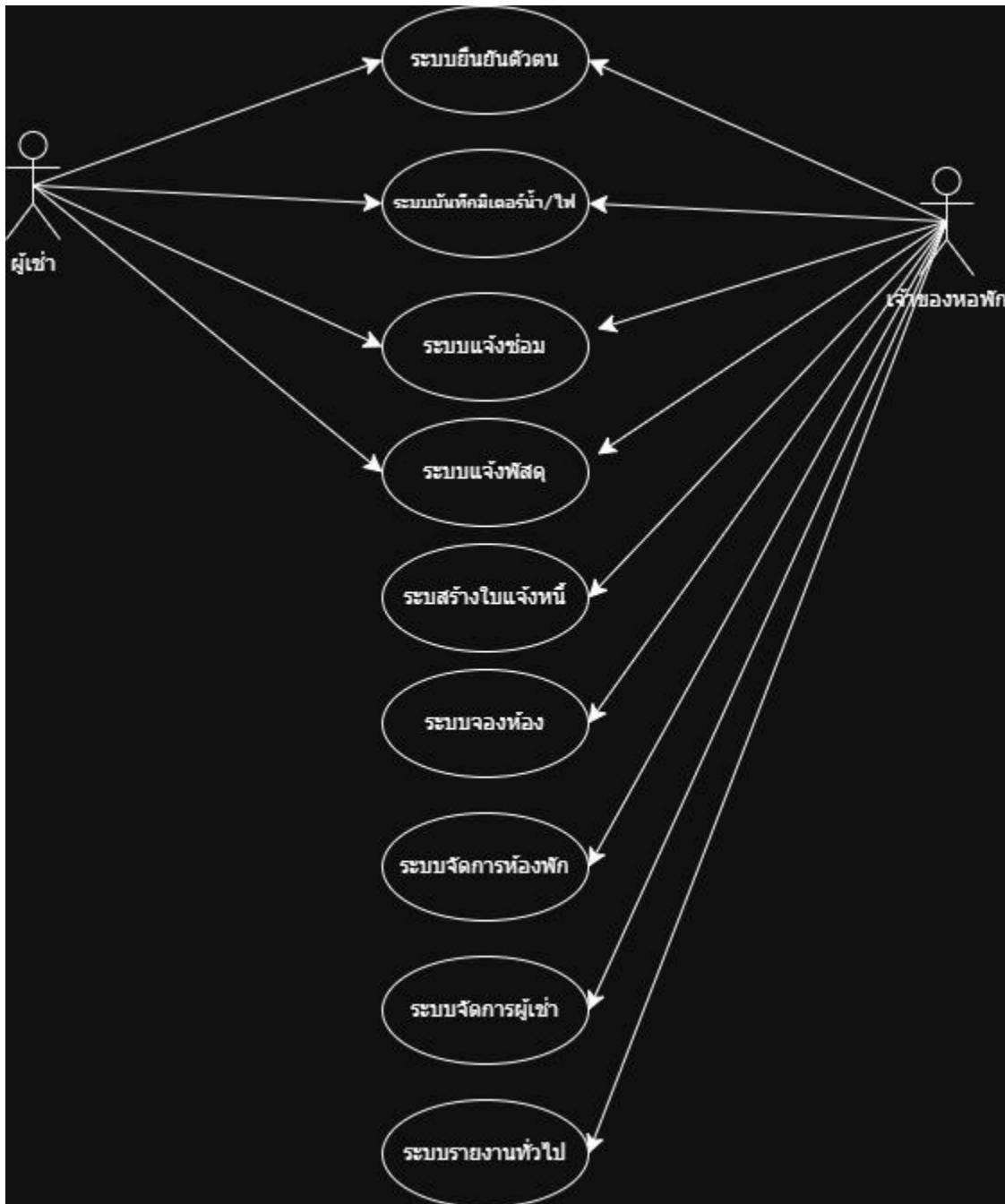
Express.js allows applications to be modular by organizing routes into separate files using the Router module.

## **Documentation and Community**

Express.js has comprehensive documentation and a large community, making it easy for developers to find resources, tutorials, and third-party plugins to extend functionality.

## 2.11 ความต้องการใช้งานของเว็บไซต์

### UML Use-case Diagram



## Use-case Descriptions

Use Case Name: ระบบยืนยันตัวตนผู้ใช้	ID:1
Primary Actor: ผู้ใช้, เจ้าของหอพัก	
Brief Description: ระบบยืนยันตัวตนของผู้ใช้ก่อนเข้าใช้งานระบบ	
Trigger: ผู้ใช้ต้องเข้าสู่ระบบ	
Preconditions: ผู้ใช้ต้องมีบัญชีในระบบ	
Post conditions: ระบบยืนยันตัวตนสำเร็จ	
<p><b>Normal Flow of Events:</b></p> <ol style="list-style-type: none"> <li>1. ผู้ใช้กรอกชื่อผู้ใช้และรหัสผ่าน</li> <li>2. ระบบตรวจสอบข้อมูลจากฐานข้อมูล</li> <li>3. หากข้อมูลถูกต้อง ระบบอนุมัติให้เข้าระบบ</li> <li>4. ระบบแสดงหน้าหลักของผู้ใช้ตามบทบาท</li> </ol>	
<p><b>Alternate/Exceptional Flows:</b></p> <ul style="list-style-type: none"> <li>- หากรหัสผ่านไม่ถูกต้อง: ระบบแจ้งเตือนและให้ผู้ใช้ลองใหม่</li> </ul>	

Use Case Name: ระบบบันทึกมิเตอร์น้ำและไฟฟ้า	ID:2
Primary Actor: ผู้เช่า, เจ้าของหอพัก	
Brief Description: ให้ผู้สามารถดูข้อมูลค่าน้ำและค่าไฟได้ หรือ เจ้าของหอพักสามารถบันทึกมิเตอร์น้ำและไฟฟ้าได้	
Trigger: ผู้เช่าทำการกดปุ่ม ตรวจสอบค่าน้ำ/ไฟ หรือ เจ้าของหอพักกรอกข้อมูลมิเตอร์น้ำและไฟฟ้าจากนั้นกดปุ่มบันทึก	
Preconditions: ผู้เช่าจะสามารถตรวจสอบได้เมื่อมีข้อมูลการใช้น้ำ/ไฟในระบบ	
Post conditions: แสดงข้อมูลค่าน้ำ/ไฟฟ้าให้ผู้เช่า	
Normal Flow of Events:	
<p>ผู้เช่า</p> <ol style="list-style-type: none"> <li>1. ผู้เช่ากดปุ่ม ตรวจสอบค่าน้ำ/ไฟ</li> <li>2. ระบบตรวจสอบข้อมูลจากฐานข้อมูล</li> <li>3. ระบบแสดงข้อมูลค่าน้ำและค่าไฟของผู้เช่า</li> </ol> <p>เจ้าของหอพัก</p> <ol style="list-style-type: none"> <li>1. กรอกข้อมูลมิเตอร์น้ำและไฟฟ้า</li> <li>2. กดบันทึกข้อมูล</li> </ol>	
Alternate/Exceptional Flows:	<ul style="list-style-type: none"> <li>- 'ไม่มีข้อมูลค่าน้ำ/ไฟฟ้า': ระบบแจ้งเตือนว่าไม่มีข้อมูล</li> </ul>

Use Case Name: ระบบแจ้งช่อง	ID:3
Primary Actor: ผู้เข้า, เจ้าของหอพัก	
Brief Description: ผู้เข้าสามารถแจ้ง ติดตามสถานะ การซ้อมแซมจากเจ้าของหอพักได้	
Trigger: ผู้ใช้ทำการกดปุ่ม แจ้งช่องแซม	
Preconditions: -	
Post conditions: ระบบบันทึกประวัติการซ้อม	
<b>Normal Flow of Events:</b>	
<p>ผู้เข้า</p> <ol style="list-style-type: none"> <li>1. ผู้ใช้กดปุ่ม แจ้งช่อง</li> <li>2. กรอกข้อมูลปัญหาที่พบ</li> <li>3. ระบบยืนยันและบันทึกข้อมูล</li> <li>4. ระบบแสดงสถานะการซ้อม (แจ้งเรื่อง, กำลังดำเนินการ, เสร็จสิ้น)</li> </ol>	
<p>เจ้าของหอพัก</p> <ol style="list-style-type: none"> <li>1. กดปุ่มแจ้งช่อง</li> <li>2. ดูรายละเอียดการแจ้งช่อง</li> <li>3. เปลี่ยนสถานะการแจ้งช่อง</li> <li>4. บันทึกการแจ้งช่อง</li> </ol>	
<b>Alternate/Exceptional Flows:</b>	

Use Case Name: ระบบแจ้งพัสดุ	ID:4
Primary Actor: เจ้าของหอพัก, ผู้เช่า	
Brief Description: ระบบแจ้งเตือนพัสดุเข้าจากการบันทึก	
Trigger: เจ้าของหอพักกดปุ่มบันทึกพัสดุเข้า	
Preconditions: ต้องมีข้อมูลผู้เช่าและห้องพัก	
Post conditions: ระบบแจ้งเตือนไปยังผู้เช่าห้องที่มีพัสดุมาถึง	
<p><b>Normal Flow of Events:</b></p> <p>เจ้าของหอพัก</p> <ol style="list-style-type: none"> <li>เจ้าของหอกรอกข้อมูลเลขห้องที่มีพัสดุมาถึง</li> <li>กรอกจำนวนพัสดุ</li> <li>กดปุ่มบันทึกข้อมูล</li> </ol> <p>ผู้เช่า</p> <ol style="list-style-type: none"> <li>กดเข้าสู่ระบบแจ้งพัสดุ</li> <li>ตรวจสอบรายการแจ้งเตือนพัสดุเข้า</li> </ol>	
<b>Alternate/Exceptional Flows:</b>	

Use Case Name: ระบบสร้างใบแจ้งหนี้	ID:5
Primary Actor: เจ้าของหอพัก	
Brief Description: ระบบช่วยสร้างใบแจ้งหนี้ค่าเช่า ค่าน้ำ ค่าไฟ	
Trigger: เจ้าของหอพักเลือก สร้างใบแจ้งหนี้	
Preconditions: มีข้อมูลค่าน้ำ ค่าไฟ และค่าเช่าในระบบ	
Post conditions: ระบบสร้างใบแจ้งหนี้สำเร็จ	
<b>Normal Flow of Events:</b>	
<ol style="list-style-type: none"> <li>1. เจ้าของหอพักเลือก สร้างใบแจ้งหนี้</li> <li>2. ระบบดึงข้อมูลค่าเช่าและค่าน้ำ/ไฟฟ้า</li> <li>3. ระบบสร้างใบแจ้งหนี้</li> <li>4. ระบบส่งใบแจ้งหนี้ให้ผู้เช่า</li> </ol>	
<b>Alternate/Exceptional Flows:</b>	
<ul style="list-style-type: none"> <li>- 'ไม่มีข้อมูลค่าใช้จ่าย': ระบบแจ้งเตือนว่าไม่มีข้อมูล</li> <li>- 'ระบบแจ้งเตือนกรณีการใช้งานผิดปกติ (ค่าน้ำหรือค่าไฟเพิ่มขึ้น 1.5 เท่าจากเดือนที่แล้ว)'</li> </ul>	

Use Case Name: ระบบจองห้อง	ID:6
<b>Primary Actor:</b> เจ้าของหอพัก	
<b>Brief Description:</b> ระบบช่วยระบุผู้เช่าห้อง ระยะเวลาการเช่า และสถานะของห้อง (ว่าง,ไม่ว่าง)	
<b>Trigger:</b> เจ้าของหอพักเลือกห้องและจองให้กับผู้เช่าคนใด	
<b>Preconditions:</b>	
<ul style="list-style-type: none"> <li>- มีข้อมูลผู้เช่า</li> <li>- ห้องต้องว่างในช่วงเวลาที่จะจอง</li> </ul>	
<b>Post conditions:</b> ระบบจองห้องตามระยะเวลา	
<b>Normal Flow of Events:</b>	
<ol style="list-style-type: none"> <li>1. เจ้าของหอพักเลือก ห้องเช่า</li> <li>2. เลือกระยะเวลาการเช่าห้อง</li> <li>3. เจ้าของหอพักเลือกผู้เช่าใส่ในห้องพัก</li> <li>4. กดปุ่มบันทึกการจอง</li> <li>5. ระบบบันทึกข้อมูลการจองห้องพัก</li> </ol>	
<b>Alternate/Exceptional Flows:</b>	

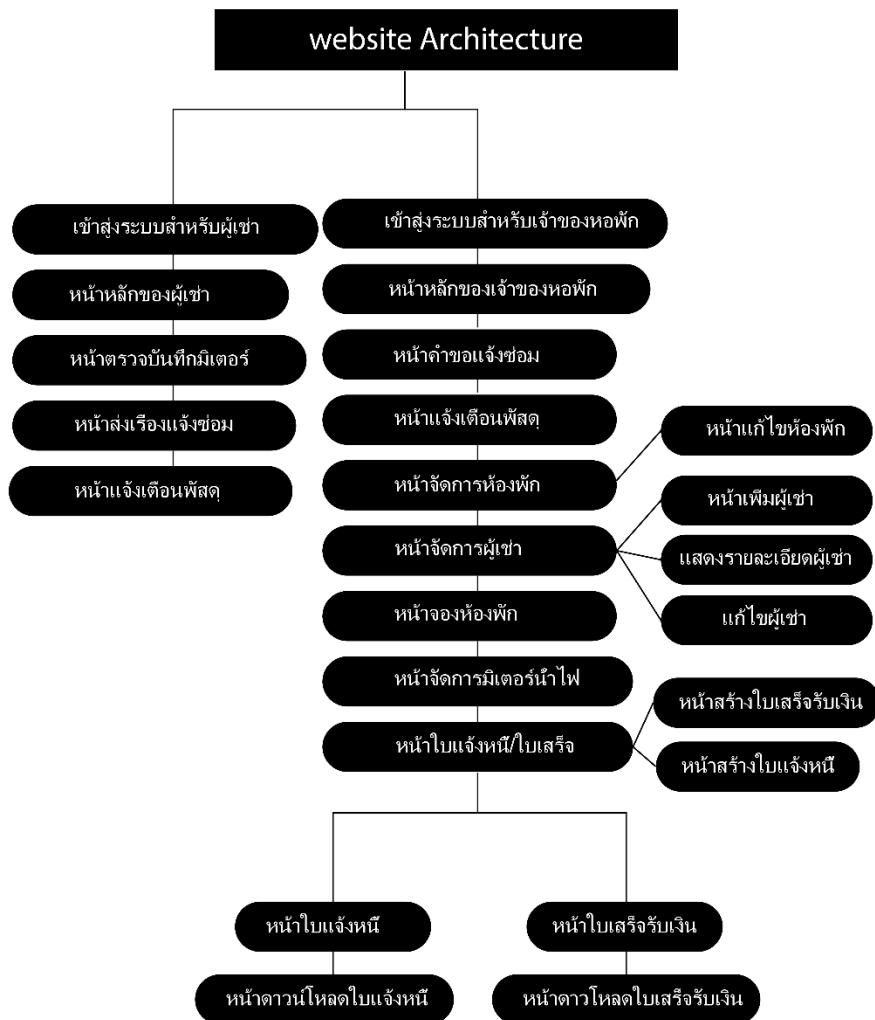
Use Case Name: ระบบช่วยจัดการข้อมูลห้องพัก	ID:7
Primary Actor: เจ้าของหอพัก	
Brief Description: ระบบช่วยจัดการข้อมูลห้องพัก	
Trigger: เจ้าของหอพักเลือก จัดการห้องพัก	
Preconditions: ต้องมีข้อมูลห้องพักในระบบ	
Post conditions: ระบบบันทึกข้อมูลสำเร็จ	
Normal Flow of Events:	
<ol style="list-style-type: none"> <li>1. เจ้าของหอพักเลือก จัดการห้องพัก</li> <li>2. เจ้าของหอพักสามารถ เพิ่ม/แก้ไข/ลบข้อมูล ค่าเช่าต่อเดือนของห้องพัก</li> <li>3. ระบบบันทึกข้อมูลลงฐานข้อมูล</li> </ol>	
Alternate/Exceptional Flows:	

Use Case Name: ระบบช่วยจัดการผู้เช่า	ID:8
Primary Actor: เจ้าของหอพัก	
Brief Description: ระบบช่วยจัดการข้อมูลผู้เช่า	
Trigger: เจ้าของหอพักเลือก จัดการผู้เช่า	
Preconditions: ต้องมีข้อมูลผู้เช่าในระบบ	
Post conditions: ระบบบันทึกข้อมูลสำเร็จ	
Normal Flow of Events:	
<ol style="list-style-type: none"> <li>1. เจ้าของหอพักเลือก จัดการผู้เช่า</li> <li>2. เจ้าของหอพักสามารถ เพิ่ม/แก้ไข/ลบข้อมูลผู้เช่า</li> <li>3. ระบบบันทึกข้อมูลลงฐานข้อมูล</li> </ol>	
Alternate/Exceptional Flows:	

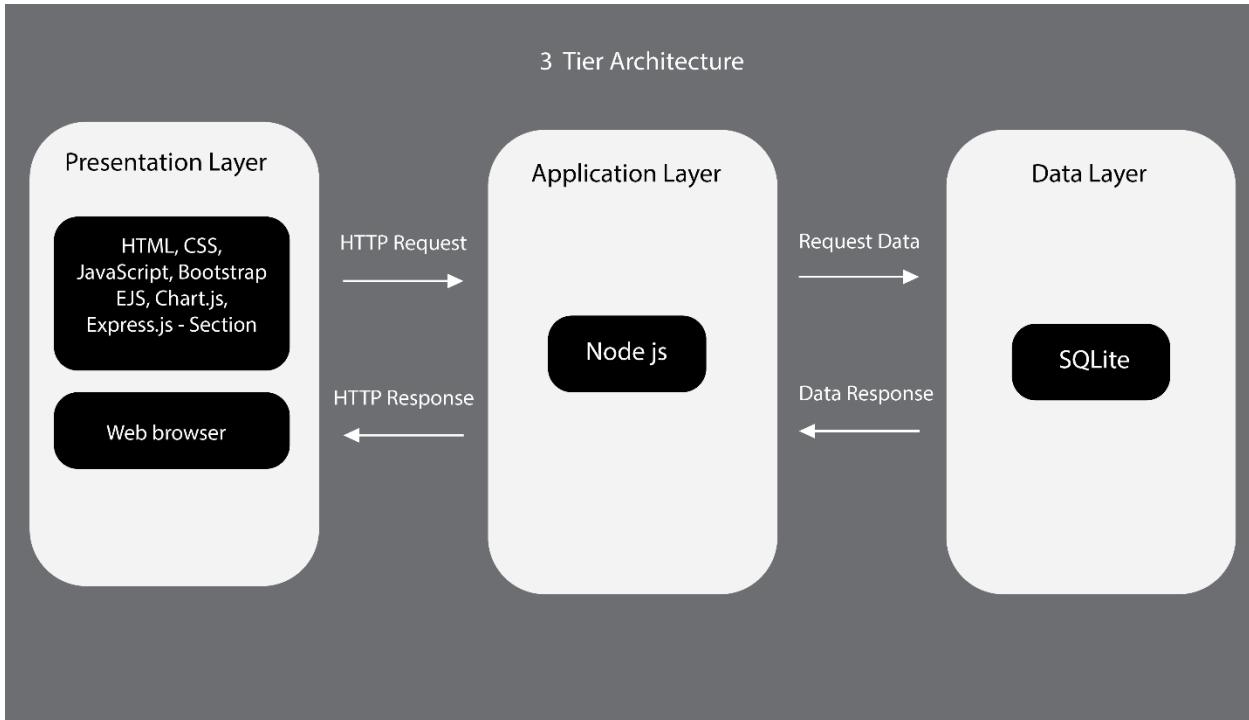
Use Case Name: ระบบรายงานทั่วไป	ID:9
Primary Actor: เจ้าของหอพัก	
Brief Description: ระบบสร้างรายงานสรุปค่าใช้จ่าย ค่าน้ำ ค่าไฟ และสถานะของผู้เช่า	
Trigger: เจ้าของต้องการดูรายงานประจำเดือน/ปี	
Preconditions: มีข้อมูลการใช้จ่าย ของผู้เช่าอยู่บันทึกครบถ้วน	
Post conditions: ระบบแสดงผลรายงาน	
<b>Normal Flow of Events:</b>	
<ol style="list-style-type: none"> <li>1. เจ้าของหอพักเลือก ประเภทรายการที่ต้องการ(รายเดือน/รายปี)</li> <li>2. ระบบดึงข้อมูลจากฐานข้อมูล</li> <li>3. ระบบคำนวณและจัดทำรายงานสรุป</li> <li>4. ระบบแสดงผลรายงาน</li> </ol>	
<b>Alternate/Exceptional Flows:</b>	
<ul style="list-style-type: none"> <li>- หากข้อมูลไม่เพียงพอ ระบบจะแจ้งเตือนเจ้าของหอพัก</li> </ul>	

## 2.12 องค์ประกอบเชิงสถาปัตยกรรมของเว็บไซต์

### Website Architecture



### 3 – Tier Architecture



#### การทำงานของระบบตาม 3-Tier Architecture

เว็บไซต์ถูกแบ่งออกเป็นสองฝั่งหลัก ได้แก่ ฝั่งผู้ใช้งาน (Client Side) และ ฝั่งเซิร์ฟเวอร์ (Server Side) โดยแต่ละส่วนทำงานดังนี้:

##### 1. Presentation Layer (ฝั่งผู้ใช้งาน)

เป็นส่วนที่ผู้ใช้มองเห็นและโต้ตอบผ่านเว็บเบราว์เซอร์ ใช้เทคโนโลยีหลัก ได้แก่

- HTML, CSS, JavaScript สำหรับโครงสร้างและดีไซน์
- Bootstrap เพื่อความสวยงามและความสามารถในการแสดงผลที่ตอบสนองต่ออุปกรณ์
- EJS ใช้เป็น Template Engine สำหรับแสดงผลข้อมูล
- Chart.js สำหรับสร้างกราฟและแผนภูมิ
- Express.js - Section เพื่อจัดการส่วน Login

เมื่อผู้ใช้ทำการร้องขอข้อมูลผ่านหน้าเว็บ Presentation Layer จะทำการส่ง HTTP Request ไปยังเซิร์ฟเวอร์ ที่ Application Layer

## 2. Application Layer (ผู้เชิร์ฟเวอร์ - Node.js)

เป็นส่วนกลางที่ประมวลผลคำร้องขอจากผู้ใช้ โดยใช้ Node.js และ Express.js เพื่อจัดการกับ Routing และ Logic ของแอปพลิเคชัน

- เมื่อได้รับ HTTP Request จาก Presentation Layer, Node.js จะประมวลผลและตัดสินใจว่า ต้องการข้อมูลอะไรจากฐานข้อมูล
- จากนั้น จะทำการส่ง Request Data ไปยัง Data Layer เพื่อดึงข้อมูลที่ต้องการ

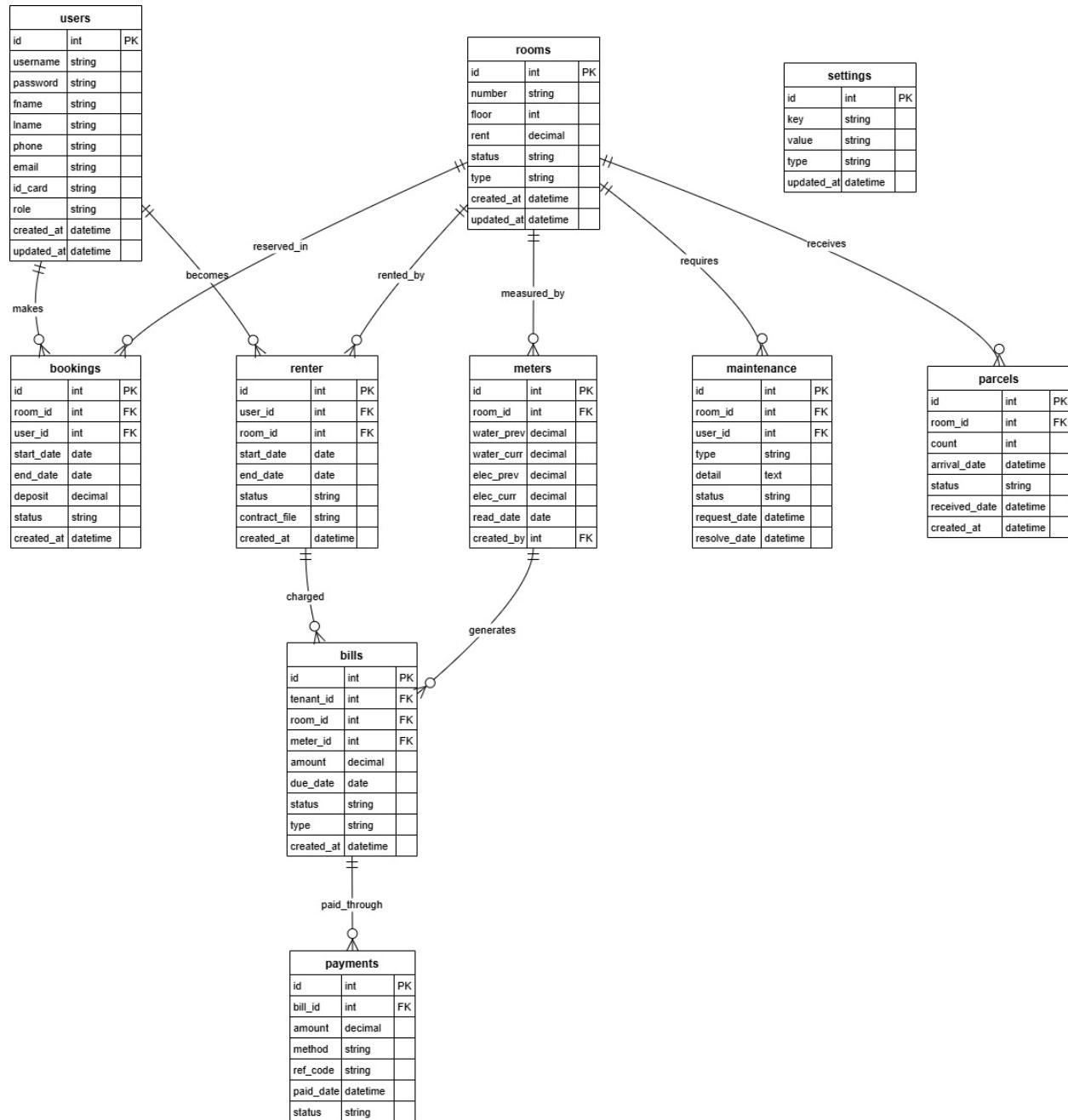
## 3. Data Layer (ฐานข้อมูล - SQLite)

เป็นส่วนที่เก็บข้อมูลของระบบ โดยใช้ SQLite เป็นฐานข้อมูลหลัก

- เมื่อได้รับคำขอ Request Data จาก Application Layer, SQLite จะทำการดึงข้อมูลที่เกี่ยวข้อง
- หลังจากประมวลผลแล้ว ระบบจะส่งข้อมูลกลับไปยัง Application Layer ผ่าน Data Response  
**ขั้นตอนสุดท้าย - ส่งข้อมูลกลับไปยังผู้ใช้**
  1. Application Layer ได้รับข้อมูลจาก Data Layer และนำไปประมวลผล
  2. จากนั้นจะส่งข้อมูลไปยัง Presentation Layer ผ่าน HTTP Response
  3. ผู้ใช้จะได้รับผลลัพธ์ที่แสดงผลผ่านเว็บเบราว์เซอร์

## การออกแบบฐานข้อมูล

### ER Diagram



## พจนานุกรมข้อมูล (Data Dictionary)

### ตาราง users (ผู้ใช้งาน)

Attribute Name	Description	Type	Key	Referenced
id(PRIMARY KEY)	รหัสผู้ใช้งาน	int	PK,NN	
username	ชื่อผู้ใช้	string	NN	
password	รหัสผ่าน	string	NN	
fname	ชื่อจริง	string	NN	
lname	นามสกุล	string	NN	
phone	เบอร์โทรศัพท์	string	NN	
email	อีเมล	string	N	
id_card	หมายเลขบัตรประชาชน	string	NN	
profile_image	รูปโปรไฟล์	string	N	
created_at	วันที่สร้างบัญชี	datetime	NN	
updated_at	วันที่แก้ไขบัญชี	datetime	N	

### ตาราง rooms (ห้องพัก)

Attribute Name	Description	Type	Key	Referenced
id(PRIMARY KEY)	รหัสห้องพัก	int	PK,NN	
number	หมายเลขห้อง	string	NN	
rent	ค่าเช่า	decimal	NN	
status	สถานะห้อง	string	NN	
created_at	วันที่สร้างห้อง	datetime	NN	
updated_at	วันที่แก้ไข	datetime	N	

### ตาราง meters (มิเตอร์น้ำและไฟ)

Attribute Name	Description	Type	Key	Referenced
id(PRIMARY KEY)	รหัสมิเตอร์	int	PK, NN	
room_id	รหัสห้องพัก	int	FK, NN	rooms(id)
water_prev	หน่วยน้ำก่อนหน้า	decimal	NN	
water_curr	หน่วยน้ำปัจจุบัน	decimal	NN	
elec_prev	หน่วยไฟก่อนหน้า	decimal	NN	
elec_curr	หน่วยไฟปัจจุบัน	int	NN	
read_date	วันที่อ่านมิเตอร์	datetime	NN	

### ตาราง bookings (การจองห้อง)

Attribute Name	Description	Type	Key	Referenced
id(PRIMARY KEY)	รหัสการจอง	int	PK	
room_id	รหัสห้องพัก	int	FK,NN	rooms(id)
user_id	รหัสผู้จอง	Int	FK,NN	users(id)
start_date	วันที่เริ่มจอง	date	NN	
end_date	วันที่สิ้นสุด	date	NN	
created_at	วันที่จอง	datetime	NN	

### ตาราง tenants (ผู้เช่า)

Attribute Name	Description	Type	Key	Referenced
id(PRIMARY KEY)	รหัสผู้เช่า	int	PK, NN	
user_id	รหัสผู้เช้งาน	int	FK, NN	users(id)
room_id	รหัสห้องพัก	int	FK, NN	rooms(id)
start_date	วันที่เริ่มเช่า	date	NN	
end_date	วันที่สิ้นสุดเช่า	date	NN	
status	สถานะการเช่า	string	NN	
created_at	วันที่บันทึก	datetime	NN	

### ตาราง bills (ใบเจ้งหนี้)

Attribute Name	Description	Type	Key	Referenced
id(PRIMARY KEY)	รหัสใบเจ้งหนี้	int	PK,NN	
tenant_id	รหัสผู้เช่า	int	FK, NN	tenants(id)
room_id	รหัสห้องพัก	Int	FK, NN	rooms(id)
meter_id	รหัสมิเตอร์	Int	FK, NN	meters(id)
water_amount	ค่าน้ำ	decimal	NN	
elec_amount	ค่าไฟ	decimal	NN	
total_amount	ยอดรวม	decimal	NN	
due_date	กำหนดชำระ	date	NN	
status	สถานะ	string	NN	
created_at	วันที่สร้าง	datetime	NN	

### ตาราง payments (การชำระเงิน)

Attribute Name	Description	Type	Key	Referenced
id(PRIMARY KEY)	รหัสการชำระเงิน	int	PK, NN	
bill_id	รหัสใบแจ้งหนี้	int	FK, NN	bill(id)
amount	จำนวนเงินที่ต้องชำระ	decimal	NN	
status	สถานะการชำระ	string	NN	
paid_at	วันที่ชำระ	datetime	NN	

### ตาราง maintenance (การแจ้งซ่อม)

Attribute Name	Description	Type	Key	Referenced
id(PRIMARY KEY)	รหัสแจ้งซ่อม	int	PK,NN	
room_id	รหัสห้องพัก	int	FK, NN	rooms(id)
user_id	รหัสผู้แจ้ง	Int	FK, NN	users(id)
type	ประเภทงานซ่อม	string	NN	
detail	รายละเอียด	text	NN	
status	สถานะงานซ่อม	string	NN	
request_date	วันที่แจ้งซ่อม	datetime	NN	
resolve_date	วันที่แก้ไข	datetime	N	

### ตาราง parcels (พัสดุ)

Attribute Name	Description	Type	Key	Referenced
id(PRIMARY KEY)	รหัสพัสดุ	int	PK, NN	
room_id	รหัสห้องพัก	int	FK, NN	rooms(id)
size	ขนาดของพัสดุ	string	NN	
arrival_date	วันที่พัสดุมาถึง	datetime	NN	
status	สถานะของพัสดุ	string	NN	
received_date	วันที่รับพัสดุ	datetime	NN	

## บทที่ 3 วิธีการดำเนินงาน

### หน้าเริ่มต้นของผู้เช่า(/home)

```

① README.md ② home.ejs ×
views > home.ejs > div > div.bg-white.p-3.mt-3.w-75.rounded-5.shadowing.row > div.col-9 > div.w-100 > div.table-responsive > table.w-100 > tr
1   <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
2   <div style="display: flex; justify-content: center; align-items: center;">
3     <div class="bg-white.p-3.mt-3.w-75 rounded-5 shadowing row">
4       <div class="col-3">
5         
6       </div>
7       <div class="col-9" style="display: flex; align-items: center;">
8         <div class="w-100">
9           <h2 style="font-size: 2vw;">ยินดีต้อนรับ, <%= data[0].fname %></h2><br>
10          <div class="table-responsive">
11            <table class="w-100">
12              <tr>
13                <th>ห้อง:</th>
14                <th>ค้างชำระ:</th>
15                <th>ฟรุต:</th>
16              </tr>
17              <% data.forEach((item)=>[ %>
18                <tr>
19                  <td><%= item.room_id %></td>
20                  <td>
21                    <% if (item.bill_status == null){ %>
22                      ใหม่
23                    <%}else if (item.bill_status == 0) { %>
24                      ค้างชำระ
25                    <%}else if (item.bill_status == 1) { %>
26                      ชำระแล้ว
27                    <% } %>
28                  </td>
29                  <td><%= item.parcel_count %> ชิ้น</td>
30                </tr>
31              <% }); %>
32            </table>
33          </div>
34        </div>
35      </div>
36    </div>
37  </div>
38  <div class="bg-white p-3 mt-4 rounded shadowing">
39    <h5>ค่าใช้จ่ายแผลงเดือน</h5>
40    <hr>
41    <canvas id="barChart"></canvas>
42  </div>
43  <div class="bg-white p-3 mt-4 rounded shadowing">
44    <h5>รายการ(ที่ลัง) ชื่อ_m </h5>
45    <hr>
46    <table class="table py-2 request-fix-table text-center" id="no-more-tables">
47      <thead class="table-dark">
48        <tr>
49          <th>ประเภทงาน</th>
50          <th>รายละเอียด</th>
51          <th>ค่าใช้จ่ายเพิ่มเติม</th>

```

```

views > home.ejs > div > div.bg-white.p-3.mt-3.w-75.rounded-5.shadowing.row > div.col-3
1   <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

```

บรรทัดนี้ทำหน้าที่ โหลดไลบรารี Chart.js ผ่าน CDN (Content Delivery Network) ซึ่งช่วยให้สามารถใช้งาน Chart.js ได้โดยไม่ต้องดาวน์โหลดไฟล์มาติดตั้งเอง  
Chart.js ใช้สำหรับแสดงข้อมูลในรูปแบบกราฟ เช่น กราฟแท่ง (Bar Chart), กราฟเส้น (Line Chart) ฯลฯ

## ตรวจสอบว่าผู้ใช้เข้าสู่ระบบหรือไม่

ในส่วนแรกของโค้ด จะใช้ข้อมูลจากตัวแปร data ที่ถูกส่งมาจากเซิร์ฟเวอร์เพื่อตรวจสอบว่าผู้ใช้ได้เข้าสู่ระบบหรือไม่ โดยอาศัยค่าจาก data[0].fname ซึ่งเป็นชื่อของผู้ใช้หากไม่มีข้อมูลใน data อาจหมายถึงผู้ใช้ยังไม่ได้เข้าสู่ระบบ และในกรณีที่ระบบรองรับการตรวจสอบสิทธิ์เพิ่มเติม อาจต้องมีการเปลี่ยนเส้นทางไปยังหน้าเข้าสู่ระบบ

## แสดงผลโลโก้ และข้อความต้อนรับ

เมื่อผู้ใช้เข้าสู่ระบบแล้ว ระบบจะแสดงโลโก้ของเว็บไซต์ (LOGO.png) ที่อยู่ภายใต้  ซึ่งถูกจัดให้อยู่กึ่งกลางด้านบน อาจจะเป็นการแสดงข้อความต้อนรับ โดยตึงชื่อของผู้ใช้จาก data[0].fname เพื่อแสดงผลข้อความ "ยินดีต้อนรับ, [ชื่อผู้ใช้]"

## แสดงข้อมูลเกี่ยวกับห้องพักและสถานะการชำระเงิน

มีการใช้ตาราง `<table>` เพื่อแสดงรายละเอียดของห้องพักที่ผู้ใช้เช่าอยู่ โดยแต่ละแถวจะแสดงข้อมูลดังนี้ หมายเลขห้อง (room\_id), สถานะการชำระเงิน (bill\_status), จำนวนพัสดุที่ได้รับ (parcel\_count) จำนวนไฟข้างต้นใช้เพื่อตรวจสอบสถานะการชำระเงิน และแสดงข้อความที่เหมาะสมให้กับผู้ใช้

## แสดงกราฟค่าใช้จ่ายแต่ละเดือน

ระบบมีการใช้ไลบรารี Chart.js เพื่อแสดงกราฟแท่ง (bar chart) สำหรับแสดงข้อมูลค่าใช้จ่ายของผู้ใช้ในแต่ละเดือน

กำหนดข้อมูลแต่ละเดือนเป็น labels, ดึงยอดค่าใช้จ่ายจากตัวแปร total\_amount เพื่อใช้เป็นค่าบันทึก จากนั้นสร้างกราฟแท่งโดยใช้ Chart.js

## ตารางแสดงรายการแจ้งซ่อม

นอกจากข้อมูลเกี่ยวกับค่าใช้จ่าย ระบบยังรองรับการแสดงข้อมูลการแจ้งซ่อม โดยใช้ตาราง `<table>`

- type → ประเภทงานที่แจ้งซ่อม
- detail → รายละเอียดเพิ่มเติม
- cost → ค่าใช้จ่ายเพิ่มเติม (ถ้ามี)
- status → แสดงสถานะของการแจ้งซ่อม (แจ้งเรื่อง หรือ กำลังดำเนินการ)

```

63     ..... <% if (item.status === 0) { %>
64     ..... แจ้งเรื่อง
65     ..... <% } else { %>
66     ..... กำลังดำเนินการ
67     ..... <% } %>
68     ..... </td>
69     ..... <td data-title="วันที่แจ้งข้อม"><%=item.request_date %></td>
70     ..... </tr>
71     ..... <% } ); %>
72     ..... </tbody>
73     ..... </table>
74   </div>
75   <script>
76     //const data = <%= JSON.stringify(bill) %>;
77   const data = [
78     ..... { created_at: '2025-01-01', total_amount: 5200 },
79     ..... { created_at: '2025-02-01', total_amount: 5100 },
80     ..... { created_at: '2025-03-01', total_amount: 5500 },
81     ..... { created_at: '2025-04-01', total_amount: 5300 },
82     ..... { created_at: '2025-05-01', total_amount: 5400 },
83     ..... { created_at: '2025-06-01', total_amount: 5200 },
84     ..... { created_at: '2025-07-01', total_amount: 5100 },
85     ..... { created_at: '2025-08-01', total_amount: 5500 },
86     ..... { created_at: '2025-09-01', total_amount: 5300 },
87     ..... { created_at: '2025-10-01', total_amount: 5400 },
88     ..... { created_at: '2025-11-01', total_amount: 5200 },
89     ..... { created_at: '2025-12-01', total_amount: 5100 }
90   ];
91   const labels = data.map(item => item.created_at);
92   const values = data.map(item => item.total_amount);
93   const ctx = document.getElementById('barChart').getContext('2d');
94   new Chart(ctx, {
95     type: 'bar',
96     data: {
97       labels: labels,
98       datasets: [{
99         label: 'ค่าใช้จ่าย',
100        data: values,
101        backgroundColor: 'rgba(54, 162, 235, 0.2)',
102        borderColor: 'rgba(54, 162, 235, 1)',
103        borderWidth: 1
104      }]
105    },
106    options: {
107      scales: {
108        y: {
109          beginAtZero: true
110        }
111      }
112    }
113  });
114 </script>

```

## หน้าเข้าสู่ระบบผู้ใช้งาน(/login)

```

views > login.ejs > html > head > style
1   <!DOCTYPE html>
2   <html lang="en">
3
4   <head>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <link rel="preconnect" href="https://fonts.googleapis.com">
8     <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
9     <link
10        href="https://fonts.googleapis.com/css2?family=K2D:ital,wght@0,100;0,200;0,300;0,400;0,
11        rel="stylesheet">
12   <title>Login</title>
13   <link rel="website icon" type="png" href="img/LOGO.png">
14   <style>
15     form,
16     body {
17       font-family: "K2D", serif;
18
198  <body>
199    <form action="/welcome" method="POST">
200      <div class="container <%= shake ? 'shake' : '' %>">
201        
202        <!--
203        <div>
204          <label for="username"> ឈ្មោះឱ្យចូល : </label>
205          <input type="text" id="username" name="username" value="<%= formdata.username %>" required>
206        </div>
207        <div>
208          <label for="password"> ពាសវណា : </label>
209          <input type="password" id="password" name="password" required>
210          <span class="toggle-password" onclick="togglePassword()></span>
211        </div>
212        <p class="<%= shake ? 'invalid show' : 'invalid' %>">Username or password is incorrect</p>
213        <div>
214          <button type="submit">ចូលរួម</button>
215        </div>
216      </div>
217    </form>
218    <script>
219      function togglePassword() {
220        let passwordInput = document.getElementById("password");
221        let switchImg = document.getElementById("switch");
222        if (passwordInput.type === "password") {
223          passwordInput.type = "text";
224          switchImg.src = '/img/view.png';
225          switchImg.style.transform = "rotate(180deg)";
226        } else {
227          passwordInput.type = "password";
228          switchImg.src = '/img/hide.png';
229          switchImg.style.transform = "rotate(0deg)";
230        }
231      }
232    </script>
233  </body>

```

หน้าเข้าสู่ระบบ (Login Page) เป็นจุดเริ่มต้นของระบบ ที่อนุญาตให้ผู้ใช้งานารถเข้าสู่ระบบได้โดยใช้ชื่อผู้ใช้ (Username) และรหัสผ่าน (Password) ซึ่งข้อมูลที่ผู้ใช้ป้อนจะถูกส่งไปยังเซิร์ฟเวอร์เพื่อตรวจสอบความถูกต้อง หากข้อมูลถูกต้อง ผู้ใช้จะถูกนำเข้าสู่ระบบ หากไม่ถูกต้อง ระบบจะแสดงข้อความแจ้งเตือน

โครงสร้างพื้นฐานของเอกสาร (HTML Structure)

เอกสารนี้โครงสร้าง HTML5 ประกอบด้วยองค์ประกอบสำคัญ ดังนี้:

- <head> : กำหนดการตั้งค่าเบื้องต้น เช่น การเขื่อมต่อ กับ Google Fonts และไอคอนของเว็บไซต์
- <body> : ประกอบด้วยฟอร์มล็อกอิน (<form>) ซึ่งรับค่าชื่อผู้ใช้และรหัสผ่าน
- <script> : โค้ด JavaScript ที่ใช้สำหรับฟังก์ชันเพิ่มเติม เช่น การแสดง/ซ่อนรหัสผ่าน

โครงสร้างหลักของฟอร์มล็อกอิน

ฟอร์มล็อกอินถูกสร้างขึ้นภายใต้แท็ก <form> โดยกำหนดให้ส่งข้อมูลผ่าน HTTP POST ไปยังเซิร์ฟเวอร์ที่ /welcome

- action="/welcome" → ระบุให้ส่งข้อมูลไปที่เซิร์ฟเวอร์ในเส้นทาง /welcome
- method="POST" → ใช้วิธี POST ในการส่งข้อมูล ซึ่งปลอดภัยกว่าการใช้ GET
- class="<% shake ? 'shake' : '' %>" → หากมีการป้อนข้อมูลผิดพลาด ฟอร์มจะมีเอฟเฟกต์ สั่น (shake effect) เพื่อแจ้งเตือน

การแสดงโลโก้ของระบบ

แสดงภาพโลโก้ของระบบ โดยใช้ id="logolmg" เพื่อกำหนดขนาดและตำแหน่ง  
ช่องป้อนข้อมูลชื่อผู้ใช้ (Username)

- <label> → ใช้กำหนดชื่อช่องกรอกข้อมูล
- <input type="text"> → ช่องกรอกข้อมูลสำหรับชื่อผู้ใช้
- required → กำหนดให้ต้องกรอกข้อมูลก่อนส่งฟอร์ม

ช่องป้อนข้อมูลรหัสผ่าน (Password) และปุ่มแสดง/ซ่อนรหัสผ่าน

- <input type="password"> → ช่องกรอกข้อมูลรหัสผ่าน
- <span class="toggle-password"> → ปุ่มแสดง/ซ่อนรหัสผ่าน
- onclick="togglePassword()" → เรียกใช้ฟังก์ชัน JavaScript togglePassword()

แสดงข้อความแจ้งเตือนหากป้อนข้อมูลผิดพลาด

- ข้อความ "Username or password is incorrect" จะปรากฏขึ้นเมื่อชื่อผู้ใช้หรือรหัสผ่านผิดพลาด
- ใช้คลาส shake เพื่อให้กล่องล็อกอินสั่นเมื่อป้อนข้อมูลผิด

### ปุ่มเข้าสู่ระบบ (Login Button)

- เมื่อกดปุ่ม ระบบจะส่งข้อมูลไปที่เซิร์ฟเวอร์เพื่อทำการตรวจสอบ

### ฟังก์ชัน JavaScript สำหรับแสดง/ซ่อนรหัสผ่าน

- เมื่อคลิกที่ไอคอน จะเปลี่ยน type ของ <input> จาก "password" เป็น "text" เพื่อให้มองเห็นรหัสผ่าน
- เปลี่ยนภาพของไอคอนจาก hide.png เป็น view.png ตามสถานะ

### การออกแบบและจัดรูปแบบ (CSS Styling)

- ใช้ display: flex เพื่อจัดให้อยู่ริ่งกลางทั้งแนวตั้งและแนวนอน
- ตั้งค่าพื้นหลังเป็น สีฟ้าอ่อน
- ใช้ box-shadow เพื่อเพิ่มเงาให้ดูมีมิติ
- ใช้ border-radius: 12px เพื่อให้มุมโค้งมน
- เมื่อมีข้อผิดพลาด ฟอร์มจะขยาย-ขยาย-ขวาเป็นเวลา 0.5 วินาที

## หน้าตรวจสอบบันทึกมิเตอร์ไฟฟ้า&น้ำ(/meter)

```

views > meter.ejs > ...
1  <div class="d-flex justify-content-between align-items-center">
2  ...<div class="">
3  ...<h4 class="title">ตรวจสอบบันทึกมิเตอร์/</h4>
4  ...<p>ตรวจสอบบันทึก <span class="text-danger">มิเตอร์</span></p>
5  ...</div>
6  </div>
7  ...
8  <div class="bg-white p-3 mt-3 rounded shadowing">
9  ...<h4 class="title">บันทึกมิเตอร์ไฟฟ้า & น้ำ</h4>
10 ...<hr>
11 ...<div class="row mb-3">
12 ...  <div class="col-12">
13 ...    <label for="room_id" class="form-label fw-bold">เลือกห้องที่ต้องการเช็ค มิเตอร์</label>
14 ...    <select name="room_id" id="room_id" class="form-control" onchange="meterSelectChange(event)" required>
15 ...      <option value="">-</option>
16 ...      <% room.forEach((item)=>{ %>
17 ...        <option value="<%= item.room_id %>">
18 ...          <%= item.room_id %>
19 ...        </option>
20 ...      <% }); %>
21 ...    </select>
22 ...  </div>
23 ...</div>
24 ...<div class="row mb-3">
25 ...  <div class="col-lg-6 col-sm-12 mb-2">
26 ...    <label for="electricmeter" class="form-label text-warning fw-bold">มิเตอร์ค่าไฟ</label>
27 ...    <input type="number" class="form-control mb-2" id="electricmeter" name="electricmeter" required readonly>
28 ...    <p class="p-0 m-0 text-danger fw-bold electric-date">วันที่บันทึกมิเตอร์ไฟฟ้าล่าสุด : (วันที่)</p>
29 ...
30 ...  <div class="col-lg-6 col-sm-12 mb-2">
31 ...    <label for="watermeter" class="form-label text-info fw-bold">มิเตอร์ค่าน้ำ</label>
32 ...    <input type="number" class="form-control mb-2" id="watermeter" name="watermeter" required readonly>
33 ...    <p class="p-0 m-0 text-danger fw-bold water-date">วันที่บันทึกมิเตอร์น้ำล่าสุด : (วันที่)</p>
34 ...
35 ...</div>
36 </div>

```

### ส่วนแสดงหัวข้อและคำอธิบาย

- ใช้ <h4> เพื่อแสดงหัวข้อ "ตรวจสอบบันทึกมิเตอร์"
- ใช้ <p> เพื่อแสดงคำอธิบายเสริมเกี่ยวกับการตรวจสอบทึก
- <span class="text-danger"> ทำให้คำว่า "มิเตอร์" แสดงเป็นสีแดง

### ส่วนเลือกห้องพักเพื่อแสดงค่ามิเตอร์

- ส่วนนี้แสดงเป็น Dropdown (<select>) สำหรับเลือกหมายเลขห้อง
- ใช้ EJS (<%= item.room\_id %>) เพื่อแสดงห้องทั้งหมดที่มีอยู่ในระบบ
- มี onchange="meterSelectChange(event)" เพื่อให้เกิดฟังก์ชันเมื่อเลือกห้อง (ฟังก์ชัน meterSelectChange จะทำงานใน viewMeter.js)

### แสดงค่ามิเตอร์ไฟฟ้าและน้ำของห้องที่เลือก

- แสดงค่ามิเตอร์ไฟฟ้า (electricmeter) และ น้ำ (watermeter)
- ใช้ <input type="number" readonly> เพื่อป้องกันไม่ให้ผู้ใช้แก้ไขค่าเอง

- มี <p class="electric-date"> และ <p class="water-date"> สำหรับแสดงวันที่บันทึกค่ามิเตอร์ล่าสุด แสดงอัตราค่าบริการต่อหน่วย
  - ใช้ EJS (<%= item.elec\_rate %> และ <%= item.water\_rate %>) ดึงอัตราค่าไฟและค่าน้ำจากฐานข้อมูล
  - <input type="number" readonly> ป้องกันไม่ให้ผู้ใช้เปลี่ยนแปลงค่า
  - <p class="text-danger">หมายเหตุ : ราคายังคง : ราคายังคงที่บันทึกทั้งหมด</p> เพื่อแจ้งให้ผู้ใช้ทราบว่าราคนี้ใช้กับทุกห้อง

ใช้งาน Java Script สำหรับดึงค่ามิเตอร์

• 58 <script src="viewMeter.js"></script>

ไฟล์ viewMeter.js จะทำหน้าที่ดึงค่ามิเตอร์จากฐานข้อมูล เมื่อลูกค้าเลือกห้อง ไฟล์ viewMeter.js จะมีฟังก์ชัน meterSelectChange(event) ซึ่งจะตรวจสอบค่ามิเตอร์ของห้องที่เลือก, อัปเดตค่ามิเตอร์ไฟฟ้า, อัปเดตค่ามิเตอร์น้ำ, และแสดงวันที่บันทึกค่ามิเตอร์ล่าสุด

## หน้าส่งเรื่องแจ้งซ่อม(/repair)

โค้ดนี้เป็นหน้า ส่งเรื่องแจ้งซ่อม ในระบบจัดการห้องพัก โดยช่วยให้ผู้ใช้งานสามารถแจ้งซ่อมปัญหาต่าง ๆ ที่เกิดขึ้นภายในห้องพัก พร้อมตรวจสอบสถานะการซ่อม เช่น ยืนยันหลัง ส่วนหัวข้อและคำอธิบาย

```
views > repair.ejs > ...
● 1 <div class="d-flex justify-content-between align-items-center">
  2 <div>
    3 <h4 class="title">ส่งเรื่องแจ้งซ่อม</h4>
    4 <p>จัดการ <span class="text-danger">แจ้งซ่อม</span></p>
  5 </div>
  6 </div>
  7
```

- <h4 class="title">ส่งเรื่องแจ้งซ่อม</h4> → แสดงหัวข้อของหน้า
- <p>จัดการ <span class="text-danger">แจ้งซ่อม</span></p> → คำอธิบายเนื้อหา (คำว่า “แจ้งซ่อม” ถูกเน้นเป็นสีแดงเพื่อดึงดูดสายตา)

แบบฟอร์มแจ้งซ่อม

```
8 <div class="bg-white p-3 mt-3 rounded shadowing">
  9 <h4 class="title">ส่งเรื่องแจ้งซ่อม</h4>
 10 <hr>
 11 <form action="/request-repair" method="post">
 12   <input type="text" name="room_id" placeholder="ห้องพัก">
```

- ส่วนนี้เป็น ฟอร์มแจ้งซ่อม ที่ใช้ method="post" และส่งข้อมูลไปยัง /request-repair
- พื้นหลังสีขาว (bg-white), มีเงา (shadowing) และขอบมน (rounded) เพื่อความสวยงาม

เลือกห้องพักที่ต้องการแจ้งซ่อม

```
13 <div class="col-lg-6 col-sm-12 mb-2">
  14   <label for="room_id" class="form-label fw-bold">เลือกห้องที่ต้องการแจ้ง</label>
  15   <select name="room_id" id="room_id" class="form-control required">
  16     <option value="">-</option>
  17     <% room.forEach(item=>{ %>
  18       <option value="<%= item.room_id %>"><%= item.room_id %></option>
  19     <% }); %>
  20   </select>
  21 </div>
```

- ใช้ Dropdown (<select>) ให้ผู้ใช้เลือกหมายเลขห้องพักที่ต้องการแจ้งซ่อม
- ใช้ EJS (<%= item.room\_id %>) เพื่อดึงข้อมูลหมายเลขห้องจากฐานข้อมูล

### เลือกประเภทงานซ่อม

```

22 |     <div class="col-lg-6 col-sm-12 mb-2">
23 |       <label for="type" class="form-label fw-bold">ประเภทงาน</label>
24 |       <select name="type" id="type" class="form-control" required>
25 |         <option value="">-</option>
26 |         <option value="ฉุกเฉิน">ฉุกเฉิน</option>
27 |         <option value="เร่งด่วน">เร่งด่วน</option>
28 |         <option value="อื่นๆ">อื่นๆ</option>
29 |
30 |     </div>

```

- Dropdown สำหรับเลือก ประเภทของงานซ่อม มี 3 ประเภท : ฉุกเฉิน, เร่งด่วน, อื่นๆ

### รายละเอียดการแจ้งซ่อม

```

32 |     <div class="row mb-2">
33 |       <div class="col-12">
34 |         <label for="detail" class="form-label fw-bold">รายละเอียด</label>
35 |         <input type="text" class="form-control mb-2" id="detail" name="detail" required>
36 |
37 |     </div>

```

- ช่อง <input type="text"> ให้ผู้ใช้กรอกรายละเอียดปัญหาที่ต้องการแจ้งปั๊มส่งฟอร์มแจ้งซ่อม
- ปุ่มสีเขียว (btn-success) ขนาดใหญ่ (btn-lg) เพื่อให้มองเห็นได้ชัด
- เมื่อกดปุ่ม ข้อมูลจะถูกส่งไปยัง /request-repair ผ่าน POST Request

### ตารางประวัติการแจ้งซ่อม

```

42 <div class="bg-white p-3 mt-4 rounded shadowing">
43   <h5 class="title">ประวัติการแจ้งซ่อม</h5>
44   <hr>
45   <table class="table py-2 request-fix-table text-center" id="no-more-tables">

```

- แสดง ประวัติการแจ้งซ่อม ในรูปแบบตาราง
- ตารางใช้ class="table request-fix-table text-center" เพื่อจัดให้อยู่กึ่งกลาง

## หัวข้อ colum ของตาราง

```

42 <div class="bg-white p-3 mt-4 rounded shadowing">
43   <h5 class="title">ประวัติการแจ้งซ่อม</h5>
44   <hr>
45   <table class="table py-2 request-fix-table text-center" id="no-more-tables">
46     <thead class="table-dark">
47       <tr>
48         <th>ห้องพัก</th>
49         <th>ประเภทงาน</th>
50         <th>รายละเอียด</th>
51         <th>วันที่แจ้งซ่อม</th>
52         <th>ค่าใช้จ่ายเพิ่มเติม</th>
53         <th>สถานะ</th>
54     </tr>
55   </thead>

```

- ใช้ <thead class="table-dark"> ทำให้หัวตารางเป็นสีเข้ม
- คอลัมน์แสดงข้อมูล หมายเลขห้อง, ประเภทงาน, รายละเอียด, วันที่แจ้ง, ค่าใช้จ่าย และ สถานะ แสดงข้อมูลแจ้งซ่อมที่ได้มาจากฐานข้อมูล

```

56   <tbody>
57     <% data.forEach((item)=>{ %>
58       <tr>
59         <td data-title="ห้องพัก"><%=item.room_id %></td>
60         <td data-title="ประเภทงาน"><%=item.type %></td>
61         <td data-title="รายละเอียด"><%=item.detail %></td>
62         <td data-title="วันที่แจ้งซ่อม"><%=item.request_date %></td>
63         <td data-title="ค่าใช้จ่ายเพิ่มเติม"><%=item.cost %></td>
64         <td data-title="สถานะ">
65           <% if (item.status === 0) { %>
66             แจ้งเรื่อง
67           <% } else if (item.status === 1) { %>
68             กำลังดำเนินการ
69           <% } else { %>
70             เสร็จสิ้น
71           <% } %>
72         </td>
73       </tr>
74     <% }); %>
75   </tbody>

```

- ใช้ EJS (<%= item.room\_id %>) ดึงข้อมูลแจ้งซ่อมจากฐานข้อมูล

- <td data-title="สถานะ"> กำหนดสถานะของการแจ้งซ่อม: status === 0 → "แจ้งเรื่อง", status === 1 → "กำลังดำเนินการ", status === 2 → "เสร็จสิ้น"
  - 1) ฟอร์มแจ้งซ่อม → ให้ผู้ใช้เลือกห้อง, ประเภทงาน และกรอกรายละเอียดก่อนกดปุ่มแจ้งซ่อม
  - 2) ตารางแสดงประวัติการแจ้งซ่อม → ดึงข้อมูลจากฐานข้อมูลมาแสดง
  - 3) สถานะการแจ้งซ่อม → เปลี่ยนไปตามสถานะจริงของงานซ่อม
  - 4) รองรับการแจ้งซ่อมหลายประเภท → ทั้งฉุกเฉิน, เร่งด่วน และอื่น ๆ
  - 5) ใช้งานง่าย → มี UI ที่เป็นระเบียบ ปุ่มเด่นชัด

## หน้าแจ้งเตือนพัสดุ(/parcel)

หน้านี้ใช้สำหรับ แจ้งเตือนพัสดุ ที่ไม่มีห้องพัก โดยให้ผู้พักอาศัยสามารถตรวจสอบพัสดุที่ยังไม่ได้รับ และคุ้มครองสิ่งของในห้อง

ส่วนหัวข้อและคำอธิบาย

```

● 1 <div class="d-flex justify-content-between align-items-center">
  2 <div class="">
  3   <% if (data3.length > 0 && data3[0].room_id) { %>
  4     <h4 class="title">แจ้งเตือนพัสดุ - ของห้อง <span class="fw-bold"><%= data3[0].room_id %></span></h4>
  5   <% } else { %>
  6     <h4 class="title">แจ้งเตือนพัสดุ - ไม่มีห้องพัก</h4>
  7   <% } %>
  8   <p>ข้อมูล <span class="text-danger">แจ้งเตือนพัสดุ</span></p>
  9 </div>
10 </div>
```

- ตรวจสอบว่าผู้ใช้มีหมายเลขห้องพักหรือไม่
- ถ้ามี data3[0].room\_id → แสดงหมายเลขห้องพักของผู้ใช้
- ถ้าไม่มี → แสดงข้อความ "แจ้งเตือนพัสดุ - ไม่มีห้องพัก"

ตารางพัสดุที่กำลังรอรับ

```

12 <div class="bg-white p-3 mt-3 rounded shadowing">
13   <h5 class="title">พัสดุที่กำลังรอรับ</h5>
14   <hr>
```

- แสดงพัสดุที่ยังไม่ได้รับ
- ใช้พื้นหลังสีขาว (bg-white), มีเงา (shadowing) และขอบมน (rounded)

## ส่วนหัวของตาราง

```

12  <div class="bg-white p-3 mt-3 rounded shadow">
13  |   <h5 class="title">พัสดุที่กำลังรอรับ</h5>
14  |   <hr>
15  |   <table class="table py-2 wait-parcel-table" id="no-more-tables">
16  |       <thead class="align-middle">
17  |           <tr>
18  |               <th>ห้องพัก</th>
19  |               <th>ชื่อผู้รับ</th>
20  |               <th>ขนาดพัสดุ</th>
21  |               <th>วันที่พัสดุถึง</th>
22  |               <th class="text-center">ยืนยันการรับพัสดุ</th>
23  |           </tr>
24  |       </thead>

```

- colum ห้องพัก, ชื่อผู้รับ, ขนาดพัสดุ, วันที่พัสดุมาถึง และ ปุ่มยืนยันรับพัสดุ

## แสดงรายการพัสดุที่ยังไม่ได้รับ

```

25      <tbody>
26          <% data1.forEach((item)=>{ %>
27              <tr>
28                  <td data-title="#"><%=item.room_id%></td>
29                  <td data-title="ชื่อผู้รับ"><%=item.receiver_name%></td>
30                  <td data-title="ขนาดพัสดุ">
31                      <% if (item.size === 'S') { %>
32                          ขนาดเล็ก
33                      <% } else if (item.size === 'M') { %>
34                          ขนาดกลาง
35                      <% } else { %>
36                          ขนาดใหญ่
37                      <% } %>
38                  </td>
39                  <td data-title="วันที่พัสดุถึง"><%=item.arrival_date%></td>
40                  <td data-title="ยืนยันการรับพัสดุ" class="text-center">
41                      <form action="/accept-parcel/<%=item.id%>" method="post">
42                          <button type="submit" class="btn btn-success">ยืนยันการรับ</button>
43                      </form>
44                  </td>
45              </tr>
46          <%}); %>
47      </tbody>

```

- ใช้ data1.forEach((item)=>{ ... }) เพื่อวนลูปแสดงข้อมูลพัสดุที่ยังไม่ได้รับ
- แสดง ชื่อผู้รับ, หมายเลขห้อง, วันที่พัสดุมาถึง และ ขนาดพัสดุ
- ปุ่ม "ยืนยันการรับ":
  - ใช้ <form action="/accept-parcel/<%=item.id%>" method="post">
  - เมื่อกดปุ่ม ระบบจะส่ง POST Request ไปที่ /accept-parcel/:id
  - ใช้ btn-success เพื่อให้ปุ่มเป็นสีเขียว

## ตารางประวัติการรับพัสดุ

```

51  <div class="bg-white p-3 mt-4 rounded shadowing">
52  |    <h5 class="title">ประวัติการรับพัสดุ</h5>
53  |    <hr>
```

- แสดงพัสดุที่รับไปแล้ว
- ใช้พื้นหลังสีขาว (bg-white), มีเงา (shadowing) และขอบมน (rounded)

### ส่วนหัวของตาราง

```

55  <thead class="align-middle table-dark">
56  |    <tr>
57  |        <th>ห้องพัก</th>
58  |        <th>ชื่อผู้รับ</th>
59  |        <th>ขนาดพัสดุ</th>
60  |        <th>วันที่พัสดุถึง</th>
61  |        <th>วันที่ยืนยันรับพัสดุ</th>
62  |        <th>สถานะ</th>
63  |    </tr>
64  </thead>
```

- colum ห้องพัก, ชื่อผู้รับ, ขนาดพัสดุ, วันที่พัสดุถึง, วันที่รับพัสดุ และ สถานะ

### แสดงประวัติการรับพัสดุ

```

65  <tbody>
66  |    <% data2.forEach((item)=>{ %>
67  |        <tr>
68  |            <td data-title="ห้องพัก"><%=item.room_id%></td>
69  |            <td data-title="ชื่อผู้รับ"><%=item.receiver_name%></td>
70  |            <td data-title="ขนาดพัสดุ">
71  |                <% if (item.size === 'S') { %>
72  |                    ขนาดเล็ก
73  |                <% } else if (item.size === 'M') { %>
74  |                    ขนาดกลาง
75  |                <% } else { %>
76  |                    ขนาดใหญ่
77  |                <% } %>
78  |            </td>
79  |            <td data-title="วันที่พัสดุถึง"><%=item.arrival_date%></td>
80  |            <td data-title="วันที่ยืนยันรับพัสดุ"><%=item.receive_date%></td>
81  |            <td data-title="สถานะ"><%=item.status === 1 ? 'รับแล้ว' : 'ยังไม่รับ'%></td>
82  |        <% } %>
83  |    </tr>
84  </tbody>
85 </div>
```

- ใช้ EJS (<%= item.room\_id %>) ดึงข้อมูลพัสดุที่เคยได้รับ
- แสดง หมายเลขห้อง, ชื่อผู้รับ, วันที่พัสดุมาถึง, วันที่รับพัสดุ และ สถานะ
- status === 1 ? 'รับแล้ว' : 'ยังไม่รับ' → เปลี่ยนสถานะตามข้อมูลจริง

- 1) แจ้งเตือนพัสดุ → แสดงหมายเลขอหงของผู้ใช้
- 2) ตารางพัสดุที่กำลังรอรับ → ตรวจสอบรายการพัสดุที่ยังไม่ได้รับ
- 3) ปุ่มยืนยันรับพัสดุ → ส่ง POST Request เพื่ออัปเดตสถานะพัสดุ
- 4) ตารางประวัติการรับพัสดุ → แสดงพัสดุที่เคยรับไปแล้ว
- 5) แสดงขนาดพัสดุ → เล็ก, กลาง, ใหญ่

## หน้าเริ่มต้นของแอดมิน(/admin)

โค้ดที่ให้มาคือหน้าแรกของแอดมิน (/admin) ซึ่งแสดงข้อมูลสถิติและสรุปสถานะต่างๆ ของห้องพัก รวมถึง กราฟและตารางต่างๆ เพื่อช่วยให้แอดมินจัดการห้องพักได้ง่ายขึ้น โดยมีองค์ประกอบสำคัญดังนี้:

```

1 <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
2 <script src="https://cdn.jsdelivr.net/npm/chartjs-plugin-datalabels"></script>
3 <h4 class="title">ยินดีต้อนรับ, <%= data[0].fname %></h4>
4 <div class="card shadow" style="padding-bottom: 50px;">
5   <div class="d-flex flex-wrap justify-content-center bg-white">
6     <div class="card m-2 shadow" style="width: 34rem;">
7       <h5 class="card-title me-3 text-center mt-4">จำนวนห้อง</h5>
8       <canvas id="roomChart"></canvas>
9     </div>
10    <div class="card m-2 shadow" style="width: 34rem; padding-bottom: 50px;">
11      <h5 class="card-title me-3 text-center mt-4">เพศ</h5>
12      <canvas id="sexChart"></canvas>
13    </div>

```

- แสดงข้อความต้อนรับแอดมิน โดยดึงชื่อ (fname) มาจาก data[0]

ใช้ Chart.js สร้างกราฟ 2 แบบ:

1. roomChart : แสดงจำนวนห้องที่ว่างและไม่ว่าง
2. sexChart : แสดงสัดส่วนเพศของผู้พักอาศัย

## การสร้าง Chart.js

```

139 <script>
140     // แปลง id จาก 'myChart' เป็น 'roomChart'
141     const roomChartData = {
142         type: 'pie',
143         labels: ['ห้องว่าง', 'ห้องไม่ว่าง'],
144         dataValues: [<%= availableRooms %>, <%= occupiedRooms %>],
145         backgroundColors: ['rgba(255, 255, 255, 1)', 'rgba(63, 114, 175, 0.8)'],
146         borderColors: ['rgba(214, 219, 224, 1)', 'rgba(44, 89, 138, 1)'],
147         label: 'จำนวนห้อง',
148         tooltipsCallback: (tooltipItem) => `${tooltipItem.label}: ${tooltipItem.raw} ห้อง`
149     };
150
151     const sexChartData = {
152         type: 'pie',
153         labels: ['ชาย', 'หญิง', 'อื่นๆ'],
154         dataValues: [
155             <%= (sexData.filter(s => s.sex === 'M')[0] || {}).count || 0 %>,
156             <%= (sexData.filter(s => s.sex === 'F')[0] || {}).count || 0 %>,
157             <%= (sexData.filter(s => s.sex === 'O')[0] || {}).count || 0 %>
158         ],
159         backgroundColors: ['rgba(54, 162, 235, 0.8)', 'rgba(255, 99, 132, 0.8)', 'rgba(255, 206, 86, 0.8)'],
160         borderColors: ['rgba(54, 162, 235, 1)', 'rgba(255, 99, 132, 1)', 'rgba(255, 206, 86, 1)'],
161         label: 'เพศของผู้ใช้',
162         tooltipsCallback: (tooltipItem) => `${tooltipItem.label}: ${tooltipItem.raw} คน`
163     };
164
165 
```

- ใช้ availableRooms และ occupiedRooms ใน roomChartData เพื่อคำนวณจำนวนห้องว่างกับไม่ว่าง
- ใช้ sexData ใน sexChartData เพื่อดึงจำนวนผู้พักแต่ละเพศ

### ตารางข้อมูลที่สำคัญ

ห้องที่ต้องออกใบแจ้งหนี้

```

<% bill
  .filter(item =>
    currentTime.setDate(currentTime.getMonth() - 1) &&
    new Date(item.created_at) <= currentTime)
  .forEach((item)=>%
    <tr>
      <td data-title="ห้อง"><%= item.room_id %></td>
    </tr>
  <% ); %>
</tbody>
```

- แสดงห้องที่ต้องออกใบแจ้งหนี้ โดยคัดกรอง (filter) จากวันที่ (created\_at) ก่อนหรือเท่ากับ วันที่ปัจจุบัน

## ห้องที่ยังไม่จ่าย

```

54      <tbody>
55          <% notyet.forEach((item)=>{ %>
56              <tr>
57                  <td data-title="ห้อง"><%= item.room_id %></td>
58                  <td data-title="ค้างชำระ"><%= item.total_amount %></td>
59                  <td data-title="วันที่สร้าง"><%= item.created_at %></td>
60              </tr>
61          <% }); %>
62      </tbody>

```

- แสดงห้องที่มีค่าค้างชำระ (total\_amount)

## วันที่เข้า-ออกห้องนี้

```

79      <tbody>
80          <% checkin.forEach((item)=>{ %>
81              <tr>
82                  <td data-title="ห้อง"><%= item.room_id %></td>
83                  <td data-title="ชื่อเจ้าของห้อง"><%= item.fname %></td>
84              </tr>
85          <% }); %>
86      </tbody>
87  </table>
88 </div>
89 </div>
90 <div class="col">
91     <div class="bg-white p-4 mt-3 rounded shadowing">
92         <h5 class="title">ห้องที่เข้าออกห้องนี้ (<%= currentDate %>)</h5>
93         <hr>
94         <table class="table py-2 text-center menu-table" id="no-more-tables">
95             <thead class="table-dark">
96                 <tr>
97                     <th>ห้อง</th>
98                     <th>ชื่อเจ้าของห้อง</th>
99                 </tr>
100            <thead>
101            <tbody>
102                <% checkout.forEach((item)=>[ %>
103                    <tr>
104                        <td data-title="ห้อง"><%= item.room_id %></td>
105                        <td data-title="ชื่อเจ้าของห้อง"><%= item.fname %></td>
106                    </tr>
107                <% ]); %>
108            </tbody>
109        </table>

```

- ห้องที่เข้า-ออก (checkin) และย้ายออก (checkout) แสดงตามวันที่ (currentDate)

## ห้องที่เกินกำหนดชำระ

```

112 <div class="col">
113   <div class="bg-white p-4 mt-3 rounded shadowing">
114     <h5 class="title">ห้องที่เกินกำหนดชำระ</h5>
115     <hr>
116     <table class="table py-2 text-center menu-table" id="no-more-tables">
117       <thead class="table-dark">
118         <tr>
119           <th>ห้อง</th>
120           <th>ค่างวด</th>
121           <th>วันที่ต้องชำระ</th>
122           <th>ยอดค้างชำระ(บาท)</th>
123         </tr>
124       </thead>
125       <tbody>
126         <% overdue.forEach((item)=>{ %>
127           <tr>
128             <td data-title="ห้อง"><%= item.room_id %></td>
129             <td data-title="ค่างวด"><%= item.total_amount %></td>
130             <td data-title="วันที่ต้องชำระ"><%= item.due_date %></td>
131             <td data-title="ยอดค้างชำระ(บาท)"><%= new Date(currentDate).getDate() - new Date(item.due_date).getDate() %></td>
132           </tr>
133         <% }); %>
134       </tbody>
135     </table>
136   </div>
137 </div>
138 </div>
```

- คำนวณวันที่เกินกำหนด (overdue) โดยลบวันที่ต้องชำระ (due\_date) ออกจากวันที่ปัจจุบัน (currentDate)

## สคริปต์สร้างกราฟ

```

167   function createChart(chartId, chartData) {
168     const data = {
169       labels: chartData.labels,
170       datasets: [
171         {
172           label: chartData.label,
173           data: chartData.dataValues,
174           backgroundColor: chartData.backgroundColors,
175           borderColor: chartData.borderColors,
176           borderWidth: 2
177         }
178     };
179
180     const ctx = document.getElementById(chartId).getContext('2d');
181     new Chart(ctx, [
182       type: chartData.type,
183       data: data,
184       options: {
185         responsive: true,
186         maintainAspectRatio: true,
187         plugins: {
188           legend: {
189             display: true,
190             position: 'bottom',
191             labels: {
192               font: {
193                 size: 14,
194                 weight: 'bold',
195               },
196             },
197             dataLabels: {
198               color: '#000',
199               font: {
200                 weight: 'bold',
201                 size: 14
202               },
203               align: 'center',
204               formatter: (value) => `${value}`,
205             },
206           },
207           tooltips: {
208             enabled: true,
209             callbacks: {
210               label: chartData.tooltipsCallback,
211             }
212           },
213           ...
214         }
215       }
216     });
217   }
218 }
```

ฟังก์ชัน `createChart(chartId, chartData)` เป็น ฟังก์ชันสร้างกราฟ ซึ่งรับ 2 พารามิเตอร์ `chartId` : id ของ `<canvas>` ที่ต้องการวาดกราฟ

`chartData` : ขอบเขตที่เก็บข้อมูลต่างๆ ของกราฟ เช่น ประเภทกราฟ (`type`), ข้อมูล (`dataValues`), สี (`backgroundColors`), และตัวช่วยแสดงผล (`tooltipsCallback`)

### สร้างข้อมูลสำหรับกราฟ (data)

`labels` : ป้ายกำกับของข้อมูล เช่น ["ห้องว่าง", "ห้องไม่ว่าง"]

`datasets` : กำหนดข้อมูลของกราฟ:

- `label` : ชื่อของกราฟ
- `data` : ค่าตัวเลขของแต่ละหมวดหมู่ (เช่น จำนวนห้องว่าง-ไม่ว่าง)
- `backgroundColor` : สีของแต่ละส่วนใน Pie Chart
- `borderColor` : สีของขอบกราฟ
- `borderWidth` : ความหนาของเส้นขอบ (ตั้งค่าเป็น 2)

### คืนหา `<canvas>` และสร้างกราฟ

ใช้ `document.getElementById(chartId).getContext('2d')` เพื่อดึง Canvas API และใช้กับ Chart.js  
`new Chart(ctx, { ... })` : สร้างกราฟโดยใช้ค่าที่กำหนด

### กำหนด options ของกราฟ

`responsive: true` : กราฟสามารถปรับขนาดตามหน้าจอได้

`maintainAspectRatio: true` : รักษาสัดส่วนของกราฟให้เหมาะสม

### แสดงคำอธิบายกราฟ (legend)

`legend` : คำอธิบายสีของกราฟ

`display: true` : แสดง legend

`position: 'bottom'` : วางคำอธิบายไว้ด้านล่างกราฟ

ปรับขนาดตัวอักษร (`size: 14`) และทำให้ตัวหนา (`weight: 'bold'`)

### ป้ายกำกับข้อมูลบนกราฟ (datalabels)

ใช้ Chart.js Plugin - DataLabels แสดงตัวเลขบนกราฟ

`color: #000` : สีดำ

`font.weight: bold` : ตัวหนา

`align: center` : จัดตรงกลาง

`formatter: (value) => ${value}`` : กำหนดให้แสดงค่าของแต่ละส่วน

### กำหนด tooltips (คำแนะนำเมื่อ Hover)

enabled: true : เปิดใช้งาน tooltips

callbacks.label: chartData.tooltipsCallback : ใช้ฟังก์ชันที่กำหนดใน

chartData.tooltipsCallback : เพื่อกำหนดรูปแบบข้อความที่แสดงเมื่อชี้เม้าส์

### กำหนด animation (เอฟเฟกต์แสดงกราฟ)

duration: 1000 : ใช้เวลา 1 วินาที (1000 มิลลิวินาที) ในการแสดงผล

easing: 'easeOutBounce' : ใช้เอฟเฟกต์เด้งเมื่อโหลดกราฟ

โหลดฟังก์ชัน createChart() เมื่อหน้าเว็บเสร็จสมบูรณ์

```
222 window.onload = function() {
223     createChart('roomChart', roomChartData);
224     createChart('sexChart', sexChartData);
225 }
```

ใช้ window.onload เพื่อรอให้หน้าเว็บโหลดเสร็จก่อน แล้วค่อยสร้างกราฟ

เรียก createChart() 2 ครั้ง:

1. createChart('roomChart', roomChartData); → สร้างกราฟ จำนวนห้อง
2. createChart('sexChart', sexChartData); → สร้างกราฟ สัดส่วนเพศของผู้พัก

## หน้าคำขอแจ้งซ่อมของแอดมิน(/fixpage)

โค้ดนี้ใช้ HTML + Embedded JavaScript (EJS) ในการแสดงข้อมูลจากเซิร์ฟเวอร์ มีองค์ประกอบหลักดังนี้:

1. ตารางแสดง คำขอแจ้งซ่อมปัจจุบัน (เฉพาะที่ยังไม่เสร็จสิ้น)
2. ฟอร์ม อัปเดตสถานะการซ่อม
3. ตารางแสดง ประวัติการซ่อมที่เสร็จสิ้น

### แนวคิดหลักของโค้ดนี้

- ใช้ EJS (Embedded JavaScript) เพื่อดึงข้อมูลจากเซิร์ฟเวอร์แล้วแสดงผลบนหน้าเว็บ
- ใช้ HTML + Bootstrap ในการออกแบบ UI ที่สวยงามและใช้งานง่าย
- ใช้ POST request เพื่ออัปเดตสถานะของคำขอซ่อมผ่านฟอร์ม
- มีการแสดงข้อมูลแบบไดนามิก โดยกรองเฉพาะคำขอที่ ยังไม่เสร็จสิ้น และคำขอที่ เสร็จสิ้นแล้ว

```

8 <div class="bg-white p-3 mt-3 rounded shadowing">
9   <!-- ใช้รูปแบบแจ้งซ่อม / กำลังดำเนินการ -->
10  <table class="table py-2 request-fix-table" id="no-more-tables">
11    <thead>
12      <tr>
13        <th>ห้องพัก</th>
14        <th>ประเภทงาน</th>
15        <th>รายละเอียด</th>
16        <th>วันที่แจ้งซ่อม</th>
17        <th>สถานะ</th>
18      </tr>
19    </thead>
20    <tbody>
21      <% data.forEach((item)=>{ %>
22        <tr>
23          <td data-title="ห้องพัก"><%=item.room_id %></td>
24          <td data-title="ประเภทงาน"><%=item.type %></td>
25          <td data-title="รายละเอียด"><%=item.detail %></td>
26          <td data-title="วันที่แจ้งซ่อม"><%=item.request_date %></td>
27          <td data-title="สถานะ">
28            <% if (item.status === 0) { %>
29              แจ้งเรื่อง
30            <% } else if (item.status === 1) { %>
31              กำลังดำเนินการ
32            <% } else { %>
33              เสร็จสิ้น
34            <% } %>
35          </td>
36        </tr>
37      <% }); %>
38    </tbody>
39  </table>
40 </div>
```

## แสดงรายการคำขอแจ้งซ่อม

- แสดงรายการคำขอแจ้งซ่อมที่ยัง ไม่ได้ดำเนินการหรือกำลังดำเนินการ
  - ใช้ EJS วนลูป data.forEach() เพื่อนำข้อมูลแต่ละรายการมาแสดง
  - แปลงค่า status เป็นข้อความที่อ่านเข้าใจง่าย
- ดึงข้อมูลคำขอแจ้งซ่อมจากตัวแปร data ใช้ forEach วนลูปแสดงรายการแต่ละอัน

## แสดง สถานะการแจ้งซ่อม

0 : แจ้งเรื่อง, 1 : กำลังดำเนินการ, 2 : เสร็จสิ้น

## ฟอร์มอัปเดตสถานะ

- ให้แอดมิน เลือกคำขอซ่อม ที่ต้องการอัปเดตสถานะ
- กรอก ค่าใช้จ่ายเพิ่มเติม (หากมี)
- เลือก สถานะใหม่ (แจ้งเรื่อง, กำลังดำเนินการ, เสร็จสิ้น)
- เมื่อกด "ยืนยันการเปลี่ยนสถานะ" → ส่งข้อมูลไปยังเซิร์ฟเวอร์ ผ่าน /managefix

```

45     <form id="managefix-form" action="/managefix" method="post">
46       <div class="row">
47         <div class="col-lg-6 col-sm-12 mb-2">
48           <label for="id" class="form-label fw-bold">เลือกห้อง</label>
49           <select name="id" id="id" class="form-control" onchange="repairSelectChange(event)" required>
50             <option value="">-</option>
51             <% data.forEach((item) => { %>
52               <% if (item.status < 2) { %>
53                 <option value="<%=item.id %>"><%=item.room_id %> - <%=item.type %></option>
54               <% } %>
55             <% }); %>
56           </select>
57
58         </div>
59         <div class="col-lg-6 col-sm-12 mb-2">
60           <label for="cost" class="form-label fw-bold">ค่าใช้จ่ายเพิ่มเติม</label>
61           <input type="number" class="form-control" id="cost" name="cost" value="0" required>
62         </div>
63       </div>
64       <div class="row mb-2">
65         <div class="col-lg-6 col-sm-12">
66           <label for="description" class="form-label fw-bold">รายละเอียด</label>
67           <input type="text" class="form-control mb-2" id="description" name="description" readonly>
68         </div>
69         <div class="col-lg-6 col-sm-12 mb-3">
70           <div class="col-12 text-nowrap">
71             <p class="fw-bold mt-1 mb-2">เปลี่ยนสถานะ</p>
72             <div class="form-check form-check-inline">
73               <input class="form-check-input" type="radio" name="fixStatus" id="inlineRadio0" value="0">
74               <label class="form-check-label" for="inlineRadio0">แจ้งเรื่อง</label>
75             </div>
76             <div class="form-check form-check-inline">
77               <input class="form-check-input" type="radio" name="fixStatus" id="inlineRadio1" value="1" required>
78               <label class="form-check-label" for="inlineRadio1">กำลังดำเนินการ</label>
79             </div>
80             <div class="form-check form-check-inline">
81               <input class="form-check-input" type="radio" name="fixStatus" id="inlineRadio2" value="2">
82               <label class="form-check-label" for="inlineRadio2">เสร็จสิ้น</label>
83             </div>
84           </div>
85         </div>
86       </div>
87       <button type="submit" class="btn btn-success form-control py-3">ยืนยันการเปลี่ยนสถานะ</button>
88     </form>

```

ใช้ <select> เลือกคำขอเจ้งซ่อมที่ต้องการอัปเดต ใส่ค่าใช้จ่ายเพิ่มเติม (ถ้ามี) เลือกสถานะใหม่ (เจ้งเรื่อง, กำลังดำเนินการ, เสร็จสิ้น) เมื่อกดปุ่ม submit จะส่งข้อมูลไปยัง /managefix โดยใช้ POST request ประวัติการซ่อมที่เสร็จสิ้น

พังก์ชันหลักได้แก่ :

- แสดงรายการแจ้งซ่อมที่เสร็จสิ้น (status = 2)
- มีข้อมูลเพิ่มเติม เช่น ค่าใช้จ่ายเพิ่มเติม

```

91 <div class="bg-white p-3 mt-4 rounded shadowing">
92   <h5 class="title">ประวัติการซ่อมที่เสร็จสิ้น</h5>
93   <hr>
94   <table class="table py-2 text-center" id="no-more-tables">
95     <thead class="table-dark">
96       <tr>
97         <th>ห้องพัก</th>
98         <th>ประเภทงาน</th>
99         <th>รายละเอียด</th>
100        <th>วันที่แจ้งซ่อม</th>
101        <th>ค่าใช้จ่ายเพิ่มเติม</th>
102        <th>สถานะ</th>
103      </tr>
104    </thead>
105    <tbody>
106      <% data3.forEach((item)=>{ %>
107        <tr>
108          <td data-title="ห้องพัก"><%=item.room_id %></td>
109          <td data-title="ประเภทงาน"><%=item.type %></td>
110          <td data-title="รายละเอียด"><%=item.detail %></td>
111          <td data-title="วันที่แจ้งซ่อม"><%=item.request_date %></td>
112          <td data-title="ค่าใช้จ่ายเพิ่มเติม"><%=item.cost %></td>
113          <td data-title="สถานะ">
114            <% if (item.status === 0) { %>
115              แจ้งเรื่อง
116            <% } else if (item.status === 1) { %>
117              กำลังดำเนินการ
118            <% } else { %>
119              เสร็จสิ้น
120            <% } %>
121          </td>
122        </tr>
123      <% }); %>
124    </tbody>
125  </table>
126 </div>
```

## หน้าแจ้งเตือนพัสดุของแอดมิน(/parcelpage)

```

1 <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
2 <script src="https://cdn.jsdelivr.net/npm/chartjs-plugin-datalabels"></script>
3 <script src="https://cdn.jsdelivr.net/npm/chartjs-plugin-zoom"></script>
4

```

- โหลด Chart.js สำหรับสร้างกราฟ
- โหลด chartjs-plugin-datalabels เพื่อแสดงตัวเลขบนแทบของกราฟ
- โหลด chartjs-plugin-zoom เพื่อให้สามารถซูมหรือเลื่อนกราฟได้

### ส่วนหัวของหน้าเว็บ

แสดงหัวข้อ "แจ้งเตือนพัสดุ" และคำอธิบายสั้น ๆ เกี่ยวกับการแจ้งเตือนพัสดุ

```

5 <div class="d-flex justify-content-between align-items-center">
6   <div class="">
7     <h4 class="title">แจ้งเตือนพัสดุ</h4>
8     <p>จัดการ <span class="text-danger">แจ้งเตือนพัสดุ</span></p>
9   </div>
10 </div>

```

### กราฟแสดงจำนวนพัสดุ

ใช้ <canvas> เพื่อแสดงกราฟที่มี ID เป็น parcelChart อยู่ภายใน <div> ที่มีการจัดรูปแบบเพื่อให้แสดงผลสวยงาม

```

12 <div class="row">
13   <div class="col-lg-6 col-sm-12 d-flex justify-content-center align-items-center">
14     <div class="bg-white p-3 mt-3 rounded flex-fill shadowing">
15       <h5 class="title">จำนวนพัสดุ</h5>
16       <hr>
17       <div class="d-flex justify-content-center align-items-center">
18         <canvas id="parcelChart"></canvas>
19       </div>
20     </div>
21   </div>

```

### ฟอร์มแจ้งเตือนพัสดุ

- เลือกห้องพัก

ใช้ <select> เพื่อให้ผู้ใช้เลือกหมายเลขห้องพักข้อมูลเดิมจากตัวแปร room ซึ่งเป็นอาร์เรย์ที่ได้จากเซิร์ฟเวอร์

```

<div class="col-lg-6 col-sm-12 d-flex align-items-stretch">
  <div class="bg-white p-3 mt-3 rounded flex-fill shadowing">
    <h5 class="title">แจ้งเตือน - พัสดุ</h5>
    <hr>
    <form action="/notifyparcel" method="post">
      <div class="row">
        <div class="col-12">
          <label for="room_id" class="form-label fw-bold">ເລືອກຫ້ອງພັກ</label>
          <select name="room_id" id="room_id" class="form-select" required>
            <option value=""></option>
            <% room.forEach((item)=>{ %>
              <option value="<%=item.id%>"><%=item.id%></option>
            <%}); %>
          </select>
        </div>
      </div>
    </form>
  </div>

```

- กรอกชื่อผู้รับ

ใช้ <input type="text"> เพื่อให้ผู้ใช้ระบุชื่อผู้รับพัสดุ

```

38     <div class="row mt-2">
39         <div class="col-12">
40             <label for="receiver_name" class="form-label fw-bold">ชื่อผู้รับ</label>
41             <input type="text" name="receiver_name" id="receiver_name" class="form-input form-control" required>
42         </div>
43     </div>
```

- เลือกขนาดพัสดุ

ใช้ <input type="radio"> เพื่อให้ผู้ใช้เลือกขนาดพัสดุระหว่าง S (เล็ก), M (กลาง), L (ใหญ่)

- ปุ่มส่งข้อมูลแจ้งเตือนพัสดุ

ใช้ <button> ภายใน <form> เพื่อส่งข้อมูลไปยัง /notifyparcel

```

44     <div class="row mt-2">
45         <div class="col-12 textnowrap">
46             <p class="fw-bold mb-2">ขนาดพัสดุ</p>
47             <div class="form-check form-check-inline">
48                 <input class="form-check-input" type="radio" name="size" id="inlineRadio1"
49                 value="S">
50                 <label class="form-check-label" for="inlineRadio1">ขนาดเล็ก</label>
51             </div>
52             <div class="form-check form-check-inline">
53                 <input class="form-check-input" type="radio" name="size" id="inlineRadio2"
54                 value="M" required>
55                 <label class="form-check-label" for="inlineRadio2">ขนาดกลาง</label>
56             </div>
57             <div class="form-check form-check-inline">
58                 <input class="form-check-input" type="radio" name="size" id="inlineRadio3"
59                 value="L">
60                 <label class="form-check-label" for="inlineRadio3">ขนาดใหญ่</label>
61             </div>
62         </div>
63         <div class="mt-3">
64             <button type="submit" class="btn btn-primary form-control form-button py-3">แจ้งเตือนพัสดุ</button>
65         </div>
66     </div>
67 </form>
68 </div>
69 </div>
70 </div>
```

## ตารางแสดงรายการพัสดุทั้งหมด

ใช้ <table> แสดงข้อมูลพัสดุที่ได้รับและยังไม่ได้รับ ดึงข้อมูลจากตัวแปร data ที่เป็นอาร์เรย์

```

72 <div class="bg-white p-3 mt-4 rounded shadowed">
73   <h5 class="title">รายการพัสดุทั้งหมด</h5>
74   <hr>
75   <table class="table parcel-table" id="no-more-tables">
76     <thead>
77       <tr>
78         <th>ห้องพัก</th>
79         <th>ชื่อผู้รับ</th>
80         <th>ขนาด</th>
81         <th>วันที่พัสดุถึง</th>
82         <th>สถานะ</th>
83     </tr>
84   </thead>
85   <tbody>
86     <% data.forEach((item)=>{ %>
87       <tr>
88         <td data-title="ห้องพัก"><%=item.room_id%></td>
89         <td data-title="ชื่อผู้รับ"><%=item.receiver_name%></td>
90         <td data-title="ขนาด">
91           <% if (item.size === 'S') { %>
92             ขนาดเล็ก
93           <% } else if (item.size === 'M') { %>
94             ขนาดกลาง
95           <% } else { %>
96             ขนาดใหญ่
97           <% } %>
98         </td>
99         <td data-title="วันที่พัสดุถึง"><%=item.arrival_date%></td>
100        <td data-title="สถานะ"><%=item.status == 0 ? 'ยังไม่รับ' : 'รับแล้ว'%></td>
101      </tr>
102    <% }); %>
103  </tbody>
104 </table>
105 </div>
```

## สคริปต์สร้างกราฟ Chart.js

ดึงข้อมูลพัสดุจาก /count-parcels และแสดงเป็นกราฟแท่งใช้ Chart.js ร่วมกับ ChartDataLabels และ ChartZoom เพื่อให้สามารถดูข้อมูลแบบโต้ตอบได้

```

190   .then(response => response.json())
191   .then(data => {
192     if (!data || data.length === 0) {
193       console.log("no parcel");
194       return;
195     }
196   }
197
198   const labels = data.map(row => row.arrival_day);
199   const receivedCounts = data.map(row => row.received_count);
200   const notReceivedCounts = data.map(row => row.not_received_count);
201
202
203   const chartData = {
204     labels: labels,
205     datasets: [
206       {
207         label: 'พัสดุที่รับแล้ว',
208         data: receivedCounts,
209         backgroundColor: '#A8E6CF',
210         borderColor: 'darkgreen',
211         borderWidth: 2
212       },
213       {
214         label: 'พัสดุที่ยังไม่รับ',
215         data: notReceivedCounts,
216         backgroundColor: '#FFB3B3',
217         borderColor: 'darkred',
218         borderWidth: 2
219       }
220     ],
221     type: 'bar',
222     tooltipsCallback: function(tooltipItem) {
223       return tooltipItem.dataset.label + ': ' + tooltipItem.raw;
224     }
225   );
226   createChart('parcelChart', chartData);
227 }
228 .catch(error => {
229   console.error('Error:', error);
230 });
231 </script>
```

## หน้าจัดการห้องพัก(/manageroom)

หน้านี้มีหน้าที่แสดงจำนวนห้องว่างและห้องที่ถูกจองแล้วในรูปแบบกราฟ และยังสามารถเพิ่มข้อมูลห้องพักใหม่ได้ผ่านฟอร์มที่กำหนด ระบบนี้ช่วยให้ผู้ดูแลสามารถจัดการห้องพักได้สะดวกยิ่งขึ้น โค้ดนี้ได้มีการใช้ Chart.js และ Chart.js Plugin Datalabels เพื่อใช้สร้างและแสดงผลกราฟข้อมูลห้องพัก

```

4   <div class="d-flex justify-content-between align-items-center">
5     <div>
6       <h4 class="title">ห้องพัก</h4>
7       <p>จำนวน <span class="text-danger">ห้องพัก</span></p>
8     </div>
9   </div>
10

```

ส่วนนี้เป็นหัวข้อของหน้าจอ ซึ่งแสดงข้อความ "ห้องพัก" และคำอธิบายเพิ่มเติมเกี่ยวกับการจัดการห้องพัก การแสดงจำนวนห้องว่าง

```

11 <div class="row">
12   <div class="col-lg-6 col-sm-12 d-flex justify-content-center align-items-center">
13     <div class="bg-white p-3 rounded flex-fill shadow">
14       <h5 class="title">จำนวนห้องว่าง</h5>
15       <hr>
16       <div class="d-flex justify-content-center align-items-center" style="width: 17rem; height: 17rem; justify-self: center;">
17         <canvas id="roomChart"></canvas>
18       </div>
19     </div>
20   </div>

```

ส่วนนี้ใช้แสดงกราฟ Pie Chart ที่แสดงจำนวนห้องที่ว่างและไม่ว่าง โดยใช้ <canvas> ที่มี id="roomChart" เพื่อให้ JavaScript สามารถใช้ Chart.js วาดกราฟได้ ฟอร์มเพิ่มห้องพักใหม่

```

23   <h5 class="title">เพิ่ม - ห้องพัก</h5>
24   <hr>
25   <form action="/addroom" method="post">
26     <div class="row">
27       <div class="col-6">
28         <label>ห้องพัก</label>

```

ส่วนนี้เป็นฟอร์มสำหรับเพิ่มข้อมูลห้องพักใหม่ โดยมีฟิลด์ดังนี้:

- เลขที่ห้องพัก (noroom)
- ค่าเช่า / เดือน (roomprice)
- รายละเอียดเพิ่มเติม (roominfo)
- ปุ่มส่งฟอร์มเพื่อเพิ่มข้อมูลห้องพักไปยังเซิร์ฟเวอร์ผ่าน HTTP POST

## ตารางแสดงข้อมูลห้องพัก

```

50 <div class="bg-white p-3 mt-4 rounded shadow">
51   <table class="table py-2 manage-room-table" id="no-more-tables">
52     <thead class="align-middle">
53       <tr>
54         <th>#</th>
55         <th>ລາຍການ</th>
56         <th>ຫຼາຍ</th>
57         <th>ລາຄາ</th>
58         <th>ລາຍລະອຽດ</th>
59         <th class="text-center">ສະຖານະ</th>
60       </tr>
61     </thead>
62     <tbody class="align-middle">
63       <% let index = 1; %>
64       <% let available = 0; %>
65       <% let notava = 0; %>
66       <% data.forEach(item)>[ %>
67         <tr>
68           <td data-title="#"><%= index++ %></td>
69           <td data-title="ລາຍການ"><% item.id %></td>
70           <td data-title="ຫຼາຍ"><% item.rent %></td>
71           <td data-title="ລາຍລະອຽດ"><% item.description %></td>
72           <td data-title="ສະຖານະ"><% if (<item.status == 0>) { %>
73             <% 'ໃນ' %>
74             <% available++ %>
75           <% } else { %>
76             <% 'ບ່າງ' %>
77             <% notava++ %>
78           <% } %></td>
79           <td data-title="ຮາຄາ" class="text-center">
80             <div class="btn-table">
81               <a href="/editroom?id=<% item.id %>" class="btn btn-warning btn-sm px-lg-4 px-md-3 mx-1" href="#">ແກ້ໄຂ</a>
82               <form id="removeRoom" action="/manageroom/delete/<% item.id %>" method="post" onsubmit="return confirmDelete('removeRoom')">
83                 <button type="submit" class="btn btn-danger btn-sm px-lg-4 px-md-3 mx-1" href="#">ລົບ</button>
84               </form>
85             </div>
86           </td>
87         </tr>
88       <% }); %>
89     </tbody>
90   </table>
91 </div>

```

ตารางนี้ใช้แสดงรายการห้องพักทั้งหมด โดยมีคอลัมน์ดังนี้:

- เลขที่ห้องพัก
- ราคา
- รายละเอียด
- สถานะ (ว่าง หรือ ไม่ว่าง)
- ปั๊มจัดการ (แก้ไข/ลบ)

แต่ละ列ของตารางถูกสร้างขึ้นโดยใช้โค้ด EJS (<% ... %>) ซึ่งช่วยให้สามารถแสดงข้อมูลที่รับมาจากเชิร์ฟเวอร์ได้แบบไดนามิก

การคำนวณจำนวนห้องว่างและไม่ว่าง

```

const available = <%= available %>;
const notAva = <%= notAva %>;

```

ตัวแปร available และ notAva ใช้เก็บจำนวนห้องที่ว่างและไม่ว่างจากเชิร์ฟเวอร์ เพื่อนำไปใช้สร้างกราฟ

## การสร้างกราฟ Pie Chart

```

97 <div>
98   const roomChartData = {
99     type: 'pie',
100    labels: ['ພະນັກງານ', 'ຫຼອງໃນວ່າງ'],
101    dataValues: [available, notAva],
102    backgroundColors: ['rgba(255, 255, 255, 1)', 'rgba(63, 114, 175, 0.8)'],
103    borderColors: ['rgba(214, 219, 224, 1)', 'rgba(44, 89, 138, 1)],
104    label: 'ຈຳນວນພະນັກງານ',
105    tooltipsCallback: (tooltipItem) => `${tooltipItem.label}: ${tooltipItem.raw} ຜອນ`};
106 </div>

```

กำหนดข้อมูลสำหรับสร้างกราฟ Pie Chart:

- ประเภท: pie
- ป้ายกำกับ: "ห้องว่าง" และ "ห้องไม่ว่าง"
- ค่าสำหรับแต่ละหมวดหมู่: available และ notAvailable
- สีเพื่อหลังและลักษณะของกราฟ
- พังก์ชันสำหรับแสดงข้อมูล Tooltip

## พังก์ชันการสร้างกราฟ

```

107 |     function createChart(chartId, chartData) {
108 |       const data = {
109 |         labels: chartData.labels,
110 |         datasets: [
111 |           {
112 |             label: chartData.label,
113 |             data: chartData.dataValues,
114 |             backgroundColor: chartData.backgroundColors,
115 |             borderColor: chartData.borderColor,
116 |             borderWidth: 2
117 |           }
118 |         ];
119 |       const ctx = document.getElementById(chartId).getContext('2d');
120 |       new Chart(ctx, {
121 |         type: chartData.type,
122 |         data: data,
123 |         options: {
124 |           responsive: true,
125 |           maintainAspectRatio: true,
126 |           plugins: {
127 |             legend: {
128 |               display: true,
129 |               position: 'bottom',
130 |               labels: {
131 |                 font: {
132 |                   size: 14,
133 |                   weight: 'bold',
134 |                 },
135 |               },
136 |             },
137 |             datalabels: {
138 |               color: '#000',
139 |               font: {
140 |                 weight: 'bold',
141 |                 size: 14
142 |               },
143 |               align: 'center',
144 |               formatter: (value) => `${value}`,
145 |             }
146 |           },
147 |           tooltips: {
148 |             enabled: true,
149 |             callbacks: {
150 |               label: chartData.tooltipsCallback,
151 |             }
152 |           },
153 |         }
154 |       });
155 |     }
156 |
157 |   
```

พังก์ชันนี้ใช้สร้างกราฟโดยกำหนด **labels**, **datasets**, และ **options** จาก **chartData**

## เรียกใช้พังก์ชันสร้างกราฟ

```

161 |   // สร้างกราฟเมื่อโหลดหน้าเริ่ม
162 |   ...createChart('roomChart', roomChartData);
163 | 
```

คำสั่งนี้จะเรียกใช้พังก์ชัน **createChart()** และแสดงกราฟ Pie Chart ใน **<canvas id="roomChart">**

## หน้าจัดการผู้ใช้ (/manageuser)

หน้านี้จะช่วยให้ผู้ดูแลสามารถเพิ่ม, แก้ไข และลบข้อมูลผู้ใช้ได้อย่างสะดวก โดยใช้เทมเพลต EJS เพื่อレンเดอร์ข้อมูลที่ได้รับจากฐานข้อมูล ระบบนี้ออกแบบมาให้รองรับทั้งผู้ดูแลและผู้ใช้ โดยแสดงข้อมูลในรูปแบบตาราง พร้อมปุ่มควบคุมต่างๆ ที่ช่วยให้การจัดการข้อมูลทำได้ง่ายขึ้น

### ส่วนหัวของหน้า (Header Section)

```

1 <div class="d-flex justify-content-between align-items-center">
2   <div class="">
3     <h4 class="title">ผู้ใช้งาน & ผู้เช่า</h4>
4     <p>สถานะ <span class="text-danger">ปิด</span></p>
5   </div>
6   <div class="">
7     <a href="/adduser" class="btn btn-primary text-light btn-sm-sm btn-lg form-button"><img xmlns="http://www.w3.org/2000/svg" height="26px" viewBox="0 -960 960 960" width="26px" fill="#e8eaed">
8       <path d="M784-400v-128h600v-80h120v-120h80v120h120v80h120v1
9         20h-80z-36h-88q-66 0-113-47l-113q-66 47-113t113-47q66 0 113-
10        47t47 113q0 66-47 113t113 47ZM48-160v-112q0-34 17.5-62.5t104-378
11        62-31 126-46.5t360-448q6 0 138 15.5t616-378q29 15 46.5
12        43.5t680-272v12h40zW88-80h480v-32q0-11.5-5.5-20t580-30q5-54-27-
13        109-40.5t360-368q-56 0-111 13.5t140-306q-9 5-14.5 14t-5.5 20v32Zm240
14        -320q33 0 56.5-23.5t440-640q0-33-23.5-56.5t360-72q0-33 0-56.5 23.5t280-64
15        8q0 33 23.5 56.5t360-560zW88-80Zm0 400z"/></img> เพิ่มผู้ใช้งาน</a>
16   </div>
17 </div>
18

```

- ส่วนนี้เป็นหัวข้อหลักของหน้าเว็บ ซึ่งแสดงชื่อ "ผู้ใช้งาน & ผู้เช่า" และมีปุ่มสำหรับเพิ่มผู้ใช้งานใหม่
- ปุ่ม "เพิ่มผู้ใช้งาน" มีลิงก์ไปที่หน้า /adduser และใช้ไอคอน SVG เพื่อแสดงสัญลักษณ์การเพิ่ม (+)

### ตารางแสดงรายการผู้ใช้งาน

```

19 <div class="bg-white p-3 mt-3 rounded shadowing">
20   <table class="table py-2 manage-user-table" id="no-more-tables">
21     <thead class="align-middle">
22       <tr>
23         <th>#</th>
24         <th>ชื่อ</th>
25         <th>นาม</th>
26         <th>นามพ</th>
27         <th class="text-center">จัดการ</th>
28       </tr>
29     </thead>
30     <tbody>
31       <% let index = 1; %>
32       <% data.forEach((item)=>{ %>
33         <tr>
34           <td data-title="#"><%=index++%></td>
35           <td data-title="ชื่อ"><%=item.fullname%></td>
36           <td data-title="นาม">
37             <% if (item.sex === 'M') { %>
38               ชาย
39             <% } else if (item.sex === 'F') { %>
40               หญิง
41             <% } else if (item.sex === 'O') { %>
42               อื่นๆ
43             <% } %>
44           </td>
45           <td data-title="นามพ"><%=item.role === 1 ? 'ผู้ดูแล' : 'ผู้เช่า'%></td>
46           <td data-title="จัดการ" class="text-center">
47             <div class="btn-table">
48               <button class="btn btn-success btn-sm px-lg-4 px-md-3 mx-1" onclick="showDetails('<%= item.id %>')>แสดง</button>
49               <a href="/edituser?id=<%=item.id%>" class="btn btn-warning btn-sm px-lg-4 px-md-3 mx-1">แก้ไข</a>
50               <form id="removeUser" action="/manageuser/delete/<%=item.id%>" method="post" onsubmit="return confirmDelete('removeUser')">
51                 <button type="submit" class="btn btn-danger btn-sm px-lg-4 px-md-3 mx-1">ลบ</button>
52               </form>
53             </div>
54           </td>
55         </tr>
56       <% }; %>
57     </tbody>
58   </table>
59 </div>

```

ตารางนี้ใช้แสดงข้อมูลผู้ใช้งาน โดยรับค่าจากตัวแปร data ซึ่งเป็นอาร์เรย์ของข้อมูลผู้ใช้มีการใช้ EJS <% ... %> เพื่อวนลูป (forEach) และแสดงค่าต่างๆ

คอลัมน์ต่างๆ ประกอบด้วย:

- # - เลขลำดับของผู้ใช้
- ชื่อ - ชื่อเต็มของผู้ใช้
- เพศ - แสดง "ชาย", "หญิง" หรือ "อื่นๆ" ตามค่าที่กำหนด
- บทบาท - แสดง "ผู้ดูแล" หาก role === 1 มีฉะนั้นจะแสดง "ผู้ใช้"
- จัดการ - มีปุ่มสำหรับ แสดง, แก้ไข และลบผู้ใช้

#### ปุ่มการจัดการผู้ใช้งาน

- ปุ่ม "แสดง" (showDetails(id)) ใช้เรียกดูข้อมูลของผู้ใช้งาน
- ปุ่ม "แก้ไข" (/edituser?id=<%= item.id %>) นำผู้ใช้ไปยังหน้าแก้ไขข้อมูล
- ปุ่ม "ลบ" (/manageuser/delete/<%= item.id %>) ใช้สำหรับลบข้อมูล โดยมี onsubmit="return confirmDelete('removeUser')" เพื่อยืนยันก่อนการลบ

## หน้าจองห้องพัก(/bookroom)

โค้ดในส่วนนี้ช่วยให้ผู้ใช้สามารถจองห้องพักได้โดย:

1. เลือกห้องที่ต้องการจอง
2. เลือกผู้เช่าที่จะเข้าพัก
3. กำหนดวันที่เข้าและย้ายออก (ถ้ามี)
4. กดปุ่ม "ยืนยันการจอง" เพื่อบันทึกข้อมูล

นอกจากนี้ยังมี ตารางแสดงรายการห้องที่ว่าง เพื่อให้ผู้ดูแลสามารถตรวจสอบสถานะของห้องพักได้สะดวกขึ้น

### ฟอร์มจองห้องพัก

โค้ดส่วนแรกเป็น ฟอร์มสำหรับการจองห้องพัก ซึ่งอยู่ภายใต้ `<div class="bg-white p-4 mt-3 rounded shadowing">` เพื่อให้มีการจัดรูปแบบที่สวยงามและอ่านง่าย

- ฟิลด์การเลือกห้องและผู้เช่า

```

1 <div class="bg-white p-4 mt-3 rounded shadowing">
2   <h5 class="title">จองห้องพัก</h5>
3   <hr>
4   <form id="adduser-form" action="/bookroom-submit" method="post">
5     <div class="row">
6       <div class="col-6">
7         <label for="selectroom" class="form-label fw-bold">เลือกห้อง</label>
8         <select name="selectroom" id="selectroom" class="form-control" required>
9           <option value=""></option>
10          <% data.forEach((item)=>{ %>
11            <option value="<%=item.id%>"><%=item.id%></option>
12          <% }); %>
13        </select>

```

- `<select>` ใช้สำหรับเลือกห้องที่ต้องการจอง
- `<% data.forEach((item)=>{ %> ... <% }); %>` เป็นโค้ดของ EJS ที่ใช้นำลุปเพื่อดึงรายการห้องจากตัวแปร `data` และเพิ่มเป็นตัวเลือก (`<option>`) ให้กับ dropdown

ในลักษณะเดียวกัน ส่วนที่เลือกผู้เช่าก็ทำงานในแบบเดียวกัน โดยใช้ `user.forEach()` แทน `data.forEach()`

- ฟิลด์กำหนดวันที่เข้าพักและย้ายออก

```

24   </div>
25   <div class="row mt-2">
26     <div class="col-6">
27       <label for="movein" class="form-label fw-bold">วันที่เข้ามา</label>
28       <input type="date" class="form-control" id="movein" name="movein" required>
29     </div>
30     <div class="col-6">
31       <label for="checkout" class="form-label fw-bold">วันที่ย้ายออก</label>
32       <input type="date" class="form-control" id="checkout" name="checkout">
33     </div>
34   </div>
35   <div class="row mt-4">
36     <div class="col-12">
37       <button type="submit" class="btn btn-success form-control py-3" id="bookroom-submit">ยืนยันการจอง</button>
38     </div>
39   </div>
40 </form>
41 </div>

```

- <input type="date"> ใช้ให้ผู้ใช้สามารถเลือกวันที่เข้าพัก
- ฟิลด์ checkout (วันที่ย้ายออก) สามารถเว้นว่างไว้ได้ หากยังไม่ระบุวันออก

ปุ่มยืนยันการจอง ปุ่มนี้ใช้สำหรับส่งข้อมูลการจองไปยังเซิร์ฟเวอร์ผ่าน HTTP POST ไปยังเส้นทาง /bookroom-submit

#### ตารางแสดงรายการห้องที่ว่าง

โค้ดส่วนนี้ใช้ EJS ใน การแสดงรายการห้องว่าง โดยมีการวนลูปผ่านตัวแปร data เพื่อนำข้อมูลห้องมาแสดงใน <table>

```

43 <div class="bg-white p-4 mt-4 rounded shadowing">
44   <h5 class="title">รายการห้องที่ว่าง</h5>
45   <hr>
46   <table class="table py-2 book-room-table" id="no-more-tables">
47     <thead>
48       <tr>
49         <th>#</th>
50         <th>ห้องพัก</th>
51         <th>ราคา</th>
52         <th>รายละเอียด</th>
53         <th>สถานะ</th>
54       </tr>
55     </thead>
56   <tbody>

```

- <table> ใช้สำหรับแสดงรายการห้องพักในรูปแบบของตาราง
- <thead> กำหนดส่วนหัวของตารางที่มีคอลัมน์ต่างๆ ได้แก่ ลำดับ (#), หมายเลขห้อง (ห้องพัก), ค่าเช่า (ราคา), รายละเอียด (รายละเอียด), และสถานะ (สถานะ)

## วนลูปแสดงรายการห้อง

```

57      <% index = 1; %>
58      <% data.forEach((item)=>{ %>
59      <tr>
60          <td data-title="#"><%=index++%></td>
61          <td data-title="พื้นที่"><%=item.id%></td>
62          <td data-title="ราคา"><%=item.rent%></td>
63          <td data-title="รายละเอียด"><%=item.description%></td>
64          <td data-title="สถานะ"><%=item.status === 0 ? 'ว่าง' : 'ไม่ว่าง'%></td>
65      </tr>
66
67  </tbody>
68 <%}); %>
69 </table>
70 </div>

```

- data.forEach((item)=>{ ... }) ใช้วนลูปเพื่อนำข้อมูลแต่ละรายการมาแสดงเป็นแถว (<tr>) ในตาราง
- <%= item.id %> และ <%= item.rent %> ใช้แสดงหมายเลขห้องและค่าเช่า
- <%= item.status === 0 ? 'ว่าง' : 'ไม่ว่าง' %> ใช้ตรวจสอบสถานะของห้อง หากค่า status เป็น 0 จะแสดง ว่าง มิฉะนั้นจะแสดง ไม่ว่าง

## หน้าจัดการมิเตอร์ไฟฟ้า/น้ำ (/managemeter)

หน้าบันทึกมิเตอร์ไฟฟ้า/น้ำ และการกำหนดราคาก่าไฟฟ้า/น้ำสำหรับห้องพักในระบบ โดยโค้ดนี้ใช้เทคโนโลยี HTML ผสมกับ EJS (Embedded JavaScript) เพื่อแสดงข้อมูลและรับอินพุตจากผู้ใช้

โค้ดนี้ช่วยให้ผู้ดูแลระบบสามารถ:

1. บันทึกค่ามิเตอร์ไฟฟ้าและน้ำสำหรับแต่ละห้อง
2. ตั้งค่าราคาค่าไฟฟ้าและค่าน้ำต่อหน่วย
3. ดึงข้อมูลที่เกี่ยวข้องจากฐานข้อมูลและแสดงผลแบบไดนามิกผ่าน EJS

โดยโค้ดนี้รองรับการทำงานบนเว็บแอปพลิเคชันที่ใช้ Node.js และ Express เป็นเชิร์ฟเวอร์หลัก

### 1. บันทึกมิเตอร์ไฟฟ้า/น้ำ

โค้ดนี้ใช้สำหรับบันทึกค่ามิเตอร์ไฟฟ้าและน้ำของแต่ละห้อง โดยให้ผู้ใช้เลือกห้องที่ต้องการบันทึก จากนั้นกรอกค่ามิเตอร์ไฟฟ้าและค่าน้ำล่าสุด

```

4   <form id="editmeter-form" action="/editmeter" method="post">
5     <div class="row mb-2">
6       <div class="col-12">
7         <label for="selectroom" class="form-label fw-bold">เลือกห้อง</label>
8         <select name="selectroom" id="selectroom" class="form-control" onchange="meterSelectChange(event)" required>
9           <option value="">-</option>
10          <% room.forEach(item) => { %>
11            <option value="<%= item.id %>"><%= item.id %></option>
12          <% };>%<
13        </select>
14      </div>
15    </div>

```

ฟอร์มนี้จะส่งข้อมูลไปยังเส้นทาง /editmeter ผ่านเมธอด POST เพื่ออัปเดตค่ามิเตอร์ของห้องที่เลือก  
เลือกห้องพัก

- ดึงข้อมูลห้องทั้งหมดจากตัวแปร room
- เมื่อเลือกห้อง ระบบจะเรียกใช้ฟังก์ชัน meterSelectChange(event) เพื่อโหลดค่ามิเตอร์ล่าสุดของห้องนั้น ๆ

```

16     <div class="row mb-3">
17       <div class="col-lg-6 col-sm-12 mb-2">
18         <label for="electricmeter" class="form-label text-warning fw-bold">มิเตอร์ค่าไฟฟ้า</label>
19         <input type="number" class="form-control mb-2" id="electricmeter" name="electricmeter" required>
20         <p class="p-0 m-0 text-danger fw-bold electric-date">วันที่บันทึกมิเตอร์ไฟฟ้าล่าสุด : (วันที่)</p>
21         <p class="p-0 m-0 text-danger fw-bold electric-prev">เลขมิเตอร์ก่อนบันทึก : (เลข)</p>
22       </div>
23       <div class="col-lg-6 col-sm-12 mb-2">
24         <label for="watermeter" class="form-label text-info fw-bold">มิเตอร์ค่าน้ำ</label>
25         <input type="number" class="form-control mb-2" id="watermeter" name="watermeter" required>
26         <p class="p-0 m-0 text-danger fw-bold water-date">วันที่บันทึกมิเตอร์น้ำล่าสุด : (วันที่)</p>
27         <p class="p-0 m-0 text-danger fw-bold water-prev">เลขมิเตอร์ก่อนบันทึก : (เลข)</p>
28       </div>
29     </div>
30     <button type="submit" class="btn btn-success form-control py-3">ยืนยันการบันทึก</button>
31   </form>
32 </div>

```

อินพุตค่ามิเตอร์ไฟฟ้า ให้ผู้ใช้กรอกค่ามิเตอร์ไฟฟ้าล่าสุดของห้องที่เลือกข้อมูลจะถูกบันทึกในฐานข้อมูลเมื่อตอบส่งฟอร์ม อินพุตค่ามิเตอร์น้ำ ใช้บันทึกค่ามิเตอร์น้ำล่าสุดของห้อง

### แสดงค่ามิเตอร์ล่าสุด

แสดงวันที่และค่ามิเตอร์ก่อนหน้าของไฟฟ้าและน้ำ (อัปเดตผ่าน JavaScript)

```

29 |     </div>
30 |     <button type="submit" class="btn btn-success form-control py-3">ยืนยันการบันทึก</button>
31 |   </form>
32 | </div>
33 |

```

ปุ่มยืนยันบันทึกมิเตอร์ เมื่อคลิกปุ่ม ระบบจะส่งค่ามิเตอร์ไฟฟ้าและน้ำไปยังเซิร์ฟเวอร์

## 2. กำหนดราคาไฟฟ้า/น้ำ

โค้ดนี้ใช้สำหรับตั้งราคาค่าไฟฟ้าและน้ำต่อหน่วย ซึ่งจะใช้เมื่อกันทุกห้องพัก

```

34 <div class="bg-white p-4 mt-4 rounded shadowing">
35   <h4 class="title">กำหนดราคา - ไฟฟ้า / น้ำ</h4>
36   <p><span class="text-danger">หมายเหตุ : ราคាត่อหน่วยเท่ากันทุกห้อง</span></p>
37   <hr>
38   <form id="editprice-form" action="/editprice" method="post">
39     <div class="row mb-3">
40       <% data.forEach(item)>{ %
41         <div class="col-lg-6 col-sm-12 mb-2">
42           <label for="electricprice" class="form-label text-warning fw-bold">ค่าไฟฟ้า / หน่วย</label>
43           <input type="number" class="form-control mb-2" id="electricprice" name="electricprice" value="<%item.elec_rate%>" required>
44         </div>
45         <div class="col-lg-6 col-sm-12 mb-2">
46           <label for="waterprice" class="form-label text-info fw-bold">ค่าน้ำ / หน่วย</label>
47           <input type="number" class="form-control mb-2" id="waterprice" name="waterprice" value="<%item.water_rate%>" required>
48         </div>
49       <% } %>
50     </div>
51     <button type="submit" class="btn btn-success form-control py-3">ยืนยันการกำหนดราคา</button>
52   </form>
53 </div>
54
55 <script src="selectMeter.js"></script>

```

- ฟอร์มนี้ใช้ส่งข้อมูลไปยัง /editprice ผ่านเมธอด POST เพื่ออัปเดตราคาค่าไฟฟ้าและน้ำ อินพุตค่าไฟฟ้าต่อหน่วย ค่าตั้งต้นของค่าไฟฟ้าต่อหน่วยจะถูกโหลดจากตัวแปร item.elec\_rate อินพุตค่าค่าน้ำต่อหน่วย ค่าน้ำต่อหน่วยจะถูกโหลดจากตัวแปร item.water\_rate ปุ่มยืนยันการกำหนดราคา เมื่อคลิกปุ่ม ระบบจะส่งข้อมูลราคาค่าไฟฟ้าและค่าน้ำไปยังเซิร์ฟเวอร์ การใช้งาน JavaScript โค้ดนี้โหลดไฟล์ selectMeter.js ซึ่งน่าจะมีฟังก์ชัน meterSelectChange(event) สำหรับดึงค่ามิเตอร์ล่าสุดเมื่อผู้ใช้เลือกห้อง

## หน้าใบแจ้งหนี้/ใบเสร็จ

โค้ดนี้เป็นระบบจัดการธุกรรมที่ช่วยให้เจ้าของห้องสามารถสร้างใบแจ้งหนี้และใบเสร็จรับเงินได้สะดวก โดยใช้ EJS ร่วมกับ JavaScript เพื่อให้ระบบทำงานได้แบบดีนามิก โดยมีฟีเจอร์หลักดังนี้:

- เลือกห้องเพื่อสร้างใบแจ้งหนี้
- กำหนดวันออกบิลและวันกำหนดชำระ
- แสดงรายการธุกรรมพร้อมสถานะการชำระเงิน
- อัปเดตข้อมูลโดยอัตโนมัติด้วย JavaScript

### ส่วนหัวของหน้า (Header)

```

1 <div class="d-flex justify-content-between align-items-center">
2   <div class="">
3     <h4 class="title">จัดการธุกรรม</h4>
4     <p>จัดการ <span class="text-danger">ธุกรรม</span></p>
5   </div>
6   <div class="">
7     <h5 class="title text-end" id="time"></h5>
8     <a href="/showreceipt" class="btn btn-primary text-light btn-lg form-button">ใบเสร็จรับเงิน</a>
9     <a href="/showinvoice" class="btn btn-primary text-light btn-lg form-button">ใบแจ้งหนี้</a>
10  </div>
11 </div>
12

```

ใช้ Bootstrap เพื่อจัดโครงสร้างหน้าเว็บให้อยู่ในรูปแบบ Flexbox และปุ่มสำหรับไปยังหน้า ใบเสร็จรับเงิน และ ใบแจ้งหนี้ มี `<h5 id="time">` ที่ใช้สำหรับแสดงวันที่ปัจจุบัน ซึ่งจะถูกอัปเดตโดย JavaScript

### แบบฟอร์มสร้างใบชำระเงิน (Invoice Creation Form)

```

19   <form id="insert-bill-form" method="POST" action="/insertbill">
20     <div class="row">
21       <div class="col-4">
22         <label for="room_id" class="form-label fw-bold">ห้อง</label>
23         <select name="room_id" id="room_id" class="form-select" required>
24           <option value=""></option>
25           <% data.forEach((item) => { %>
26             <option value="<%= item.room_id %>">
27               data-owner="<%= item.owner_name %>">
28               data-rent="<%= item.rent %>">
29               data-meter="<%= item.meter_id %>">
30               data-user="<%= item.user_id %>">
31               data-cost="<% item.maintenance_cost_filtered %>"><%= item.room_id %> (Meter: <%= item.meter_id %>)</option>
32           <% }); %>
33         </select>
34         <input type="hidden" class="form-control" id="user_id" name="user_id" readonly>
35         <input type="hidden" class="form-control" id="meter_id" name="meter_id" readonly>
36         <input type="hidden" class="form-control" id="cost_id" name="cost_id" readonly>
37       </div>
38       <div class="col-4">
39         <label for="owner-name" class="form-label fw-bold">ชื่อเจ้าของห้อง</label>
40         <input type="text" class="form-control" id="owner-name" name="owner-name" readonly>
41       </div>
42       <div class="col-4">
43         <label for="rent" class="form-label fw-bold">ค่าเช่าห้อง</label>
44         <input type="text" class="form-control" id="rent" name="rent" readonly>
45       </div>
46     </div>
47

```

มีฟอร์ม <form> ที่มี method="POST" เพื่อส่งข้อมูลไปที่ /insertbill ใช้ <select> ให้ผู้ใช้เลือกห้องที่ต้องการสร้างใบชำระเงิน พร้อมข้อมูลเพิ่มเติมที่ช่อนอยู่ใน data-attributes เช่น owner\_name, rent, meter\_id, user\_id ข้อมูลใน <select> ถูกดึงมาจาก EJS ที่วนลูป data.forEach()

### ตารางแสดงรายการใบชำระเงิน (Invoice List)

```

68 <div class="bg-white p-3 mt-3 rounded shadowing">
69   <table class="table py-2 invoice-tables" id="no-more-tables">
70     <thead class="align-middle">
71       <tr>
72         <th>#</th>
73         <th>ห้อง</th>
74         <th>นักลงทุน</th>
75         <th>ห้องน้ำ</th>
76         <th>สถานะ</th>
77         <th class="text-center">ดำเนินการ</th>
78       </tr>
79     </thead>
80     <tbody>
81       <tr bills.forEach((item)=>[] %>
82         <td>
83           <td data-title="#"><%=item.id%></td>
84           <td data-title="ห้อง"><%=item.room_id%></td>
85           <td data-title="นักลงทุน">
86             <%=item.created_at%>
87           </td>
88           <td data-title="ห้องน้ำ"><%=item.due_date%></td>
89           <td data-title="สถานะ">
90             <% if (item.status == '2') { %>
91               ชำระแล้ว
92             <% } else { %>
93               ยังไม่ชำระ
94             <% } %>
95           </td>
96         </td>
97         <td data-title="ดำเนินการ" class="text-center">
98           <div class="btn-table">
99             <% if (item.status == '1') { %>
100               <a href="/addrceipt?id=<%= item.room_id %>" class="btn btn-success btn-sm px-lg-4 px-md-3 mx-1">
101                 สร้างใบเสร็จรับเงิน
102               </a>
103             <% } else if (item.status == '2') { %>
104               <a href="/home" class="btn btn-dark btn-sm px-lg-4 px-md-3 mx-1" onclick="return false;">
105                 ชำระแล้ว
106               </a>
107             <% } else { %>
108               <a href="/addinvoice?id=<%= item.room_id %>" class="btn btn-secondary btn-sm px-lg-4 px-md-3 mx-1">
109                 สร้างใบแจ้งหนี้
110               </a>
111               <a href="/addreceipt?id=<%= item.room_id %>" class="btn btn-success btn-sm px-lg-4 px-md-3 mx-1" onclick="return false;">
112                 สร้างใบเสร็จรับเงิน
113               </a>
114             </div>
115           </td>
116         </tr>
117       </tbody>
118     </table>
119   </div>
120 
```

แสดงรายการใบชำระเงินที่ออกให้กับผู้เช่า โดยดึงข้อมูลจาก bills และแสดงเป็นแตรใน <tbody>

คอลัมน์ "สถานะ" จะแสดง ชำระแล้ว หรือ ยังไม่ชำระ ตามค่าของ item.status

มีปุ่มสำหรับสร้างใบแจ้งหนี้และใบเสร็จรับเงิน (เฉพาะใบแจ้งหนี้ที่ยังไม่ได้ชำระเท่านั้นที่สามารถออกใบเสร็จได้)

## JavaScript สำหรับอัปเดตข้อมูลแบบไอนามิก

```

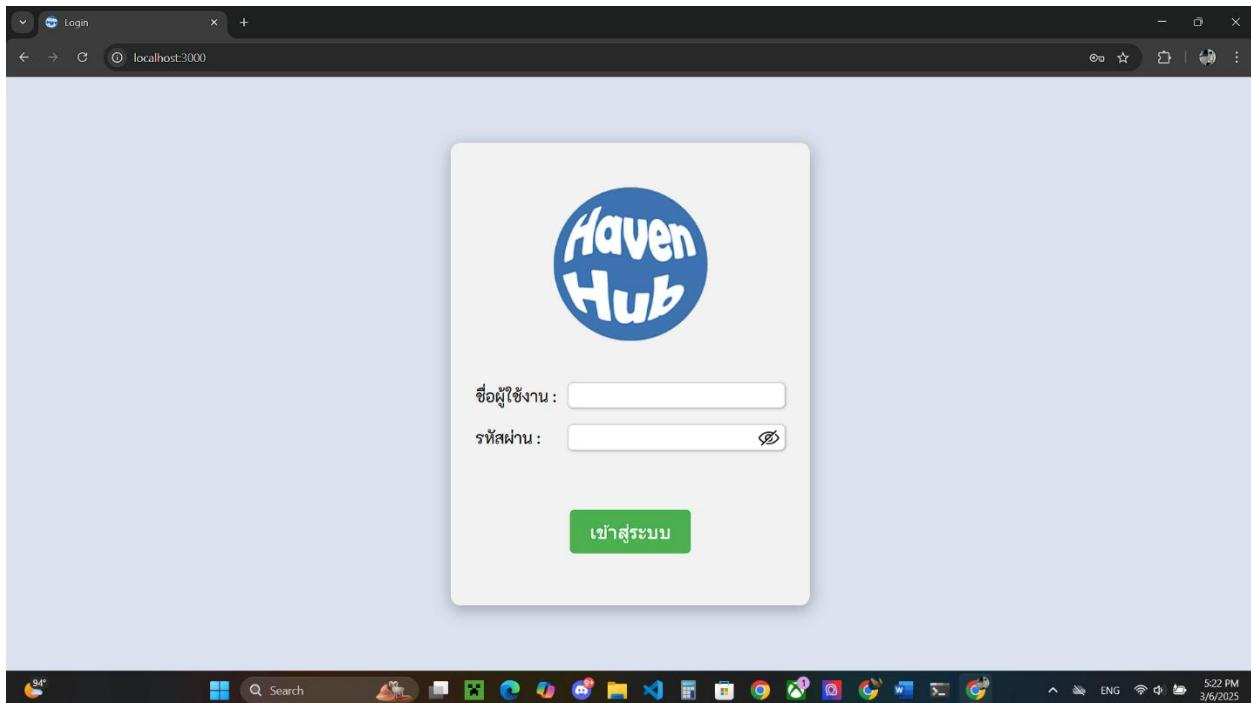
125 <script>
126   ... document.getElementById('room_id').addEventListener('change', function() {
127     ... var selectedOption = this.options[this.selectedIndex];
128     ... var roomId = selectedOption.value;
129     ... var ownerName = selectedOption.getAttribute('data-owner');
130     ... var rent = selectedOption.getAttribute('data-rent');
131     ... var meterId = selectedOption.getAttribute('data-meter');
132     ... var userId = selectedOption.getAttribute('data-user');
133     ... var costId = selectedOption.getAttribute('data-cost');
134     ... document.getElementById('owner-name').value = ownerName;
135     ... document.getElementById('rent').value = rent;
136     ... document.getElementById('meter_id').value = meterId;
137     ... document.getElementById('user_id').value = userId;
138     ... document.getElementById('cost_id').value = costId;
139   });
140   function updateDate() {
141     const currentDate = new Date();
142     const day = String(currentDate.getDate()).padStart(2, '0');
143     const month = String(currentDate.getMonth() + 1).padStart(2, '0'); // เดือนจะเริ่มต้นที่ 0
144     const year = currentDate.getFullYear();
145     const formattedDate = `${day}/${month}/${year}`;
146
147     document.getElementById('time').textContent = formattedDate;
148   }
149
150 setInterval(updateDate, 1000);
151
152 updateDate();
153
154
155
156   ... document.getElementById('createDay').value = new Date().toISOString().split('T')[0];
157 </script>
158

```

เมื่อผู้ใช้เลือกห้อง ข้อมูลเจ้าของห้อง ค่าเช่า และหมายเลขมิเตอร์จะถูกดึงมาแสดงโดยใช้ JavaScript และ data-attributes ฟังก์ชัน updateDate() ใช้ JavaScript Date API เพื่อแสดงวันที่ปัจจุบันใน <h5 id="time"> และอัปเดตอัตโนมัติทุกวินาที

## บทที่ 4 ผลการดำเนินงาน

หน้าเข้าสู่ระบบผู้ใช้งาน



## หน้าหลักผู้เช่า

The screenshot shows the HavenHub mobile application interface. At the top, there is a header bar with the logo and navigation icons. Below the header, the main content area displays the user's profile information: "ยินดีต้อนรับ, Testor". The profile includes fields for "ห้อง": "A101", "พื้นที่": "โถง", and "พื้นที่": "0 ตรม.". On the left side, there is a sidebar menu titled "MENU" with options: "หน้าแรก" (selected), "บันทึกเมล็ดไฟ/ว่า", "ส่งเรื่องแจ้งเตือน", and "แจ้งเตือนพัสดุ". Below the sidebar is a "PROFILE" section with "ออกจากระบบ". The central part of the screen features a chart titled "ค่าใช้จ่ายแต่ละเดือน" (Monthly Expenses) showing a series of blue bars representing expenses over time.

เดือน	ค่าใช้จ่าย
Jan	5,200
Feb	5,100
Mar	5,500
Apr	5,300
May	5,400
Jun	5,100
Jul	5,600
Aug	5,300
Sep	5,200

## หน้าตรวจสอบบันทึก มิเตอร์

**ตรวจสอบบันทึกมิเตอร์**

ตรวจสอบบันทึก **ไฟฟ้า**

**บันทึกมิเตอร์ไฟฟ้า & น้ำ**

เมื่อวันที่  น้ำเสีย

เมื่อวันที่  น้ำดื่ม

วันที่บันทึกมิเตอร์ไฟฟ้าล่าสุด : (วันที่)

วันที่บันทึกมิเตอร์น้ำล่าสุด : (วันที่)

**ราคา - ไฟฟ้า / น้ำ**

หมายเหตุ : ราคาค่าไฟน้ำอย่างไรก็ตามที่ลูกค้าต้องการ

ค่าไฟฟ้า / หน่วย	ค่าน้ำ / หน่วย
7	4

## หน้าส่งเรื่องแจ้งซ่อม

The screenshot shows the HavenHub website interface. On the left, there's a sidebar with a dark blue header 'HavenHub' and a menu bar below it. The menu items include 'MENU' (with 'หน้าแรก'), 'บ้านที่ก็ไม่เคยไฟ/น้ำ' (selected), 'ส่งเรื่องแจ้งซ่อม' (selected), 'แจ้งเตือนพัสดุ' (with a red notification dot), and 'PROFILE' (with 'ออกจากระบบ'). Below the menu is a URL 'localhost:3001/repair'.

The main content area has a title 'ส่งเรื่องแจ้งซ่อม' (Report) and a sub-section 'แจ้งการแจ้งซ่อม'. It contains two input fields: 'เลือกเดลท์ที่ต้องการแจ้ง' (Select the delta to be reported) and 'ประเภทงาน' (Type of work). Below these is a text area labeled 'รายละเอียด' (Details) with placeholder text 'กรุณากรอกรายละเอียด' (Please enter details).

A large green button at the bottom says 'บันทึกการแจ้งซ่อม' (Save report). Below this is a section titled 'ประวัติการแจ้งซ่อม' (History of reports) with a table header:

ลำดับ	ผู้แจ้ง	ประเภทงาน	รายละเอียด	วันที่แจ้งซ่อม	ผู้ใช้งานที่รับแจ้ง	สถานะ
-------	---------	-----------	------------	----------------	---------------------	-------

The table body is currently empty and displays the message 'ไม่มีข้อมูล' (No data).

## หน้าแจ้งเตือนพัสดุ

The screenshot shows the HavenHub application interface. At the top, there is a header bar with the logo and navigation links. Below the header, the main content area displays a delivery notification for package A101.

**แจ้งเตือนพัสดุ - ของห้อง A101**

ห้องนี้ แจ้งเตือนพัสดุ

**พัสดุที่กำลังรอรับ**

ผลลัพธ์	ชื่อผู้รับ	หมายเหตุพัสดุ	วันที่ส่งเดินทาง	บันทึกการรับพัสดุ
A101	tester	ขนาดใหญ่	2025-03-08 14:49:02	<b>บันทึกการรับ</b>

ผลลัพธ์ 1 จาก 1 ผลลัพธ์

< 1 >

**ประวัติการรับพัสดุ**

ห้องพัก	ชื่อผู้รับ	หมายเหตุพัสดุ	วันที่พัสดุถึง	วันที่บันทึกการรับพัสดุ	สถานะ

## หน้าแดร็กของเจ้าของหอพัก

The screenshot displays the HavenHub Dashboard interface. On the left, there is a vertical sidebar with a user icon, the text "HavenHub", and a menu bar with three sections: "MENU", "MANAGE", and "PROFILE".

The main content area is titled "Dashboard" and includes the following sections:

- จำนวนห้องพัก**: A pie chart showing room status. Legend: ห้องว่าง (White) and ห้องไม่ว่าง (Blue). Data: 7 (ห้องว่าง), 1 (ห้องไม่ว่าง).
- ผู้เช่า**: A pie chart showing tenant status. Legend: ชาย (Blue), หญิง (Pink), อื่นๆ (Yellow). Data: 4 (ชาย), 1 (หญิง), 0 (อื่นๆ).
- ห้องที่ต้องออกใบแจ้งหนี้ (2025-03-08)**: A horizontal bar chart showing outstanding bills. Legend: ห้อง (Black bar). Total value: 0.
- ห้องที่ยังไม่จ่าย**: A table listing rooms that have not paid. Columns: ห้อง (Room ID), ค่าเช่ารายเดือน (Monthly Rent), วันที่ต้องชำระ (Due Date). Data:
 

ห้อง	ค่าเช่ารายเดือน	วันที่ต้องชำระ
A101	5568	2025-03-06
A102	5288	2025-03-01
- ห้องที่มียอดเข้าวันนี้ (2025-03-08)**: A placeholder text indicating today's incoming payments.
- ห้องที่มียอดออกวันนี้ (2025-03-08)**: A placeholder text indicating today's outgoing payments.
- ห้องที่เกินกำหนดชำระ**: A placeholder text indicating rooms past due.

## หน้าคำขอแจ้งซ่อม

The screenshot shows the HavenHub application interface with the following details:

- Left Sidebar (MENU):**
  - หน้าแรก (Home)
  - คำขอแจ้งซ่อม (Report Maintenance)** (highlighted in blue)
  - แจ้งเดือนพัสดุ (Report Supplies)
  - จัดการห้องพัก (Room Management)
  - จัดการผู้เช่า (Tenant Management)
  - จองห้องพัก (Book Room)
  - จัดการมีเดรียฟ/บ้าน (Manage Medrif/House)
  - ใบแจ้งหนี้/ใบเสร็จ (Bill Statement)
  - ออกจากระบน (Logout)
- Top Right:** User profile icon with a red notification dot.
- Main Content Area:**
  - Section 1: คำขอแจ้งซ่อม** (Report Maintenance Request)
 

ผลลัพธ์	ประเภทงาน	รายละเอียด	วันที่แจ้งซ่อม	สถานะ
A101	อุกเดิน	ห้องน้ำแตก	2025-03-08 14:51:13	แจ้งเรื่อง
  - Section 2: ดำเนินการซ่อม** (Repair Work)
 

เลือกห้อง	ค่าใช้จ่ายเพิ่มเติม
-	0
รายละเอียด	<input type="radio"/> แจ้งเรื่อง <input type="radio"/> ก้าวลงสู่การดำเนินการ <input type="radio"/> เสิร์ฟิล์ม
ป้อนข้อมูลเพิ่มเติม	
  - Section 3: ประวัติการซ่อมที่เสร็จสิ้น** (Completed Maintenance History)
 

ผลลัพธ์	ประเภทงาน	รายละเอียด	วันที่แจ้งซ่อม	ค่าใช้จ่ายเพิ่มเติม	สถานะ
---------	-----------	------------	----------------	---------------------	-------

## หน้าแจ้งเตือนพัสดุ

The screenshot displays the HavenHub application interface. On the left, there is a vertical sidebar with a dark blue header containing the logo and the word "HavenHub". Below the header, the sidebar has three main sections: "MENU", "MANAGE", and "PROFILE". Under "MENU", there are links for "หน้าแรก" (Home), "คำขอเชิงช่อง", and "แจ้งเตือนพัสดุ" (selected). Under "MANAGE", there are links for "จัดการฟ้องทัก", "จัดการผู้ใช้", "จดหมาย", and "จัดการมีเดียไฟ/บีบ". Under "PROFILE", there is a link for "ออกจากระบบ". The main content area is titled "แจ้งเตือนพัสดุ" and contains several sections:

- จำนวนพัสดุ**: A chart showing one package received (green bar) and zero packages delivered (red bar). The chart is dated 2025-03-08.
- แจ้งเตือน - พัสดุ**: A form for reporting a package. It includes fields for "เลขที่ฟ้องพัสดุ" (package tracking number), "ชื่อผู้รับ" (recipient name), "ชนบทพัสดุ" (package type), and three radio button options: "ขนาดเด็ก" (child), "ขนาดกลาง" (middle), and "ขนาดใหญ่" (large). A blue button at the bottom right says "แจ้งเตือนพัสดุ".
- รายการพัสดุห้องน้ำ**: A table listing packages. The columns are: แมสต์ (Master), ห้องพัสดุ (Storage Unit), ชื่อผู้รับ (Recipient Name), สถานะ (Status), วันที่ห้องน้ำ (Date), สถานะ (Status), ห้องน้ำ (Water Tank), รับแล้ว (Received), and ลบ (Delete). One row is shown: "A101", "tester", "ขนาดใหญ่", "2025-03-08 14:49:02", "สถานะ", "รับแล้ว".

## หน้าจัดการห้องพัก

**เพิ่ม - ห้องพัก**

เลขที่ห้องพัก	ผู้เช่า / เสื้อ
รายละเอียดเพิ่มเติม	
<b>เพิ่มห้องพัก</b>	

**แก้ไข - ห้องพัก**

เลขที่ห้องพัก	ผู้เช่า / เสื้อ
รายละเอียดเพิ่มเติม	
<b>แก้ไขห้องพัก</b>	<b>ยกเลิกการแก้ไข</b>

**HavenHub**

**ห้องพัก**

จำนวนห้องว่าง

จำนวนห้องทั้งหมด: 8 ห้อง

จำนวนห้องว่าง: 7 ห้อง

จำนวนห้องเช่า: 1 ห้อง

กรุณากรอกข้อมูลการเข้าออก  
วันที่เข้าออก  
mm/dd/yyyy

หมายเหตุ

ห้องพัก

ค่าว่า / เดือน

ห้องพัก

เพิ่มห้องพัก

จัดการห้องพัก

จัดการผู้เช่า

จองห้องพัก

จัดการไม่มีเช่าให้/น้ำ

ใบแจ้งหนี้/ใบเสร็จ

ออกจากระบบ

ผลลัพธ์ 10 ผลลัพธ์

#	ห้องพัก	ราคา	รายละเอียด	สถานะ	ผู้เช่า	จัดการ
1	A101	5000		ไม่ว่าง	วีโรจน์ ลักษณอติศร	<button>แก้ไข</button> <button>ข้อมูล</button> <button>ลบ</button>
2	A102	5000		ว่าง	ก้าลากานุ ใจเด็ด	<button>แก้ไข</button> <button>ข้อมูล</button> <button>ลบ</button>
3	A105	5000		ว่าง	-	<button>แก้ไข</button> <button>ข้อมูล</button> <button>ลบ</button>
4	A106	5500	มีแอร์	ว่าง	-	<button>แก้ไข</button> <button>ข้อมูล</button> <button>ลบ</button>
5	A107	5500	มีแอร์	ว่าง	-	<button>แก้ไข</button> <button>ข้อมูล</button> <button>ลบ</button>
31	A401	5500	มีแอร์	มีว่าง	วีโรจน์ ลักษณอติศร	<button>แก้ไข</button> <button>ข้อมูล</button> <button>ลบ</button>
32	A402	5500	มีแอร์	มีว่าง	ก้าลากานุ ใจเด็ด	<button>แก้ไข</button> <button>ข้อมูล</button> <button>ลบ</button>
33	A403	5500	มีแอร์	ว่าง	-	<button>แก้ไข</button> <button>ข้อมูล</button> <button>ลบ</button>
34	A404	5500	มีแอร์	ว่าง	-	<button>แก้ไข</button> <button>ข้อมูล</button> <button>ลบ</button>
35	A405	5500	มีแอร์	ว่าง	-	<button>แก้ไข</button> <button>ข้อมูล</button> <button>ลบ</button>
36	A406	5500	มีแอร์	ไม่ว่าง	สมปอง นาดี	<button>แก้ไข</button> <button>ข้อมูล</button> <button>ลบ</button>
37	A407	5500	มีแอร์	ไม่ว่าง	สุชาร์ว สุวรรณลักษ์	<button>แก้ไข</button> <button>ข้อมูล</button> <button>ลบ</button>
38	A408	5500	มีแอร์	ว่าง	-	<button>แก้ไข</button> <button>ข้อมูล</button> <button>ลบ</button>
39	A409	5500	มีแอร์	ไม่ว่าง	มงคล อมขาวด	<button>แก้ไข</button> <button>ข้อมูล</button> <button>ลบ</button>
40	A410	5500	มีแอร์	ว่าง	-	<button>แก้ไข</button> <button>ข้อมูล</button> <button>ลบ</button>

ยืนยันการลบ

คุณต้องการลบห้องนี้ใช่หรือไม่?

A403

ยกเลิก ยืนยัน

## หน้าจัดการผู้เช่า

 HavenHub ☰ Logout 

ຜູ້ໃນຈຳນາ & ຜູ້ຂ່າ

ຈົດກາງ ຜູ້ໃນຈຳນາ & ຜູ້ຂ່າ

ຂໍເພີ້ມໄປໃຈວ່າ

ແລສອ 10 ແລວ

ຕົວທາງ:

#	ຊື່	ເພດ	ນາມສະກຸນ	ຈົດກາງ
1	Admin 01	ຍາຍ	ເມື່ອແລ	<span>ແລສອ</span> <span>ກັບໄປ</span> <span>ລົບ</span>
2	Testor 01	ຍາຍ	ຜູ້ຂ່າ	<span>ແລສອ</span> <span>ກັບໄປ</span> <span>ລົບ</span>
3	ຈິຈາກ ພິມເພົມບັນ	ຍາຍ	ຜູ້ຂ່າ	<span>ແລສອ</span> <span>ກັບໄປ</span> <span>ລົບ</span>
4	ວິທີສາ ຮອດຮຽນ	ໜີ້ງ	ຜູ້ຂ່າ	<span>ແລສອ</span> <span>ກັບໄປ</span> <span>ລົບ</span>
5	ນິກູ້ຍາ ອິນ	ຍາຍ	ຜູ້ຂ່າ	<span>ແລສອ</span> <span>ກັບໄປ</span> <span>ລົບ</span>

ແລສອ 1 ອື່ນ 5 ຈາກ 5 ແລວ

< 1 >

MENU

ໜ້ານເຮັດ

ຄ່າຍອເງື່ອງຂອນ

ແພີ້ມເຕັມພື້ນຖານ

MANAGE

ຈົດກາງໂຄງຫຼັກ

ຈົດກາງຜູ້ຂ່າ

ຈອນໜ່ອງທຶກ

ຈົດການນິຄອນໄຟ/ນໍາ

ໃນແຈ້ງທີ່/ໃນເສົ້າ

PROFILE

ອອກຈາກຮັບນັບ

HavenHub

ผู้ใช้งาน & ผู้เช่า

เพิ่ม - ผู้ใช้งาน & ผู้เช่า

ชื่อจริง	นามสกุล		
Testor			
เพศ	หมายเลขอปภ.ประจำบ้าน	ผลวินิจฉัย	รหัสผ่าน
ชาย	0001	test	....

**เพิ่มผู้ใช้งาน** **ยกเลิก**

HavenHub

ผู้ใช้งาน & ผู้เช่า

แก้ไข - ผู้ใช้งาน & ผู้เช่า

ชื่อจริง	นามสกุล		
Testor	01		
เพศ	หมายเลขอปภ.ประจำบ้าน	ผลวินิจฉัย	รหัสผ่าน
ชาย	0001	test	....

**แก้ไขผู้ใช้งาน** **ยกเลิก**

The screenshot shows a web-based application named "HavenHub". The left sidebar contains a navigation menu with sections like "MENU", "MANAGE", and "PROFILE". Under "MANAGE", there are several options: "ผู้ใช้งาน & ผู้ช่วย" (User & Assistant), "จัดการ ผู้ใช้งาน & ผู้ช่วย" (Manage User & Assistant), "จัดการผู้ช่วย" (Manage Assistant), "จัดการห้องพัก" (Manage Room), "จัดการมีดอาร์ท/น้า" (Manage Art/Knife), and "ใบแจ้งหนี้/ใบเสร็จ" (Bill/Bill). Under "PROFILE", there is an option "ออกจากระบบ" (Logout).

The main content area is titled "ผู้ใช้งาน & ผู้ช่วย" (User & Assistant) and displays a table of users. The table has columns: ลำดับ (Rank), ชื่อ (Name), เพศ (Gender), ประเภท (Type), จัดการ (Manage), and a search bar at the top right.

ลำดับ	ชื่อ	เพศ	ประเภท	จัดการ
1	Admin 01	ชาย	ผู้ดูแล	<button>ลบทิ้ง</button> <button>ยกเลิก</button> <button>ลบ</button>
2	Testor 01	ชาย	ผู้ช่วย	<button>ลบทิ้ง</button> <button>ยกเลิก</button> <button>ลบ</button>
3	จิรา บันหมอกบ้าน	ชาย	ผู้ช่วย	<button>ลบทิ้ง</button> <button>ยกเลิก</button> <button>ลบ</button>
4	วงศารา พลธรรมรัตน์	หญิง	ผู้ช่วย	<button>ลบทิ้ง</button> <button>ยกเลิก</button> <button>ลบ</button>
5	ปัญญา ลือแย	ชาย	ผู้ช่วย	<button>ลบทิ้ง</button> <button>ยกเลิก</button> <button>ลบ</button>

A modal dialog box is open over the fifth row, containing the text "ยืนยันการลบ" (Confirm Deletion) and "ต้องการลบผู้ใช้งานนี้ใช่หรือไม่?" (Do you want to delete this user?). It includes two buttons: "ยกเลิก" (Cancel) and "ยืนยัน" (Confirm).

At the bottom of the table, it says "แสดง 1 ถึง 5 จาก 5 ผล" (Displaying 1 to 5 of 5 results) and has navigation buttons for "ก่อน" (Previous), "1" (Page 1), and "ถัดไป" (Next).

## หน้าจอของห้องพัก

The screenshot shows the HavenHub application interface. On the left, there is a sidebar with a dark blue header containing the logo and the word "HavenHub". Below the header, the sidebar has three main sections: "MENU", "MANAGE", and "PROFILE". Under "MENU", the "จองห้องพัก" (Book Room) option is highlighted with a blue background. Under "MANAGE", the "จัดการห้องพัก" (Manage Room) option is also highlighted. Under "PROFILE", the "ออกจากระบນ" (Logout) option is shown. The main content area has a light gray header with the title "จองห้องพัก" (Book Room). Below the header, there are two input fields: "เลือกห้อง" (Select Room) and "เลือกผู้เช่า" (Select Tenant). Underneath these fields are two date inputs: "วันที่เข้ามาบ้าน" (Check-in Date) and "วันที่ออกจากบ้าน" (Check-out Date), both in "mm/dd/yyyy" format. A green button labeled "ยืนยันการจอง" (Confirm Booking) is located at the bottom of this section. Below this, another section titled "รายการห้องที่ว่าง" (List of Available Rooms) is displayed. It includes a dropdown menu for "แสดง" (Show) with options "10" and "50", and a search bar labeled "ค้นหา". A table lists seven available rooms, each with columns for room number (#), room number (ห้องพัก), price (ราคา), status (สถานะ), and remarks (รายละเอียด). All rooms listed have a status of "ว่าง" (Available).

#	ห้องพัก	ราคา	สถานะ	รายละเอียด
1	A102	5000	ว่าง	
2	A105	5000	ว่าง	
3	A106	5500	มีเมร์	
4	A107	5500	มีเมร์	
5	A108	5500	มีเมร์	
6	A109	5500	มีเมร์	
7	A110	5500	มีเมร์	

## หน้าบันทึกมิเตอร์น้ำ/ไฟ

The screenshot shows the HavenHub application interface. The left sidebar contains a navigation menu with the following items:

- MENU**
  - หน้าแรก
  - สำหรับเจ้าของบ้าน
  - แจ้งเตือนพื้นที่ดูแล
- MANAGE**
  - จัดการฟ้องทัก
  - จัดการผู้เช่า
  - จองห้องที่
  - จัดการน้ำเตือนไฟ/น้ำ** (highlighted in blue)
  - ใบแจ้งหนี้/ใบเสร็จ
- PROFILE**
  - ออกจากระบบ

The main content area displays two forms:

**บันทึก - มิเตอร์ไฟฟ้า / น้ำ**

This form includes fields for selecting a meter (เลือกเมเตอร์) and entering meter readings (บันทึกค่าไฟฟ้า). It also contains notes about water usage (จำนวนน้ำที่ใช้) and electricity usage (จำนวนไฟฟ้าที่ใช้).

**กำหนดราคา - ไฟฟ้า / น้ำ**

This form allows setting electricity rates (กำหนดราคาไฟฟ้า) and water rates (กำหนดราคาน้ำ). It includes fields for setting rates per unit (ต่อหน่วย) and total units (หน่วยรวม).

## หน้าไปแจ้งหนี้/ใบเสร็จ

The screenshot shows the HavenHub application interface. At the top, there is a dark blue header bar with the logo 'HavenHub' and a user icon. To the right of the header, the date '08/03/2025' is displayed, along with two buttons: 'ไม่เสร็จรับเงิน' (Not Paid) and 'ใบแจ้งหนี้' (Bill/Invoice).

The main content area has a title 'จัดการธุรกรรม' (Manage Transactions) and a sub-section 'จัดการ ฐานราก'. Below this, there is a form titled 'สร้างใบชำระเงิน' (Create Bill/Invoice). The form includes fields for 'รหัส' (Code), 'วันเดือนปี' (Date), 'วันที่กำหนดชำระ' (Due Date), and 'จำนวนเงิน' (Amount). A dropdown menu for 'วันเดือนปี' is open, showing '03/08/2025'.

Below the form is a table titled 'ผลลัพธ์' (Results) with three rows of data:

#	รหัส	วันเดือนปี	วันที่กำหนดชำระ	สถานะ	ดำเนินการ
3	A101	2025-03-08		ยังไม่ชำระ	<button>สร้างใบแจ้งหนี้</button> <button>สร้างใบเสร็จรับเงิน</button>
2	A102	2025-03-01	2025-03-03	ยังไม่ชำระ	<button>สร้างใบแจ้งหนี้</button>
1	A101	2025-03-06	2025-03-07	มีจำนวน	<button>สร้างใบเสร็จรับเงิน</button>

At the bottom of the table, it says 'ผลลัพธ์ 1 จาก 3 จาก 3 ผล' (Results 1 from 3 from 3 results). There are navigation buttons '<' and '>' at the bottom right of the table.

**HavenHub**

สร้างใบแจ้งหนี้

← ย้อนกลับ

มีเดวาร์ฟพ้ากวน	มีเดวาร์ฟพ้าเงิน	ใบไฟฟ้าไป	ราคา/หน่วย	ราคารวม
2025-03-08	2025-03-06 15:07:42	80	7	560

มีเดวาร์ฟพ้ากวน	มีเดวาร์ฟพ้าเงิน	ใบไฟฟ้าไป	ราคา/หน่วย	ราคารวม
2025-03-08	2025-03-06 15:07:42	2	4	8

ค่าใช้จ่ายเพิ่มเติม

รายการ	รายการ
ค่าเช่า	5000
รายการ	ราคา

5568 ยืนยันการคำนวณ

**HavenHub**

ใบแจ้งหนี้

จัดการ ใบแจ้งหนี้

← ย้อนกลับ

ผลลัพธ์ 10 ผล							ค้นหา:
#	ห้อง	วันออกบิล	วันกำหนดชำระ	สถานะ	ดำเนินการ	ดำเนินการ	
3	A101	2025-03-08		ชำระแล้ว	<span style="background-color: green; color: white; padding: 2px;">ใบแจ้งหนี้</span>	<span style="background-color: green; color: white; padding: 2px;">ใบแจ้งหนี้</span>	
2	A102	2025-03-01	2025-03-03	ชำระแล้ว	<span style="background-color: green; color: white; padding: 2px;">ใบแจ้งหนี้</span>	<span style="background-color: green; color: white; padding: 2px;">ใบแจ้งหนี้</span>	
1	A101	2025-03-06	2025-03-07	ชำระแล้ว	<span style="background-color: green; color: white; padding: 2px;">ใบแจ้งหนี้</span>	<span style="background-color: green; color: white; padding: 2px;">ใบแจ้งหนี้</span>	

ผลลัพธ์ 1 ถึง 3 จาก 3 ผล



**HavenHub**

MENU

- หน้าแรก
- รายการเชิงช่อง
- แจ้งเตือนพัสดุ

MANAGE

- จัดการเพื่อหัก
- จัดการผู้เช่า
- จดหมายหัก
- จัดการมีเดตไฟฟ้า
- ใบแจ้งหนี้/ใบเสร็จ

PROFILE

- ออกจากระบน

ใบแจ้งหนี้

ชื่อ-นามสกุล:	Testor 01	วันเดือนปี:	2025-03-08
ที่อยู่:	A101	จำนวนหน้า:	

รายการ	ราคาต่อห้อง	จำนวนห้อง	รวมรวม
ค่าไฟ	7	80	560
ค่าน้ำ	4	2	8
ค่าเช่า	5000	1	5000

รวมทั้งสิ้น: **5568 บาท**

ขอบคุณที่ใช้บริการ - HavenHub

[ดาวน์โหลดเป็น PDF](#)
[ล้างหน้า](#)



**HavenHub**

MENU

- 👤 หน้าร้าน
- 🔥 คำขอเชิงข้อมูล
- 🖨️ แจ้งเตือนพัสดุ
- 📝 จัดการของทั้งหมด
- 👤 จัดการผู้ใช้งาน
- 🖨️ รายงานพื้นที่/บันทึก
- 💻 ใบเรื่องที่/ใบเสร็จ

PROFILE

- 👤 ออกรายงาน

### ใบเรื่องที่/ใบเสร็จ

รายการ ใบเรื่องที่/ใบเสร็จ

ลำดับ	ห้อง	วันเดือนปี	วันที่หานักช่าง	สถานะ	ดำเนินการ
3	A101	2025-03-08		ชำรุด	<span>ใบเรื่องที่ที่ปรับปรุง</span>

ผลลัพธ์ 1 ถึง 1 จาก 1 ผล

< 1 >

 ใบเรื่องที่/ใบเสร็จ

ชื่อ-นามสกุล:	Testor 01	วันเดือนปี:	2025-03-08
ห้อง:	A101	วันที่หานักช่าง:	

รายการ	จำนวนหน่วย	จำนวนเงิน	หมายเหตุ
ค่าไฟ	7	80	560
ค่าน้ำ	4	2	8
ค่าเช่า	5000	1	5000

รวมทั้งสิ้น: **5568 บาท**

ขอบคุณที่ใช้บริการ - HavenHub

 ดาวน์โหลดเป็น PDF
< ย้อนกลับ

## บทที่ 5 គ្រឿងការងារ

### សំរាប់តែខោងអេដក

ហ្មាន់ខ្សោរបប

នៃហ្មាន់របស់ពីរាជរដ្ឋប្រជុំហ្មាន់ខ្សោរបប



មើលខ្សោរបបសំរើរ

ជាប្រភពខ្សោរកដោយរាយការណ៍រាជការ

អេដក	តាមថ្ងៃទី
A102	2025-03-01
A103	2025-03-08

## เมนูการใช้งาน

เมนูการใช้งานจะอยู่ทางซ้ายมือของหน้าเว็บโดยจะประกอบไปด้วย

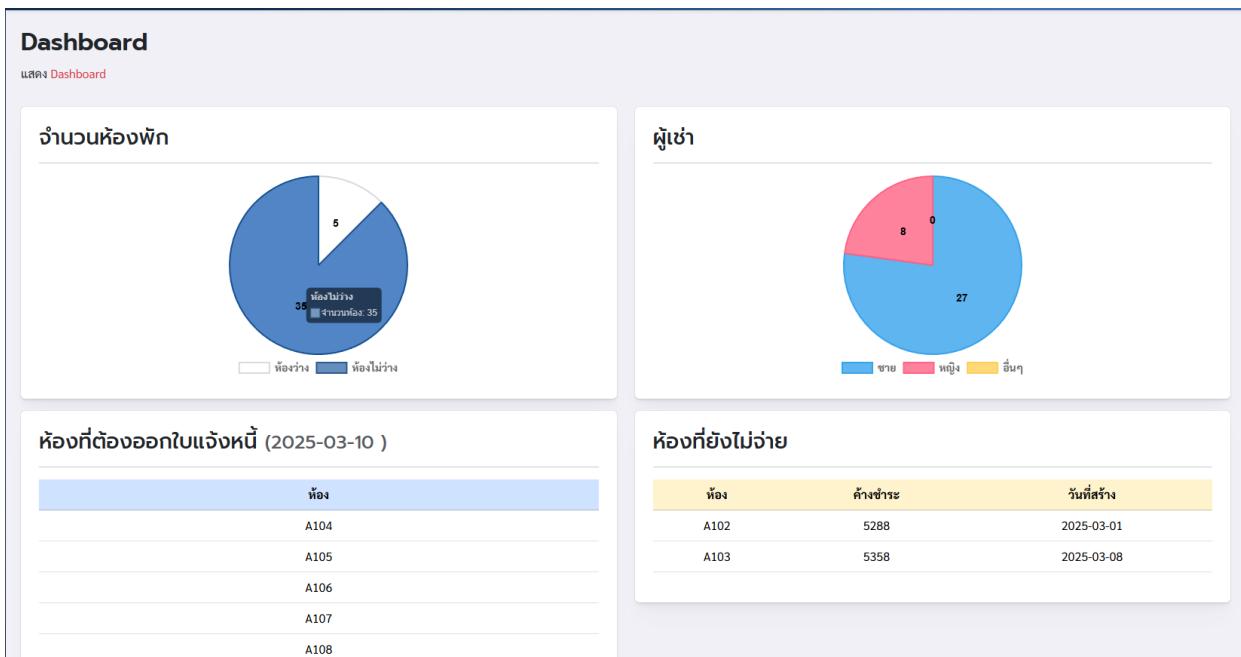
- หน้าแรก หน้าที่แสดงสรุปข้อมูลต่างๆ เช่น จำนวนห้องพัก ผู้เช่า ห้องที่ต้องออกใบแจ้งหนี้วันนี้ เป็นต้น
- คำขอแจ้งซ่อม หน้าที่แสดงให้เห็นว่าห้องใดมีการแจ้งซ่อมเข้ามาและเราสามารถเข้าไปรายงานความคืบหน้าของการซ่อมให้ผู้พักได้จากหน้านี้
- จัดการห้องพัก หน้าที่สามารถจัดการห้องพักได้ เช่น การเพิ่ม ลบ แก้ไขรายละเอียดห้องพักหรือการย้ายผู้เช่าออกจากห้องพัก
- จัดการผู้เช่า หน้าที่สามารถจัดการผู้เช่าได้ เช่น การเพิ่ม ลบ แก้ไขรายละเอียดของผู้เช่าได้
- จัดการมิเตอร์ไฟ/น้ำ หน้าที่สามารถจดบันทึกแก้ไขราคาต่อหน่วยของมิเตอร์น้ำและไฟฟ้าได้ในแต่ละห้อง
- ใบแจ้งหนี้/ใบเสร็จ หน้าที่สามารถออกใบแจ้งหนี้ ใบเสร็จให้ผู้เช่าได้

The screenshot shows the HavenHub dashboard interface. On the left, there is a vertical sidebar menu with sections: MENU (Home page, Report, Log out), MANAGE (Manage rooms, Manage tenants, Manage bills, Manage meter), and PROFILE (Logout). The main area is titled 'Dashboard' and contains four cards:

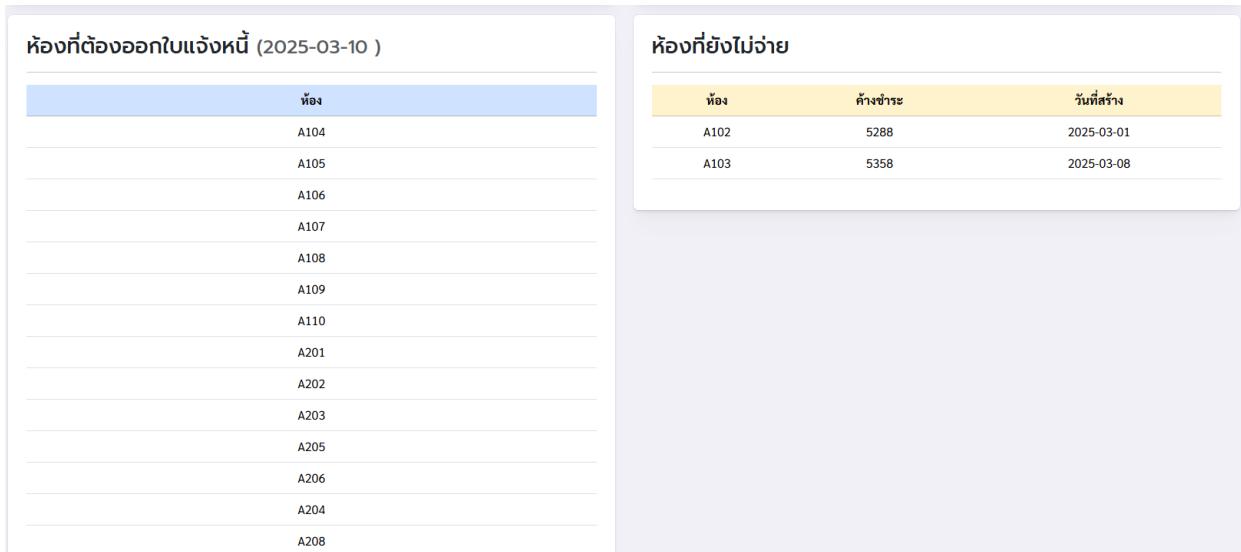
- จำนวนห้องพัก**: A pie chart showing room types: 36 rooms are 'ห้องว่าง' (Empty room) and 5 are 'ห้องในว่าง' (Occupied room).
- ผู้เช่า**: A pie chart showing tenant types: 27 are 'ชาย' (Male), 8 are 'หญิง' (Female), and 6 are 'อื่นๆ' (Others).
- ห้องที่ต้องออกใบแจ้งหนี้ (2025-03-10)**: A table listing rooms due for bills: A104, A105, A106, A107, and A108.
- ห้องที่ยังไม่จ่าย**: A table listing unpaid rooms: A102 (欠款 5288, วันที่ชำระ 2025-03-01) and A103 (欠款 5358, วันที่ชำระ 2025-03-08).

## หน้าแรก

เมื่อกดเข้าหน้าแรกจะพบกับหน้าตั้งภาพข้างล่างนี้



ซึ่งจากภาพจะเห็นว่ามีการแสดงแผนภูมิวงกลมของจำนวนห้องพักและผู้เช่า และเมื่อลองมาจะพบกับ การแสดงรายละเอียดห้องที่ห้องออกใบแจ้งหนี้ ณ วันที่เข้าใช้งานอยู่และห้องที่ยังไม่ชำระค่าเช่าพัก



เมื่อเลื่อนลงมาอีก ก็จะพบกับการแสดงรายการของห้องที่จะบัญชีเข้าและบัญชีออกในวันที่เข้าใช้งานอยู่ ขณะนั้น และแสดงรายการของห้องที่ไม่ชำระบัญชีในระยะเวลาที่กำหนด

ห้องที่บัญชีเข้าวันนี้ (2025-03-10)		ห้องที่บัญชีออกวันนี้ (2025-03-10)		ห้องที่เก็บกำหนดชำระ			
ห้อง	ชื่อเจ้าของห้อง	ห้อง	ชื่อเจ้าของห้อง	ห้อง	ค้างชำระ	วันที่ค้างชำระ	เลขกำหนด(วัน)
A204	ศรีสุข	A303	ชนกวนิม	A102	5288	2025-03-03	7
A301	ศุภนิธิ			A103	5358	2025-03-09	1

### หน้าคำขอแจ้งซ่อม

หน้านี้จะแสดงรายการแจ้งซ่อมที่แจ้งเข้ามา การแจ้งสถานะการดำเนินงานซ่อมและประวัติการ

### ทำรายการดังภาพ

#### คำขอแจ้งซ่อม

จัดการ ค่าขอแจ้งซ่อม

แสดง 5 ผล	แก้ไข	ค้นหา:		
ห้องพัก	ประเภทงาน	รายละเอียด	วันที่แจ้งซ่อม	สถานะ
ไม่พบข้อมูล				
แสดง 0 ถึง 0 จาก 0 ผล	< >			

#### ดำเนินการซ่อม

เลือกห้อง	ค่าใช้จ่ายเพิ่มเติม
-	0
รายละเอียด	เบร์บันสายด่วน
	<input type="radio"/> แจ้งเรื่อง <input type="radio"/> กำลังดำเนินการ <input type="radio"/> เสร็จสิ้น
<button>บันทึกการเปลี่ยนสถานะ</button>	

#### ประวัติการซ่อมที่เสร็จสิ้น

ห้องพัก	ประเภทงาน	รายละเอียด	วันที่แจ้งซ่อม	ค่าใช้จ่ายเพิ่มเติม	สถานะ
A102	เร่งด่วน	นายยอดฟิกบัว	2025-03-09 19:22:45		แจ้งเรื่อง

เมื่อมีรายการแจ้งซ่อมเข้ามาจะแสดงดังภาพนี้

#### คำขอแจ้งซ่อม

จัดการ ค่าขอแจ้งซ่อม

แสดง 5 ผล	แก้ไข	ค้นหา:		
ห้องพัก	ประเภทงาน	รายละเอียด	วันที่แจ้งซ่อม	สถานะ
A102	เร่งด่วน	นายยอดฟิกบัว	2025-03-09 19:22:45	แจ้งเรื่อง
แสดง 1 ถึง 1 จาก 1 ผล	< 1 >			

หลังจากที่ได้รับรายการแจ้งซ่อมแล้วเราจะมารายงานการดำเนินงานแจ้งซ่อมให้ผู้เช่าได้จากในส่วนดำเนินการซ่อมดังภาพ

**ดำเนินการซ่อม**

เลือกห้อง	ค่าใช้จ่ายเพิ่มเติม
-	0
รายละเอียด	เปลี่ยนสถานะ
	<input type="radio"/> แจ้งเรื่อง <input type="radio"/> กำลังดำเนินการ <input type="radio"/> เสร็จสิ้น
บันทึกการเปลี่ยนสถานะ	

### ขั้นตอนการแจ้งดำเนินการซ่อม

- 1) เลือกห้องที่จะแจ้งการซ่อม

**ดำเนินการซ่อม**

เลือกห้อง
-
A102 - เร่งด่วน

- 2) หากมีค่าใช้จ่ายเพิ่มเติมสามารถใส่ได้ในช่องค่าใช้จ่ายเพิ่มเติม

ค่าใช้จ่ายเพิ่มเติม
0

- 3) เลือกการเปลี่ยนสถานะการซ่อม (กำลังดำเนินการ, เสร็จสิ้น)

**เปลี่ยนสถานะ**

- แจ้งเรื่อง  กำลังดำเนินการ  เสร็จสิ้น

- 4) กดปุ่มยืนยันการเปลี่ยนสถานะ

**ดำเนินการซ่อม**

เลือกห้อง	ค่าใช้จ่ายเพิ่มเติม
A102 - เร่งด่วน	0
รายละเอียด	เปลี่ยนสถานะ
บ้าหมายเหตุเพิ่มเติม	<input checked="" type="radio"/> แจ้งเรื่อง <input type="radio"/> กำลังดำเนินการ <input type="radio"/> เสร็จสิ้น
บันทึกการเปลี่ยนสถานะ	

หากการเปลี่ยนสถานะคือกำลังดำเนินการ รายการแจ้งซ่อมจะยังไม่เสร็จสิ้นและจะอยู่ในรายการคำขอแจ้งซ่อม

**คำขอแจ้งซ่อม**

จัดการ คำขอแจ้งซ่อม

ผลลัพธ์	ประเภทงาน	รายละเอียด	วันที่แจ้งซ่อม	สถานะ
A102	เร่งด่วน	น้ำยาดครองฝึกบัว	2025-03-09 19:22:45	กำลังดำเนินการ

แสดง 1 ถึง 1 จาก 1 ผล

แต่หากการดำเนินการนี้สถานะเสร็จสิ้นแล้วจะมีรายการประวัติการแจ้งซ่อมขึ้นมา

**ประวัติการซ่อมที่เสร็จสิ้น**

ห้องพัก	ประเภทงาน	รายละเอียด	วันที่แจ้งซ่อม	คำใช้จ่ายเพิ่มเติม	สถานะ
A102	เร่งด่วน	น้ำยาดครองฝึกบัว	2025-03-09 19:22:45	0	เสร็จสิ้น

### หน้าแจ้งเตือนพัสดุ

เป็นหน้าที่แสดงจำนวนพัสดุ รายการพัสดุทั้งหมด และการแจ้งเตือนพัสดุเข้ามา

**แจ้งเตือนพัสดุ**

จัดการ แจ้งเตือนพัสดุ

**จำนวนพัสดุ**

**แจ้งเตือน - พัสดุ**

เมื่อที่ห้องพัก

ชื่อผู้รับ

ชนิดพัสดุ

ขนาดเล็ก  ขนาดกลาง  ขนาดใหญ่

แจ้งเตือนพัสดุ

**รายการพัสดุทั้งหมด**

ผลลัพธ์	ชื่อผู้รับ	ขนาด	วันที่พัสดุถึง	สถานะ
	ไม่พบข้อมูล			

แสดง 0 ถึง 0 จาก 0 ผล

ขั้นตอนการแจ้งเตือนพัสดุให้ผู้เช่า

- 1) เลือกห้องพัก

## แจ้งเตือน - พัสดุ

เลขที่ห้องพัก

- 2) ใส่ชื่อผู้รับ

ชื่อผู้รับ

คุณอนันต์

- 3) เลือกขนาดพัสดุ

### ขนาดพัสดุ

ขนาดเล็ก

ขนาดกลาง

ขนาดใหญ่

- 4) กดปุ่มแจ้งเตือนพัสดุ

## แจ้งเตือน - พัสดุ

เลขที่ห้องพัก

A102

ชื่อผู้รับ

คุณอนันต์

ขนาดพัสดุ

ขนาดกลาง

ขนาดใหญ่

**แจ้งเตือนพัสดุ**

ເນື່ອກົດແຈ້ງເຕືອນພັສດຸແລ້ວຈຳນວນພັສດຸຈະແສດງໃນຮາຍການພັສດຸທີ່ໜົມພຣັອມທີ່ສະຖານະການຮັບພັສດຸ

### ຮາຍການພັສດຸກັ້ງໝາດ

ແສດງ 5	ແດວ		ຄົນຫາ:	
ໜັກພັກ	ຈົ່ງຜູ້ຂັບ	ໜານດັກ	ວັນທີພັສດຸ	ສະຖານະ
A102	ຄູມອນນັ້ນທີ່	ຂານດກລາງ	2025-03-09 19:41:26	ບັນລຸໄວ້ຮັບ
ແສດງ 1 ດີ່ງ 1 ຈາກ 1 ແດວ	<	1	>	

ໃນສ່ວນຮາຍການແສດງຈຳນວນພັສດຸທີ່ໜົມພຣັອມທີ່ສະຖານະການຮັບພັສດຸທີ່ຮັບແລ້ວແລະຍັງໄມ້ຮັບວ່າມີກີ່ຂຶ້ນ

## ຈຳນວນພັສດຸ



ຈາກການແສດງໃຫ້ເຫັນວ່າມີພັສເຂົ້າໜຶ່ງຂຶ້ນແລະຍັງໄມ້ໄດ້ຮັບທາງມີການຮັບແລ້ວຈະເປັນດັ່ງການຂ້າງລ່າງ

## ຈຳນວນພັສດຸ



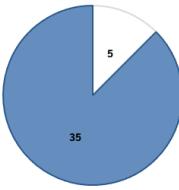
## หน้าจัดการห้องพัก

หน้าจัดการห้องพักจะแสดงจำนวนห้องว่าง/ไม่ว่าง การเพิ่ม แก้ไข แจ้งย้ายออก ลบ ห้องพัก

### ห้องพัก

จัดการ ห้องพัก

#### จำนวนห้องว่าง



ห้องว่าง	35
ห้องไม่ว่าง	5

แสดง	10	ແກ້ວ					
#	ລະຫັດ	ຫ້ອງພັກ	ຮາຄາ	ຮາຍລະເລີຍຄ	ສະຖານະ	ຜູ້ເຂົ້າ	ຈັກການ
1	A101	5000	Free wifi	ມີວ່າງ	Tester 01	<span style="color: orange;">ແກ້ໄຂ</span> <span style="color: green;">ບໍ່ຍອດ</span>	
2	A102	5000	Free wifi	ມີວ່າງ	ຈິຈາກ ບັນຫະອຸບ້ນ	<span style="color: orange;">ແກ້ໄຂ</span> <span style="color: green;">ບໍ່ຍອດ</span>	
3	A103	5000	Free wifi	ມີວ່າງ	ວິສິດຫາ ຮອຄງວາຍ	<span style="color: orange;">ແກ້ໄຂ</span> <span style="color: green;">ບໍ່ຍອດ</span>	
4	A104	5000	Free wifi	ມີວ່າງ	ປະບຸກ ວັນທີອື່ນ	<span style="color: orange;">ແກ້ໄຂ</span> <span style="color: green;">ບໍ່ຍອດ</span>	
5	A105	5000	Free wifi	ມີວ່າງ	ອັດວິນ ຫ້ວຍນິອງ	<span style="color: orange;">ແກ້ໄຂ</span> <span style="color: green;">ບໍ່ຍອດ</span>	

### ເພີ່ມ - ມີຫຼາຍ

ເລກທີ່ຫ້ອງພັກ

ຕົ້ນຫາ:

ເລກທີ່ຫ້ອງພັກ

ຄາເຂົ້າ / ເດືອນ

ຮາຍລະເລີຍຄເພີ່ມເຕີມ

ເພີ່ມຫ້ອງພັກ

ขั้นตอนการเพิ่มห้องพัก

- 1) ระบุเลขห้องพัก

## ເພີ່ມ - ມີຫຼາຍ

### ເລກທີ່ຫ້ອງພັກ

B101

#### ຮາຍລະເລີຍຄເພີ່ມເຕີມ

- 2) ระบุຄ່າເຂົ້າຫ້ອງ/ເດືອນ ທາກໄມ່ມີສາມາດເຮັ້ນວ່າງໄດ້

ຮາຍລະເລີຍຄເພີ່ມເຕີມ

ມີເຄື່ອງໃຊ້ໄຟຟ້າໂຄຣບ + free wifi

3) กดปุ่มเพิ่มห้องพัก

## เพิ่ม - ห้องพัก

เลขที่ห้องพัก	ค่าเช่า / เดือน
B101	7000
รายละเอียดเพิ่มเติม	
มีเครื่องใช้ไฟฟ้าครบ + free wifi	
<b>เพิ่มห้องพัก</b>	

เมื่อกดเพิ่มห้องพักเสร็จแล้ว ห้องพักจะแสดงในรายการห้องพักด้านล่างซึ่งจะแสดง เลขห้องพัก รายละเอียดของห้อง สถานะการเช่า และจัดการห้องพัก

แสดง	แก้					ค้นหา:
#	ห้องพัก	ราคา	รายละเอียด	สถานะ	จัดการ	
41	B101	7000	มีเครื่องใช้ไฟฟ้าครบ + free wifi	ว่าง	<b>แก้ไข</b>	<b>ลบ</b>

### การแก้ไขห้องพัก

เมื่อกดปุ่มแก้ไขที่อยู่ในแถวห้องพักใดๆ จะนำมาสู่หน้าการแก้ไขห้องพักนั้นๆ โดยจะแสดงตั้งภาพข้างล่างนี้

<b>ห้องพัก</b>	
แก้ไข	ห้องพัก
<b>แก้ไข - ห้องพัก</b>	
เลขที่ห้องพัก	ค่าเช่า / เดือน
A101	5000
รายละเอียดเพิ่มเติม	
Free wifi	
<b>แก้ไขห้องพัก</b>	
<b>ยกเลิกการแก้ไข</b>	

หากต้องการแก้ไขห้องพักในส่วนใดๆ สามารถเลือกรายการที่ต้องการเปลี่ยนและกดปุ่มแก้ไขห้องพักได้เลย หากไม่ต้องการเปลี่ยนแปลงใดๆ การกดยกเลิกการแก้ไขจะนำกลับสู่หน้าก่อนหน้าหรือหน้าจัดการห้องพัก

### การย้ายผู้เช่าออก

เมื่อกดปุ่มย้ายผู้เช่าที่อยู่ในแถวห้องพักใดๆ จะแสดงรายละเอียดให้กรอกดังภาพข้างล่างนี้

## ย้ายผู้เช่าออกจากห้อง A101

X

กรุณากรอกข้อมูลการย้ายออก

วันที่ย้ายออก

mm/dd/yyyy



หมายเหตุ

ยกเลิก

ย้ายออกและตั้งเป็นห้องว่าง

ขั้นตอนการย้ายผู้เช่าออกจากห้องพัก

- 1) เลือกวันที่ย้ายออก

กรุณากรอกข้อมูลการย้ายออก

วันที่ย้ายออก

mm/dd/yyyy



March 2025 ▾



หมายเหตุ

Su	Mo	Tu	We	Th	Fr	Sa
23	24	25	26	27	28	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Clear

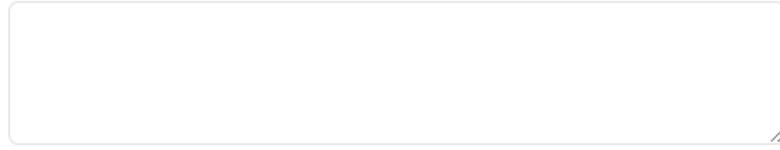
Today

ยกเลิก

ย้ายออกและตั้งเป็นห้องว่าง

- 2) หากมีหมายเหตุสามารถกรอกได้ในช่องหมายเหตุ

## หมายเหตุ



ยกเลิก

ข้ายอกและตั้งเป็นห้องว่าง

- 3) เมื่อกดข้ายอกและตั้งเป็นห้องว่าง ห้องจะเปลี่ยนสถานะเป็นห้องว่างและผู้เช่าจะไม่มีสถานะการอยู่ในห้องพักนั้นๆ

### การลบห้องพัก

ในการ刪除ห้องพักใดๆ เมื่อทำการกดลบจะเป็นการลบห้องพักนั้นออกไป

33	A403	5500	มีแอร์	ว่าง	-	<button style="background-color: #FFC107; border: none; color: black; padding: 5px 10px; border-radius: 5px;">แก้ไข</button>	<button style="background-color: #DC3545; border: none; color: white; padding: 5px 10px; border-radius: 5px;">ลบ</button>
34	A404	5500	มีแอร์	ว่าง	-	<button style="background-color: #FFC107; border: none; color: black; padding: 5px 10px; border-radius: 5px;">แก้ไข</button>	<button style="background-color: #DC3545; border: none; color: white; padding: 5px 10px; border-radius: 5px;">ลบ</button>

ก่อนจะทำการลบระบบจะถามทวนอีกรอบเพื่อความถูกต้องว่าต้องการลบจริงๆหรือไม่

แสดง 10 ผลลัพธ์							ค้นหา:
#	ห้องพัก	ราคา	รายละเอียด	สถานะ	ผู้เช่า	จัดการ	
31	A401	5500	มีแอร์	มีว่าง	วิรเจน สิตาดาอคิดาร	<button style="background-color: #FFC107; border: none; color: black; padding: 5px 10px; border-radius: 5px;">แก้ไข</button>	<button style="background-color: #4CAF50; color: white; border: none; padding: 5px 10px; border-radius: 5px;">ข้ายอก</button>
32	A402	5500	มีแอร์	มีว่าง	กล้าหาญ ใจเต็็ด	<button style="background-color: #FFC107; border: none; color: black; padding: 5px 10px; border-radius: 5px;">แก้ไข</button>	<button style="background-color: #4CAF50; color: white; border: none; padding: 5px 10px; border-radius: 5px;">ข้ายอก</button>
33	A403	5500	มีแอร์	ว่าง	-	<button style="background-color: #FFC107; border: none; color: black; padding: 5px 10px; border-radius: 5px;">แก้ไข</button>	<button style="background-color: #DC3545; border: none; color: white; padding: 5px 10px; border-radius: 5px;">ลบ</button>
34	A404	5500	มีแอร์	ว่าง	-	<button style="background-color: #FFC107; border: none; color: black; padding: 5px 10px; border-radius: 5px;">แก้ไข</button>	<button style="background-color: #DC3545; border: none; color: white; padding: 5px 10px; border-radius: 5px;">ลบ</button>
35	A405	5500	มีแอร์	ไม่ว่าง	-	<button style="background-color: #FFC107; border: none; color: black; padding: 5px 10px; border-radius: 5px;">แก้ไข</button>	<button style="background-color: #DC3545; border: none; color: white; padding: 5px 10px; border-radius: 5px;">ลบ</button>
36	A406	5500	มีแอร์	ไม่ว่าง	สมบ่อง นาดี	<button style="background-color: #FFC107; border: none; color: black; padding: 5px 10px; border-radius: 5px;">แก้ไข</button>	<button style="background-color: #4CAF50; color: white; border: none; padding: 5px 10px; border-radius: 5px;">ข้ายอก</button>
37	A407	5500	มีแอร์	ไม่ว่าง	สุขชัย สุวรรณสวัสดิ์	<button style="background-color: #FFC107; border: none; color: black; padding: 5px 10px; border-radius: 5px;">แก้ไข</button>	<button style="background-color: #4CAF50; color: white; border: none; padding: 5px 10px; border-radius: 5px;">ข้ายอก</button>
38	A408	5500	มีแอร์	ว่าง	-	<button style="background-color: #FFC107; border: none; color: black; padding: 5px 10px; border-radius: 5px;">แก้ไข</button>	<button style="background-color: #DC3545; border: none; color: white; padding: 5px 10px; border-radius: 5px;">ลบ</button>
39	A409	5500	มีแอร์	ไม่ว่าง	มงคล อมขาวด	<button style="background-color: #FFC107; border: none; color: black; padding: 5px 10px; border-radius: 5px;">แก้ไข</button>	<button style="background-color: #4CAF50; color: white; border: none; padding: 5px 10px; border-radius: 5px;">ข้ายอก</button>
40	A410	5500	มีแอร์	ว่าง	-	<button style="background-color: #FFC107; border: none; color: black; padding: 5px 10px; border-radius: 5px;">แก้ไข</button>	<button style="background-color: #DC3545; border: none; color: white; padding: 5px 10px; border-radius: 5px;">ลบ</button>

## หน้าจัดการผู้เช่า

จะมีการทำงานคือแสดงผู้เช่าทั้งหมดและสามารถ เพิ่ม แก้ไข ลบข้อมูลผู้เช่าได้

### ผู้เช่า & ผู้เช่า

จัดการ ผู้เช่า & ผู้เช่า

บล. เพิ่มผู้เช่า

#	ชื่อ	เพศ	บทบาท	จัดการ
1	สมศักดิ์ พังกระแทก	ชาย	ผู้เช่า	<button>แสดง</button> <button>แก้ไข</button> <button>ลบ</button>
2	แสง บานดาสาว	ชาย	ผู้เช่า	<button>แสดง</button> <button>แก้ไข</button> <button>ลบ</button>
3	กิตติมา ใจไม่เคย	หญิง	ผู้เช่า	<button>แสดง</button> <button>แก้ไข</button> <button>ลบ</button>
4	สมหญิง ยิ่งเม่น	หญิง	ผู้เช่า	<button>แสดง</button> <button>แก้ไข</button> <button>ลบ</button>
5	สมศรี มีทอง	หญิง	ผู้เช่า	<button>แสดง</button> <button>แก้ไข</button> <button>ลบ</button>
6	สุราราม มีเงิน	ชาย	ผู้เช่า	<button>แสดง</button> <button>แก้ไข</button> <button>ลบ</button>
7	น้ำใจ ใจเรือง	หญิง	ผู้เช่า	<button>แสดง</button> <button>แก้ไข</button> <button>ลบ</button>
8	จันทร์ มั่นใจ	ชาย	ผู้เช่า	<button>แสดง</button> <button>แก้ไข</button> <button>ลบ</button>
9	ศุภณัฐ งามเกตุสุข	ชาย	ผู้เช่า	<button>แสดง</button> <button>แก้ไข</button> <button>ลบ</button>
10	วราะศิลป์ หัวหงษ์ทอง	ชาย	ผู้เช่า	<button>แสดง</button> <button>แก้ไข</button> <button>ลบ</button>

แสดง 1 ถึง 10 จาก 23 ผล

< 1 2 3 >

### ส่วนของการเพิ่มผู้เช่า

#### ขั้นตอนการเพิ่มผู้เช่า

- กดปุ่มเพิ่มผู้เช่า

ผู้เช่า & ผู้เช่า

เพิ่ม - ผู้เช่า & ผู้เช่า

ชื่อจริง	นามสกุล
เพศ	หมายเหตุบัตรประชาชน
เบอร์โทรศัพท์	รหัสบ้าน
<input type="button" value="เพิ่มผู้เช่า"/>	
<input type="button" value="ยกเลิก"/>	

- กรอกชื่อจริงและนามสกุล

ชื่อจริง	นามสกุล
สุขุมวิท	รักเมือง

- เลือกเพศและกรอกหมายเลขอปตประชาน

เพศ	หมายเลขอปตประชาน
ชาย	12

หากกรอกบัตรประชาชนไม่ครบ 13 ตัว ระบบจะให้ทำการกรอกใหม่อีกครั้ง

#### เพิ่ม - ผู้ใช้งาน & ผู้เช่า

ชื่อจริง	นามสกุล
เพศ	หมายเลขบัตรประชาชน
	เบอร์โทรศัพท์
	รหัสผ่าน

เพิ่มผู้ใช้งาน

ยกเลิก

หมายเลขบัตรประชาชนไม่ถูกต้อง

- 4) กรอกหมายเลขโทรศัพท์ (หมายเลขโทรศัพท์จะเป็นเลขที่เอ้าไว้ใช้ล็อกอินเข้าเว็บไซต์) และรหัสผ่านสำหรับเข้าใช้งานเว็บไซต์

เบอร์โทรศัพท์

รหัสผ่าน

--	--

หากกรอกเบอร์ไม่ครบ 10 ตัวระบบจะให้ทำการกรอกใหม่

#### เพิ่ม - ผู้ใช้งาน & ผู้เช่า

ชื่อจริง	นามสกุล
เพศ	หมายเลขบัตรประชาชน
	เบอร์โทรศัพท์
	รหัสผ่าน

เพิ่มผู้ใช้งาน

ยกเลิก

เบอร์โทรศัพท์ไม่ถูกต้อง

หลังจากกดเพิ่มผู้ใช้งาน รายการเพิ่มจะมาอยู่ในรายการผู้ใช้งาน

#### ผู้ใช้งาน & ผู้เช่า

จัดการ ผู้ใช้งาน & ผู้เช่า

บวก เพิ่มผู้ใช้งาน

แสดง	10	แมว	ค้นหา:	
#	ชื่อ	เพศ	บทบาท	จัดการ
21	Steven Jobs	ชาย	ผู้เช่า	<button>แสดง</button> <button>แก้ไข</button> <button>ลบ</button>
22	ทักษิณ ชินวัตร	ชาย	ผู้เช่า	<button>แสดง</button> <button>แก้ไข</button> <button>ลบ</button>
23	สุรชัย รักเมือง	ชาย	ผู้เช่า	<button>แสดง</button> <button>แก้ไข</button> <button>ลบ</button>

แสดง 21 ถึง 23 จาก 23 ผล

< 1 2 3 >

ในผลของการใช้งานแต่ละແຄວຈະມີການຈັດກາຮອຍຢູ່ສາມປຸນດ້ວຍກັນຄື່ອ ແສດງ ແກ້ໄຂ ແລະລບ ໃນສ່ວນຂອງກາຮ  
ແສດງເມື່ອກົດໄປແລ້ວຈະແສດງຂໍ້ມູນຂອງຜູ້ໃຊ້ງານ

**ຜູ້ໃຊ້ງານ & ຜູ້ເຫຼົ່າ**

ຈົດກາຮ ຜູ້ໃຊ້ງານ & ຜູ້ເຫຼົ່າ

ອຳນວຍຜູ້ໃຊ້ງານ

#	ຊື່	ຮ່ວມເລືອດຜູ້ໃຊ້ : 12	ຈົດກາຮ
1	ສາມເທິກ໌ ທ່ວງກະທັກ	ຊື່ເລີດ: ແສດງ ນາດຕາລາວ ເລກບັດປະຈາກ: 4775892266873 ເນອໂທຣີທີ່ພໍາ: 0874592696 ຮັດສຳນັກ: 0274 ໜ່ອງທັດ: A210	ແສດງ ແກ້ໄຂ ລບ
2	ແສດງ ບາດຕາສາວ		ແສດງ ແກ້ໄຂ ລບ
3	ກິດເທິງ ຊ່ານີ້ຕາຍ		ແສດງ ແກ້ໄຂ ລບ
4	ສາມເທິກ໌ ອິນເມນິນ		ແສດງ ແກ້ໄຂ ລບ
5	ສາມເທິກ໌ ມືກອງ		ແສດງ ແກ້ໄຂ ລບ
6	ສໍາວາດູ ມືເຈີນ		ແສດງ ແກ້ໄຂ ລບ
7	ນໍາໄສໄຈຈິງ	ແກ້ໄຂ ຮຶດ	
8	ຈັກຍຸ ມືນິຈີ		ແສດງ ແກ້ໄຂ ລບ
9	ຄຸກຫັກ ຈຳນເກດສູງ	ໝາຍ ຜູ້ເຫຼົ່າ	ແສດງ ແກ້ໄຂ ລບ
10	ວາພະເຄີບ ຫ້ວຍຫ່າຍທອງ	ໝາຍ ຜູ້ເຫຼົ່າ	ແສດງ ແກ້ໄຂ ລບ

ແສດງ 1 ປຶ້ງ 10 ຈາກ 23 ແລກ

< 1 2 3 >

ຕ່ອມາໃນສ່ວນຂອງກາຮແກ້ໄຂ ເມື່ອກົດແກ້ໄຂໄປແລ້ວເຮົາສາມາດກາຮໄຂຮາຍລະເວີຍດຜູ້ໃຊ້ງານໄດ້ ທາກໄມ່ຕ້ອງກາຮ  
ແກ້ໄຂໃຫ້ກົດປຸ່ມຍົກເລີກ ຈາກນັ້ນຮະບບຈະນຳກລັບມາສູ່ໜ້າເດີມ

**ຜູ້ໃຊ້ງານ & ຜູ້ເຫຼົ່າ**

ຈົດກາຮ ຜູ້ໃຊ້ງານ & ຜູ້ເຫຼົ່າ

**ແກ້ໄຂ - ຜູ້ໃຊ້ງານ & ຜູ້ເຫຼົ່າ**

ຊື່ອໍອັງ	ນາມສັກ
ແສດງ	ບາດຕາສາວ
ເພດ	ໝາຍ
ໝາຍ	ໝາຍເລີບປະປາກສູງ
	ເບີໂທຣີທີ່ພໍາ
	0874592696
	ຮັດສຳນັກ
	****

ແກ້ໄຂຜູ້ໃຊ້ງານ ຍົກເລີກ

และปุ่มสุดท้ายคือปุ่มลบ หากทำการกดลบรอบจะขึ้นให้ทวนอีกรังเพื่อความถูกต้องว่าต้องการลบผู้ใช้คนนี้หรือไม่หากไม่ต้องการให้กดยกเลิก หากต้องการลบให้กดลบ

The screenshot shows a user management interface. At the top, there are navigation icons for back, forward, search, and user profile. Below that is a header with the title "ผู้ใช้งาน & ผู้เช่า". A sub-header indicates the current view: "ผู้ใช้งาน & ผู้เช่า". On the right, there is a button labeled "เพิ่มผู้ใช้งาน". The main area is a table with 10 rows of user data. Each row includes columns for ID, First Name, Last Name, Gender, and Status. To the right of each row is a set of three buttons: ลบ (Delete), แก้ไข (Edit), and สถานะ (Status). A red modal dialog box is overlaid on the table, containing the text "ยืนยันการลบ" (Confirm Deletion) and the question "คุณต้องการลบผู้ใช้งานนี้ใช่หรือไม่?". At the bottom of the modal are two buttons: "ยกเลิก" (Cancel) and "ยืนยัน" (Confirm). At the very bottom of the page, there is a footer with page numbers (1, 2, 3, >) and a link to "ผลลัพธ์" (Results).

### หน้าการจองห้องพัก

จะแสดงรายการห้องที่ว่าง และการจองห้องพักให้กับผู้เช่าโดยจะแสดงดังรูปข้างล่าง

The screenshot shows a booking application interface. At the top, there is a search form with fields for "เลือกห้อง" (Select Room) and "เลือกผู้เช่า" (Select Tenant), both with dropdown menus. Below these are fields for "วันที่เข้า" (Check-in date) and "วันที่ออก" (Check-out date), each with a date picker and a "mm/dd/yyyy" placeholder. A large green button at the bottom of the form is labeled "ยืนยันการจอง" (Confirm Booking). Below the form, there is a section titled "รายการห้องที่ว่าง" (List of Available Rooms). This section contains a table with 5 rows of room information. Each row includes columns for ID, Room Number, Price, Status, and Status. The status column for all rooms shows "ว่าง" (Available). At the bottom of the page, there is a footer with page numbers (1, 2, >) and a link to "ผลลัพธ์" (Results).

## ขั้นตอนการจองห้องพัก

- เลือกห้องที่ว่าง

**จองห้องพัก**

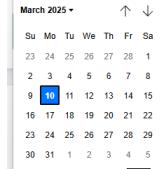
เลือกห้อง	เลือกผู้เช่า
<input type="text" value="A403"/> A403 A404 A405 A408 A410	<input type="text"/> วันที่เข้าออก mm/dd/yyyy
<input type="button" value="บันทึกการจอง"/>	

- เลือกผู้เช่า

**จองห้องพัก**

เลือกห้อง	เลือกผู้เช่า
<input type="text" value="A404"/> วันที่เข้าออก mm/dd/yyyy	<input type="text"/> กิตตินา ชานดา สุขเรือง รักเมว
<input type="button" value="บันทึกการจอง"/>	

- เลือกวันที่ย้ายเข้ามา

วันที่เข้าเช่า	วันที่เข้าออก
<input type="text"/> mm/dd/yyyy	<input type="text"/> mm/dd/yyyy
	
<input type="button" value="บันทึกการจอง"/>	
ค้นหา: <input type="text"/>	

- หากไม่ทราบวันที่จะออกสามารถเว้นว่างไว้ได้

**จองห้องพัก**

เลือกห้อง	เลือกผู้เช่า
<input type="text" value="A404"/> วันที่เข้าเช่า	<input type="text"/> วันที่เข้าออก mm/dd/yyyy
<input type="button" value="บันทึกการจอง"/>	

- กดยืนยันการจอง

เมื่อกดยืนยันการจองแล้วผู้เช่าจะมีข้อมูลการเข้าพัก ณ ห้องนั้นๆ

## หน้าจัดการมิเตอร์ไฟ/น้ำ

หน้านี้จะมีการบันทึกมิเตอร์ไฟฟ้าและน้ำในแต่ละห้องรวมไปถึงการตั้งค่าราคาต่อหน่วยของไฟฟ้าและน้ำ

**บันทึก - มิเตอร์ไฟฟ้า / น้ำ**

เลือกห้อง

มิเตอร์ไฟฟ้า

วันที่บันทึกมิเตอร์ไฟฟ้าล่าสุด : (วันที่)  
เลขมิเตอร์ท่อนบันทึก : (เลข)

มิเตอร์น้ำ

วันที่บันทึกมิเตอร์น้ำล่าสุด : (วันที่)  
เลขมิเตอร์ท่อนบันทึก : (เลข)

**บันทึกการบันทึก**

**กำหนดราคา - ไฟฟ้า / น้ำ**

หมายเหตุ : ราคาต่อหน่วยที่กำหนดทั้งหมด

ค่าไฟฟ้า / หน่วย	ค่าน้ำ / หน่วย
7	4

**บันทึกการกำหนดราคา**

### ขั้นตอนการบันทึกมิเตอร์ไฟฟ้าและน้ำ

- เลือกห้องที่ต้องการบันทึก

**บันทึก - มิเตอร์ไฟฟ้า / น้ำ**

เลือกห้อง

- 
- A101
- A102
- A103**
- A104
- A105
- A106
- A107
- A108
- A109
- A110

2) กรอกตัวเลขในช่องมิเตอร์น้ำและไฟฟ้า ซึ่งจะมีบอกตัวเลขก่อนบันทึกว่าบันทึกล่าสุดวันไหนและมีหน่วยเป็นเท่าใด

- กรอกตัวเลขในช่องมิเตอร์น้ำและไฟฟ้า ซึ่งจะมีบอกตัวเลขก่อนบันทึกว่าบันทึกล่าสุดวันไหนและมีหน่วยเป็นเท่าใด

**บันทึก - มิเตอร์ไฟฟ้า / น้ำ**

เลือกห้อง

67

วันที่บันทึกมิเตอร์ไฟฟ้าล่าสุด : 06-03-2025  
เลขมิเตอร์ท่อนบันทึก : 40

4

วันที่บันทึกมิเตอร์น้ำล่าสุด : 06-03-2025  
เลขมิเตอร์ท่อนบันทึก : 2

**บันทึกการบันทึก**

- กดยืนยันการบันทึก

การกำหนดราคาต่อหน่วยไฟฟ้าเป็นการกำหนดราคาต่อหน่วยของห้องเช่าทุกห้องเพื่อนำไปคำนวณค่าไฟฟ้าและค่าน้ำของผู้เช่า

### ขั้นตอนการกำหนดราคา ค่าไฟ-ค่าน้ำ

- กรอกราคาที่ต้องการต่อหน่วยในช่องค่าไฟและค่าน้ำ

กำหนดราคา - ไฟฟ้า / น้ำ

หมายเหตุ : ราคาต่อหน่วยที่กันทุกห้อง

ค่าไฟฟ้า / น้ำ

7

ค่าน้ำ / น้ำ

4

ยืนยันการกำหนดราคา

- กดยืนยันการกำหนดราคา

หน้าใบแจ้งหนี้และใบเสร็จ

หน้านี้จะมีการออกบิล แสดงรายการ ใบเสร็จรับเงินและใบแจ้งหนี้

จัดการธุรกรรม

10/03/2025

ใบเสร็จรับเงิน | ใบแจ้งหนี้

สร้างใบชำระเงิน

#	ห้อง	วันออกบิล	วันกำหนดชำระ	สถานะ	ดำเนินการ
3	A103	2025-03-08	2025-03-09	ยังไม่ชำระ	<span style="background-color: green; color: white;">ชำระเงิน</span>
2	A102	2025-03-01	2025-03-03	ยังไม่ชำระ	<span style="background-color: green; color: white;">ชำระเงิน</span>
1	A101	2025-03-06	2025-03-07	ชำระแล้ว	<span style="background-color: black; color: white;">ชำระแล้ว</span>

แสดง 1 ถึง 3 จาก 3 ผล

สิงแกรกที่ต้องทำก่อนจะสร้างใบแจ้งหนี้ต้องสร้างใบชำระเงินก่อนเป็นอย่างแรก

สร้างใบชำระเงิน

ห้อง

ชื่อเจ้าของห้อง

ค่าเช่าห้อง

วันกำหนดชำระ

จำนวนเงิน

เพิ่มบิล

## ขั้นตอนการสร้างใบชำระเงิน

- 1) เลือกห้องที่จะออกใบชำระเงิน

**สร้างใบชำระเงิน**

ห้อง	ชื่อเจ้าของห้อง	ค่าเช่าห้อง
<input type="text" value="A101 (Meter: 48)"/> <input checked="" type="text" value="A102 (Meter: 83)"/> <input type="text" value="A103 (Meter: 50)"/> <input type="text" value="A104 (Meter: 51)"/> <input type="text" value="A105 (Meter: 52)"/>	<input type="text" value="ธิรา豫 บันทะอุบ้าน"/> <input type="text" value="03/10/2025"/>	<input type="text" value="วันกำหนดชำระ"/> <input type="text" value="mm/dd/yyyy"/>
<input type="button" value="เพิ่มบิล"/>		

- 2) จานวนจะมีรายละเอียดของผู้เช่าขึ้นมา

**สร้างใบชำระเงิน**

ห้อง	ชื่อเจ้าของห้อง	ค่าเช่าห้อง
<input type="text" value="A102 (Meter: 83)"/> <input type="text" value="03/10/2025"/>	<input type="text" value="ธิรา豫 บันทะอุบ้าน"/> <input type="text" value="03/10/2025"/>	<input type="text" value="วันกำหนดชำระ"/> <input type="text" value="mm/dd/yyyy"/>
<input type="button" value="เพิ่มบิล"/>		

- 3) เลือกวันกำหนดชำระหากไม่มีสามารถเว้นว่างได้

**สร้างใบชำระเงิน**

ห้อง	ชื่อเจ้าของห้อง	ค่าเช่าห้อง
<input type="text" value="A102 (Meter: 83)"/> <input type="text" value="03/10/2025"/>	<input type="text" value="ธิรา豫 บันทะอุบ้าน"/> <input type="text" value="03/10/2025"/>	<input type="text" value="วันกำหนดชำระ"/> <input type="text" value="03/14/2025"/>
<input type="button" value="เพิ่มบิล"/>		

- 4) กดเพิ่มบิล

หลังจากที่เพิ่มบิลมาแล้วบิลจะแสดงในหน้ารายการบิลซึ่งจะมีการดำเนินการที่แตกต่างกันตามสถานะบิล

ลำดับ	ห้อง	วันออกบิล	วันกำหนดชำระ	สถานะ	ดำเนินการ
4	A102	2025-03-10	2025-03-14	ยังไม่ชำระ	<input type="button" value="สร้างใบเมล์หนี้"/> <input type="button" value="สร้างใบเสร็จรับเงิน"/>
3	A103	2025-03-08	2025-03-09	ยังไม่ชำระ	<input type="button" value="สร้างใบเมล์หนี้"/> <input type="button" value="สร้างใบเสร็จรับเงิน"/>
2	A102	2025-03-01	2025-03-03	ยังไม่ชำระ	<input type="button" value="สร้างใบเมล์หนี้"/> <input type="button" value="สร้างใบเสร็จรับเงิน"/>
1	A101	2025-03-06	2025-03-07	ชำระแล้ว	<input type="button" value="ชำระแล้ว"/>

แสดง 1 ถึง 4 จาก 4 ผล

เมื่อสร้างบิลมาแล้วจะไม่สามารถกดสร้างใบเสร็จรับเงินได้ ต้องกดสร้างใบแจ้งหนี้ก่อน

### ขั้นตอนการสร้างใบแจ้งหนี้

- 1) กดสร้างใบแจ้งหนี้ในແຄວຂອງທ້ອນນັ້ນๆ

#	ห้อง	วันออกบิล	วันกำหนดชำระ	สถานะ	ดำเนินการ
4	A102	2025-03-10	2025-03-14	ยังไม่ชำระ	<button>สร้างใบแจ้งหนี้</button> <button>สร้างใบเช็คบิลเงิน</button>
3	A103	2025-03-08	2025-03-09	ยังไม่ชำระ	<button>สร้างใบเช็คบิลเงิน</button>
2	A102	2025-03-01	2025-03-03	ยังไม่ชำระ	<button>สร้างใบเช็คบิลเงิน</button>
1	A101	2025-03-06	2025-03-07	ชำระแล้ว	<button>ข้อมูลเดิม</button>

แสดง 1 ถึง 4 จาก 4 รายการ < > 1 >

- 2) เมื่อกดเข้ามาแล้วระบบจะเข้าสู่หน้าสร้างใบแจ้งหนี้

#### สร้างใบแจ้งหนี้

สร้าง ใบแจ้งหนี้

ค่าไฟ				
มีเดือนที่พักราคาบัน	มีเดือนที่พักถัดไป	ใช้ไฟฟ้าไป	ราคา/หน่วย	ราคารวม
2025-03-09	2025-03-09 22:59:03	78	7	546

ค่าน้ำ				
มีเดือนที่พักราคาบัน	มีเดือนที่พักถัดไป	ใช้น้ำไป	ราคา/หน่วย	ราคารวม
2025-03-09	2025-03-09 22:59:03	4	4	16

← ย้อนกลับ

#### ค่าใช้จ่ายเพิ่มเติม

รายการ		ราคารวม
ค่าเช่า		5000
รายการ		ราคา
		5562

บันทึกการสร้าง

ระบบจะแสดงข้อมูลการใช้ไฟฟ้าและน้ำ รวมไปถึงราคากำไรเช่าและหากมีรายการเพิ่มเติมสามารถเพิ่มได้ที่นี่ ซึ่งหากมีค่าใช้จ่ายในการซ่อม ค่าใช้จ่ายการซ่อมก็จะขึ้นมาด้วย

#### สร้างใบแจ้งหนี้

สร้าง ใบแจ้งหนี้

ค่าไฟ				
มีเดือนที่พักราคาบัน	มีเดือนที่พักถัดไป	ใช้ไฟฟ้าไป	ราคา/หน่วย	ราคารวม
2025-03-09	2025-03-09 22:59:03	78	7	546

ค่าน้ำ				
มีเดือนที่พักราคาบัน	มีเดือนที่พักถัดไป	ใช้น้ำไป	ราคา/หน่วย	ราคารวม
2025-03-09	2025-03-09 22:59:03	4	4	16

← ย้อนกลับ

#### ค่าใช้จ่ายเพิ่มเติม

รายการ		ราคารวม
ค่าเช่า		5000
ค่าบำรุงรักษา		200
รายการ		ราคา
		5762

บันทึกการสร้าง

- 3) กดยืนยันการสร้าง
- 4) เมื่อกดยืนยันการสร้างระบบจะนำมายังหน้ารายการใบแจ้งหนี้

**ใบแจ้งหนี้**

จัดการ ใบแจ้งหนี้

ผลลัพธ์	ห้อง	วันออกบิล	วันกำหนดชำระ	สถานะ	ดำเนินการ
4	A102	2025-03-10	2025-03-14	ยังไม่ชำระ	ใบแจ้งหนี้
3	A103	2025-03-08	2025-03-09	ยังไม่ชำระ	ใบแจ้งหนี้
2	A102	2025-03-01	2025-03-03	ยังไม่ชำระ	ใบแจ้งหนี้
1	A101	2025-03-06	2025-03-07	ชำระแล้ว	ใบแจ้งหนี้

ผลลัพธ์ 1 จาก 4 ผลลัพธ์ < 1 >

- 5) เมื่อกดดำเนินการใบแจ้งหนี้ ระบบจะนำไปสู่หน้าใบแจ้งหนี้

**ใบแจ้งหนี้**

ชื่อ-นามสกุล:	วรรษ พิษณุโลก	วันออกบิล:	2025-03-10
ห้อง:	A102	วันกำหนดชำระ:	2025-03-14

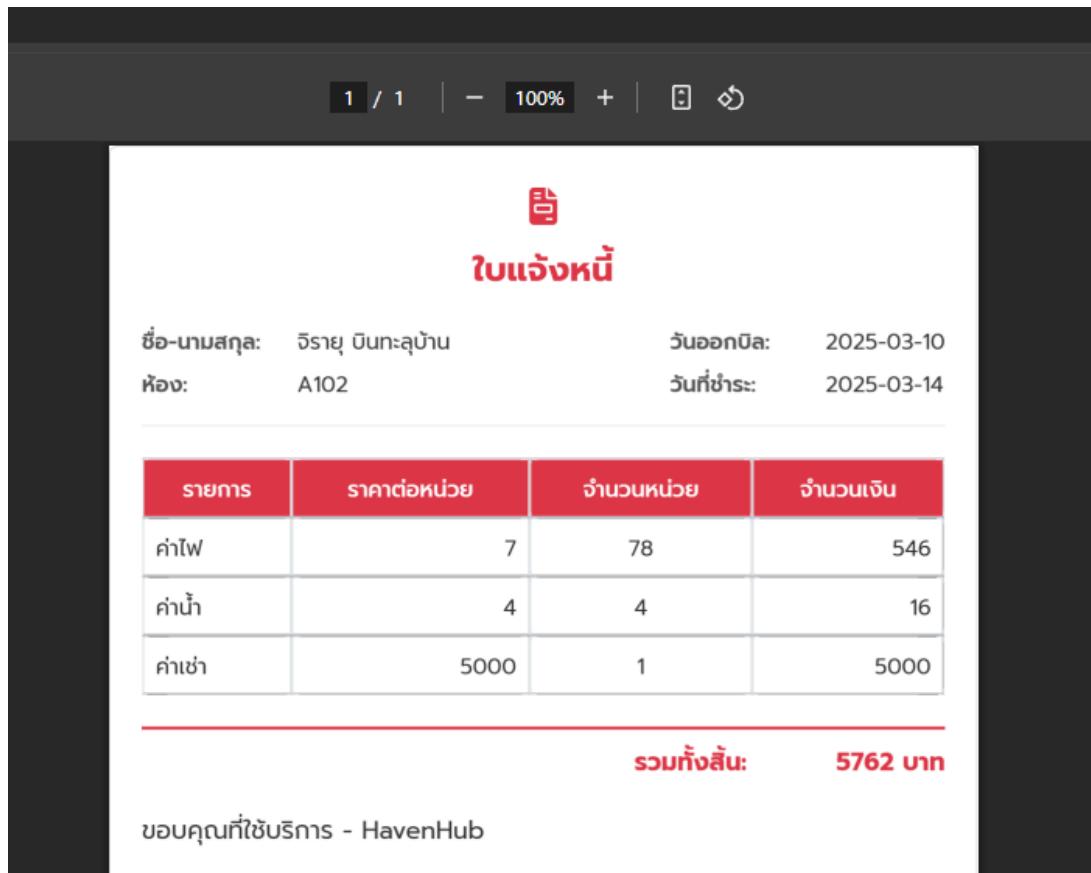
รายการ	ราคาต่อหน่วย	จำนวนหน่วย	รวมเบี้ยเช่า
ค่าไฟ	7	78	546
ค่าน้ำ	4	4	16
ค่าเช่า	5000	1	5000

รวมทั้งสิ้น: **5762 บาท**

ขอบคุณที่ใช้บริการ - HavenHub

[ดาวน์โหลดเป็น PDF](#) [← ย้อนกลับ](#)

6) เมื่อดาวน์โหลดเป็น PDF จะโหลดไฟล์ใบแจ้งหนี้เป็น PDF ดังตัวอย่าง



ขอบคุณที่ใช้บริการ - HavenHub

เมื่อสร้างใบแจ้งหนี้เสร็จแล้วระบบจะไม่แสดงการสร้างใบแจ้งหนี้ จะแสดงแค่สร้างใบเสร็จชำระเงิน เมื่อลูกค้าชำระเงินเรียบร้อยแล้วให้กดสร้างใบเสร็จชำระเงิน

ค้นหา:					
#	ห้อง	วันออกบิล	วันกำหนดชำระ	สถานะ	ดำเนินการ
4	A102	2025-03-10	2025-03-14	ยังไม่ชำระ	<button>สร้างใบเสร็จรับเงิน</button>
3	A103	2025-03-08	2025-03-09	ยังไม่ชำระ	<button>สร้างใบเสร็จรับเงิน</button>
2	A102	2025-03-01	2025-03-03	ยังไม่ชำระ	<button>สร้างใบเสร็จรับเงิน</button>
1	A101	2025-03-06	2025-03-07	ชำระแล้ว	<button>ชาร์จมือถือ</button>

แสดง 1 ถึง 4 จาก 4 ผล

< 1 >

### ขั้นตอนการสร้างใบเสร็จชำระเงิน

1) กดสร้างใบเสร็จชำระเงินในแถบของห้องนั้นๆ

ค้นหา:					
#	ห้อง	วันออกบิล	วันกำหนดชำระ	สถานะ	ดำเนินการ
4	A102	2025-03-10	2025-03-14	ยังไม่ชำระ	<button>สร้างใบเสร็จรับเงิน</button>
3	A103	2025-03-08	2025-03-09	ยังไม่ชำระ	<button>สร้างใบเสร็จรับเงิน</button>
2	A102	2025-03-01	2025-03-03	ยังไม่ชำระ	<button>สร้างใบเสร็จรับเงิน</button>
1	A101	2025-03-06	2025-03-07	ชำระแล้ว	<button>ชาร์จมือถือ</button>

แสดง 1 ถึง 4 จาก 4 ผล

< 1 >

2) ตรวจสอบรายละเอียด

มิเตอร์ไฟฟ้าที่วาง	มิเตอร์ไฟฟ้าที่รับ	ใช้ไฟฟ้าไป	ราคา/หน่วย	รวมรวม
2025-03-09	2025-03-09 22:59:03	78	7	546

มิเตอร์น้ำที่วาง	มิเตอร์น้ำที่รับ	ใช้น้ำไป	ราคา/หน่วย	รวมรวม
2025-03-09	2025-03-09 22:59:03	4	4	16

รายการ		รายการ
ค่าเช่า		5000
ค่าบำรุงรักษา		200

ช่องทางการชำระเงิน  
 เงินสด  โอน  อื่นๆ

5762 ยืนยันการสร้าง

3) เลือกช่องทางการชำระเงินที่ผู้เช่าได้ชำระมา

มิเตอร์ไฟฟ้าที่วาง	มิเตอร์ไฟฟ้าที่รับ	ใช้ไฟฟ้าไป	ราคา/หน่วย	รวมรวม
2025-03-09	2025-03-09 22:59:03	78	7	546

มิเตอร์น้ำที่วาง	มิเตอร์น้ำที่รับ	ใช้น้ำไป	ราคา/หน่วย	รวมรวม
2025-03-09	2025-03-09 22:59:03	4	4	16

รายการ		รายการ
ค่าเช่า		5000
ค่าบำรุงรักษา		200

ช่องทางการชำระเงิน  
 เงินสด  โอน  อื่นๆ

5762 ยืนยันการสร้าง

4) จานวนกดยืนยันการสร้าง

5) ระบบจะนำมาสู่หน้ารายการใบเสร็จรับเงิน

เลขที่	ห้อง	วันออกบิล	วันกำหนดชำระ	สถานะ	ดำเนินการ
4	A102	2025-03-10	2025-03-14	ชำระแล้ว	<span style="background-color: green; color: white; border: 1px solid green; padding: 2px;">ใบเสร็จรับเงิน</span>
1	A101	2025-03-06	2025-03-07	ชำระแล้ว	<span style="background-color: green; color: white; border: 1px solid green; padding: 2px;">ใบเสร็จรับเงิน</span>

ผลลัพธ์ 1 จาก 2 ผลลัพธ์

- 6) เมื่อกดที่ไปเสร็จชำระเงินในแต่ละແຄວຂອງທ້ອງນັ້ນຈະນຳໄປສູ່ທັນໄປເສົ້າຈັກເຈັບເງິນ

รายการ	ราคาต่อหน่วย	จำนวนหน่วย	จำนวนเงิน
ค่าไฟ	7	78	546
ค่าน้ำ	4	4	16
ค่าเช่า	5000	1	5000

รวมทั้งสิ้น: 5762 บาท

ขอบคุณที่ใช้บริการ - HavenHub

[ดาวน์โหลดเป็น PDF](#) [บันทึก](#)

- 7) เมื่อกดดาวน์โหลดเป็น PDF จะโหลดໄປເສົ້າຈັກເຈັບເງິນເປັນໄຟລ໌ PDF

รายการ	ราคาต่อหน่วย	จำนวนหน่วย	จำนวนเงิน
ค่าไฟ	7	78	546
ค่าน้ำ	4	4	16
ค่าเช่า	5000	1	5000

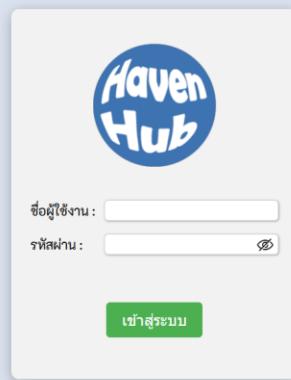
รวมทั้งสิ้น: 5762 บาท

ขอบคุณที่ใช้บริการ - HavenHub

ສໍາหารັບຜູ້ເຂົ້າ

ຫນ້າເຂົ້າສູ່ຮະບບ

ໃນຫນ້າແຮກທີ່ເຮົາຈະພບຄືອຫນ້າເຂົ້າສູ່ຮະບບ



ເມື່ອເຂົ້າສູ່ຮະບບສໍາເລັດ

ຈະພບກັບຫນ້າແຮກດັ່ງການພ້າງຕ່າງ

The image displays the 'HavenHub' mobile application interface. At the top, there is a dark header bar with the 'HavenHub' logo and a user icon. On the left side, a vertical sidebar contains a 'MENU' section with options: 'ຫນ້າແຮກ' (selected), 'ບັນທຶກມືນຄອງໄຟ້ຫນ້າ', 'ເລື່ອງຮັມຈິງຫອມ', and 'ແຈ້ງເຄືອນຫັດ'. Below this is a 'PROFILE' section with 'ອອກຈາກຮະບນ'. The main content area features a large 'Haven Hub' logo and the text 'ຍິນດีຕ້ອນຮັບ, ຈີරາຍ'. It also shows 'ຫົວໜ້າ: A102', 'ຕົວໜ້າຮັບ: ທຸກຮອນເກົ້າ', and 'ຫົວໜ້າ: 0 ຊົ່ວໂມງ'. Below this is a chart titled 'ຄ່າໃຊ້ຈ່າຍແຕ່ລະເດືອນ' (Cost consumption per month) with a Y-axis ranging from 3,000 to 6,000. The chart area is mostly blank, showing a single light blue bar.

## เมนูการใช้งาน

ในส่วนของผู้ใช้เม뉴การใช้งานจะประกอบไปด้วย

- หน้าแรก ซึ่งจะประกอบไปด้วยการแสดงค่าใช้จ่ายในแต่ละเดือนและรายการที่ต้องซ่อม
- หน้าบันทึกมิเตอร์ไฟ/น้ำ แสดงรายละเอียดการจดมิเตอร์ล่าสุดและค่าไฟฟ้า ค่าน้ำต่อหน่วยเป็นเท่าใด
- ส่วนเรื่องแจ้งซ่อม เป็นหน้าที่สามารถส่งเรื่องไปให้เจ้าของหอพักทราบได้ว่าต้องการซ่อมอะไร
- แจ้งเตือนพัสดุ หน้าที่แจ้งเตือนพัสดุเข้าและการยืนยันรับพัสดุ

### หน้าแรก

ในส่วนของหน้าแรกจะแสดงค่าใช้จ่ายในแต่ละเดือนที่ผ่านมาว่าแต่ละเดือนจ่ายไปเท่าไหร่บ้าง และแสดงรายการซ่อมที่กำลังดำเนินการอยู่ว่าถึงขั้นตอนใดแล้ว

The screenshot shows the main dashboard of the Haven Hub application. At the top, there is a header bar with a logo, a search icon, and user profile icons. Below the header, the application title "Haven Hub" is displayed inside a blue circular logo. To the right of the logo, the text "ยินดีต้อนรับ, วิราษ" (Welcome, Viras) is shown. Below this, there are three status indicators: "ห้อง: A102", "ค้างชำระ: ชาระแล้ว", and "พัสดุ: 0 ชิ้น".

**ค่าใช้จ่ายแต่ละเดือน**

A chart titled "ค่าใช้จ่าย" (Expenses) shows a single bar representing the total expense for March 2025, which is 5,762. The y-axis ranges from 3,000 to 6,000.

**รายการ(กี่ต้อง)ซ่อม**

This section displays a table of maintenance requests. The columns are: ประเภทงาน (Type of Work), รายละเอียด (Details), ค่าใช้จ่ายเพิ่มเติม (Additional Expenses), สถานะ (Status), and วันที่แจ้งซ่อม (Report Date). One row is visible:

ประเภทงาน	รายละเอียด	ค่าใช้จ่ายเพิ่มเติม	สถานะ	วันที่แจ้งซ่อม
เร่งด่วน	อุบัติเหตุไม่ได้	0	แจ้งเรื่อง	2025-03-09 23:26:36

At the bottom left, it says "แสดง 1 ถึง 1 จาก 1 ผล" (Displaying 1 to 1 of 1 results). At the bottom right, there are navigation arrows for the table.

## ตรวจบันทึกมิเตอร์ไฟ/น้ำ

หน้านี้จะแสดงให้เห็นว่ามิเตอร์น้ำและมิเตอร์ไฟฟ้าที่ใช้ไปล่าสุดนั้นใช้ไปเท่าไหร่แล้ว นอกจากนี้ยังมีเรื่องราคาดต่อหน่วยของน้ำและไฟฟ้าอยู่

### ตรวจบันทึกมิเตอร์

ตรวจบันทึก มิเตอร์

#### บันทึกมิเตอร์ไฟฟ้า & น้ำ

เลือกห้องที่ต้องการเช็คเมเตอร์

A102

มิเตอร์ค่าไฟฟ้า

78

วันที่บันทึกมิเตอร์ไฟฟ้าล่าสุด : 09-03-2025

มิเตอร์ค่าน้ำ

4

วันที่บันทึกมิเตอร์น้ำล่าสุด : 09-03-2025

#### ราคา - ไฟฟ้า / น้ำ

หมายเหตุ : ราคาที่อยู่บนงบท่านทุกท้อง

ค่าไฟฟ้า / หน่วย

7

ค่าน้ำ / หน่วย

4

### ส่งเรื่องแจ้งซ่อม

หน้านี้แสดงการส่งเรื่องซ่อมและประวัติการส่งเรื่อง

#### ส่งเรื่องแจ้งซ่อม

จัดการ แจ้งซ่อม

#### ส่งเรื่องแจ้งซ่อม

เลือกห้องที่ต้องการแจ้ง

-

ประเภทงาน

-

รายละเอียด

ยืนยันการแจ้งซ่อม

#### ประวัติการแจ้งซ่อม

ผลลัพธ์ 5 ผลลัพธ์

ค้นหา:

ห้องพัก	ประเภทงาน	รายละเอียด	วันที่แจ้งซ่อม	ค่าใช้จ่ายเพิ่มเติม	สถานะ
A102	เร่งด่วน	ลูกบิดเปิดไม่ได้	2025-03-09 23:26:36	0	แจ้งเรื่อง
A102	เร่งด่วน	น้ำยาคงคลังฝึกบัว	2025-03-09 23:09:12	200	เสร็จสิ้น

ผลลัพธ์ 1 ถึง 2 จาก 2 ผลลัพธ์

< 1 >

## ขั้นตอนการล่งเรืองแจ้งซ่อม

- เลือกห้องแจ้งเรือง

**ส่งเรื่องแจ้งซ่อม**

ห้องการ แจ้งซ่อม

**ส่งเรื่องแจ้งซ่อม**

เลือกห้องที่ต้องการแจ้ง

A102

ประเภทงาน

ยืนยันการแจ้งซ่อม

- เลือกประเภทงาน

**ส่งเรื่องแจ้งซ่อม**

ห้องการ แจ้งซ่อม

**ส่งเรื่องแจ้งซ่อม**

เลือกห้องที่ต้องการแจ้ง

A102

รายละเอียด

รายการ

ประเภทงาน

เรื่องคุณ

อื่นๆ

ยืนยันการแจ้งซ่อม

- กรอกรายละเอียด

**ส่งเรื่องแจ้งซ่อม**

ห้องการ แจ้งซ่อม

เลือกห้องที่ต้องการแจ้ง

A102

ประเภทงาน

เรื่องคุณ

รายละเอียด

แอร์ตู้

ประเภทงาน

เรื่องคุณ

ยืนยันการแจ้งซ่อม

- กดยืนยันการซ่อม

เมื่อกดยืนยันแล้วระบบจะแสดงในรายการประวัติการซ่อมและรายการที่ต้องซ่อมในหน้าแรก ส่วนรายการซ่อมเสร็จแล้วหากมีค่าใช้จ่ายเพิ่มเติมก็จะแสดงค่าใช้จ่ายเพิ่มเติมด้วย

**ประวัติการแจ้งซ่อม**

ห้องพัก	ประเภทงาน	รายละเอียด	วันที่แจ้งซ่อม	ค่าใช้จ่ายเพิ่มเติม	สถานะ
A102	เรื่องคุณ	แอร์ตู้	2025-03-09 23:41:28	0	แจ้งเรือง
A102	เรื่องคุณ	ถูกบิดเปิดไม่ได้	2025-03-09 23:26:36	0	แจ้งเรือง
A102	เรื่องคุณ	น้ำยาทำความสะอาด	2025-03-09 23:09:12	200	เสร็จสิ้น

แสดง 5 ผล 3 จาก 3 ผล

< 1 >

## รายการ(ก่อต้อง)ซ่อน

ประเภทงาน	รายละเอียด	ค่าใช้จ่ายเพิ่มเติม	สถานะ	วันที่แจ้งซ่อน
เร่งด่วน	ลูกบิเบเดเม็ด	0	แจ้งเรื่อง	2025-03-09 23:26:36
เร่งด่วน	แอร์ดับ	0	แจ้งเรื่อง	2025-03-09 23:41:28

แสดง 1 ถึง 2 จาก 2 ผล

หน้าแจ้งพัสดุ

หน้าแจ้งพัสดุจะแสดงรายการพัสดุที่เข้ามาและประวัติการรับพัสดุ

## แจ้งเตือนพัสดุ - ของห้อง A102

ข้อมูล แจ้งเตือนพัสดุ

### พัสดุที่กำลังรอรับ

ห้องพัก	ชื่อผู้รับ	ขนาดพัสดุ	วันที่พัสดุถึง	ยืนยันการรับพัสดุ
ไม่พบข้อมูล				

แสดง 0 ถึง 0 จาก 0 ผล

### ประวัติการรับพัสดุ

ห้องพัก	ชื่อผู้รับ	ขนาดพัสดุ	วันที่พัสดุถึง	วันที่ยืนยันรับพัสดุ	สถานะ
A102	A102	ขนาดใหญ่	2025-03-09 23:45:56	2025-03-09 23:46:57	รับแล้ว

หากมีพัสดุเข้ามายังอยู่ในรายการพัสดุรอรับ

## แจ้งเตือนพัสดุ - ของห้อง A102

ข้อมูล แจ้งเตือนพัสดุ

### พัสดุที่กำลังรอรับ

ห้องพัก	ชื่อผู้รับ	ขนาดพัสดุ	วันที่พัสดุถึง	ยืนยันการรับพัสดุ
A102	A102	ขนาดใหญ่	2025-03-09 23:45:56	ยืนยันการรับ

แสดง 1 ถึง 1 จาก 1 ผล

เมื่อกดยืนยันรับพัสดุเรียบร้อยแล้ว พัสดุที่รับแล้วจะไปอยู่ในประวัติการรับ

### ประวัติการรับพัสดุ

ห้องพัก	ชื่อผู้รับ	ขนาดพัสดุ	วันที่พัสดุถึง	วันที่ยืนยันรับพัสดุ	สถานะ
A102	A102	ขนาดใหญ่	2025-03-09 23:45:56	2025-03-09 23:46:57	รับแล้ว