

**Database System  
Project Report  
Kost Management System using PHP and mySQL**



Created by:

Michelle  
Shanty

001202400017  
001202400006

Informatics Class 1 Batch 2024

# CHAPTER A

## Database Design Case Study

### A1. Case study overview

This website is used by kost residents to view information related to their room, check payment status, and submit maintenance requests when needed. The system provides a convenient platform that allows each resident to interact with the kost management without needing to contact the admin directly.

Case Study Form:

1. Room Form:
  - Room ID: (Auto Generated ID)
  - Room Number: (Number)
  - Floor
  - Price: (Price per Month)
  - Facilities: (text)
  - Status: (Available/Occupied)
2. Resident Form
  - Resident ID: (Auto-generated ID)
  - User id: (from user table)
  - Name: (Resident's Name)
  - Phone: (Phone Number)
  - Emergency contact
  - Id card number
  - Move-in Date:
  - Move-out Date:
  - Status
3. Payment Form
  - Payment ID: (Auto-generated ID)
  - Resident ID: (Linked from Resident)
  - Room id (linked from room table)
  - Payment date
  - Payment method
  - Description
  - Status: (Paid/Unpaid)
  - Created at
4. Maintenance Request Form
  - Request ID: (Auto-generated ID)

- Resident ID: (Linked from Resident)
- Room ID: (Linked from Room)
- Request type
- Description
- Request date
- Status: (Pending/In Progress/Completed)
- Resolved Date
- Notes

### Case Study Outline

The system provides a user-friendly platform for kost residents to monitor their room and rent details, check the status of payments, and report any problems in their rooms easily. This increases transparency and helps maintain a good communication flow between the resident and the kost manager.

### **A2. Functions supported by the database (business function)**

1. User Register and Login
  - Users must have an account before accessing the website features
  - Users need to register by filling in personal details such as name,
  - After registering, users can log in using their username, and password
2. Room Viewing and Status Checking
  - Users can view available rooms with details such as room type, price, and status (available/occupied)
3. Resident Dashboard
  - After logging in, users can view their personal profile and room detail
4. Payment History and Status
  - Users can view a list of their monthly payments including month, amount, and status (paid/unpaid)
5. Submitting Maintenance Request
  - Users can submit a maintenance request by filling in a form with issue details
  - Users must select their room and write a description of the issue
6. Tracking Maintenance Request Status
  - Users can track the status of their submitted maintenance request (pending, in progress, or completed)
7. Update Resident Contact Information
  - Users can update their contact number through their profile page

### **A3. Data Requirements**

#### 1. Rooms

- Room\_id (Primary Key)
- Room number
- Floor
- Price
- Facilities
- Status

#### 2. Residents

- Resident\_id (Primary Key)
- User id (Foreign Key)
- Room id (Foreign Key)
- Full name
- Phone
- Emergency contact
- Id card number
- Move\_in\_date
- Move\_out\_date
- Status

#### 3. Payments

- Payment\_id (Primary Key)
- Resident\_id (Foreign Key)
- Room id (Foreign key)
- Payment date
- Payment method
- Payment proof
- Description
- Status
- Created at

#### 4. Maintenance\_request

- Request\_id (Primary Key)
- Resident\_id (Foreign Key)
- Room\_id (Foreign Key)
- Request id
- Description
- Request\_date
- Status

- Resolved date
  - Notes
5. Userser id (Primary Key)
    - Username
    - Password
    - Email
    - Role
    - Created at

#### **A4. Business Rule**

1. User has to register or login before accessing the features
2. Each resident can only be linked to one room at a time
3. Room data must be provided with complete details (type, price, and status)
4. Payment must include month, amount, and status
5. Maintenance request must be linked to a valid room and resident
6. Move-out date cannot be earlier than move-in date
7. Request date must be the current date or earlier
8. Only logged-in residents can submit maintenance requests
9. Resident must fill contact number with valid format
10. Resident can only view their own data

# CHAPTER B

## Conceptual data modelling

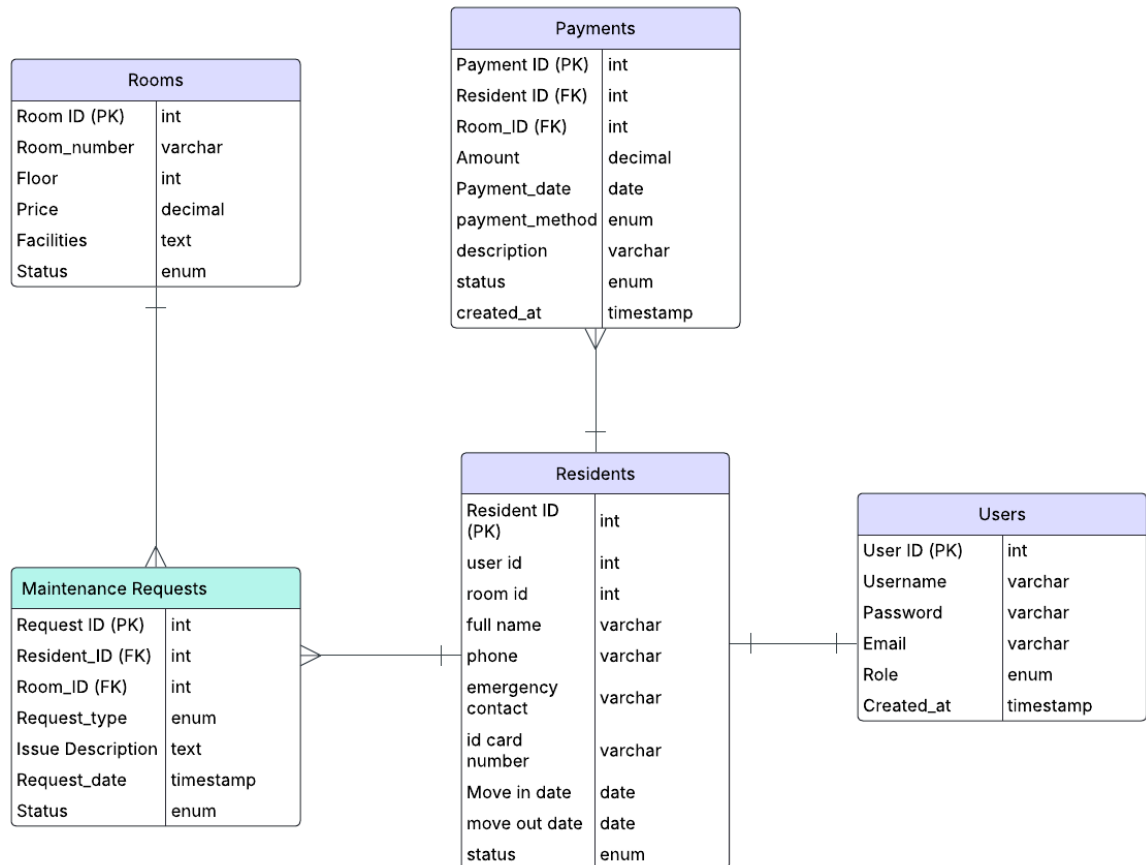
### B1. Overview

A kost management website that helps residents manage their room-related needs. Users can log in to view room details, including type, price, and availability. They can also check their monthly payment history and payment status. The website allows residents to submit and track maintenance requests directly. This platform simplifies communication with kost management without needing direct admin contact.

### B2. Revised Business Rules and Assumption

The business rules is remain unchanged from part A

### B3. Entity Relationship Diagram (ERD)



#### **B4. Justification ERD based on the business rules**

- Each resident have their own unique resident ID (PK)
- Connection between resident and payment is one to many
- Each payment have a unique payment ID (PK)
- Each room have their unique ID
- Connection between rooms and maintenance request is one to many
- Each maintenance request have unique ID, and also resident ID, and room ID to identify who submit the maintenance request
- Each user have their unique user ID
- Connection between resident and user is one to one
- Each resident can only has one user/account

# CHAPTER C

## Logical design : Schema conversion and Normalisation

### C1. Revised Business rules and assumptions (from part B)

There are some changes in the rules and assumptions

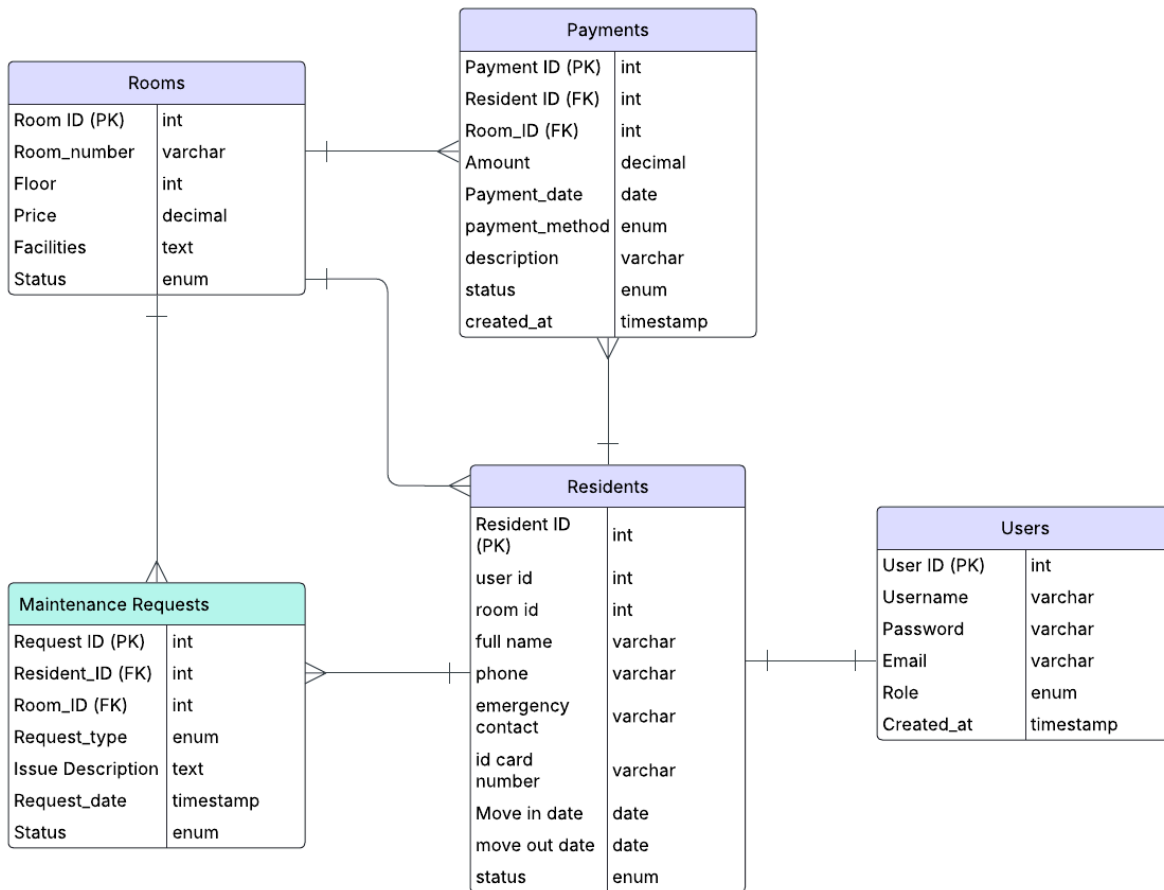
#### BUSINESS RULES

1. Every user must register with their full name, contact number, and password before accessing any features.
2. User login is required to access personalized features such as dashboard, payments, and maintenance requests.
3. Each resident can only be linked to one room at a time
4. Resident details include full name, contact number, move-in and move-out dates.
5. Residents are only allowed to view and update their own data.
6. Room data must be provided with complete details (type, price, and status)
7. Payment must include month, amount, and status
8. Residents can submit maintenance requests for their own rooms only.
9. Each request has a status: Pending, In Progress, or Completed.
10. Maintenance request must be linked to a valid room and resident
11. Maintenance must put the description in it else can't submit
12. Residents can update their contact number at any time through their profile.
13. Contact numbers must follow a valid phone number format.
14. Move-out date cannot be earlier than move-in date
15. Request date must be the current date or earlier
16. Only logged-in residents can submit maintenance requests
17. Resident must fill contact number with valid format
18. Resident can only view their own data

## **ASSUMPTION**

1. Admin functionalities such as adding or editing room data are handled outside of this user-facing system.
2. Room data (Room ID, Type, Price, Status) are already available and managed by admin.
3. The system does not support shared room booking by multiple users at once.
4. Each resident is assumed to have one account only.
5. Time-based constraints (e.g., monthly billing) are managed via system scheduling (e.g., automated by backend).
6. Only logged-in users can access any dashboard-related feature.
7. Request status is updated by kost management, not the resident.
8. Phone numbers must consist of 10 to 15 numeric digits without special characters.
9. All system dates (move-in/out, request dates, payment months) are validated during input.

## C2. Revised ERD of Part B



## C3. Relations

- **Users** (User\_ID PK, Name, Contact, Password)
- **Residents** (Resident\_ID PK, Name, Contact, Move\_in\_date, Move\_out\_date, User\_ID FK)
- **Rooms** (Room\_ID PK, Type, Price, Status)
- **Payments** (Payment\_ID PK, Resident\_ID FK, Month, Amount, Status)
- **Maintenance\_Requests** (Request\_ID PK, Resident\_ID FK, Room\_ID FK, Issue\_Description, Status, Request\_Date)

#### **C4 List of functional dependencies related to each business rules**

- User\_ID → Name, Contact, Password
- Resident\_ID → Name, Contact, Move\_in\_date, Move\_out\_date, User\_ID
- Room\_ID → Type, Price, Status
- Payment\_ID → Resident\_ID, Month, Amount, Status
- Request\_ID → Resident\_ID, Room\_ID, Issue\_Description, Status, Request\_Date
- Resident\_ID → Room\_ID (Each resident can only be linked to one room)
- Request\_ID → Room\_ID (Each maintenance request is tied to a specific room)
- Resident\_ID → User\_ID (One-to-one relationship between resident and user)

#### **C5. Normalisation (Logical design)**

##### **USERS TABLE:**

User\_ID → Name, Contact, Password

##### **RESIDENTS TABLE:**

Resident\_ID → Name, Contact, Move\_in\_date, Move\_out\_date, User\_ID

##### **ROOMS TABLE:**

Room\_ID → Type, Price, Status

##### **PAYMENTS TABLE:**

Payment\_ID → Resident\_ID, Month, Amount, Status

##### **MAINTENANCE REQUEST TABLE**

Request\_ID → Resident\_ID, Room\_ID, Issue\_Description, Status, Request\_Date

##### **RESIDENT TO ROOM RELATIONSHIP**

Resident\_ID → Room\_ID (Each resident can only be linked to one room)

##### **MAINTENANCE REQUEST TO ROOM RELATIONSHIP**

Request\_ID → Room\_ID (Each maintenance request is tied to a specific room)

##### **RESIDENT TO USER RELATIONSHIP**

Resident\_ID → User\_ID (One-to-one relationship between resident and user)

## **B) Final Relations (After Normalization)**

### **1. Users Table**

Users (User\_ID PK, Name, Contact, Password)

### **2. Residents Table**

Residents (Resident\_ID PK, Name, Contact, Move\_in\_date, Move\_out\_date, User\_ID FK)

### **3. Rooms Table**

Rooms (Room\_ID PK, Type, Price, Status)

### **4. Payments Table**

Payments (Payment\_ID PK, Resident\_ID FK, Month, Amount, Status)

### **5. Maintenance Requests Table**

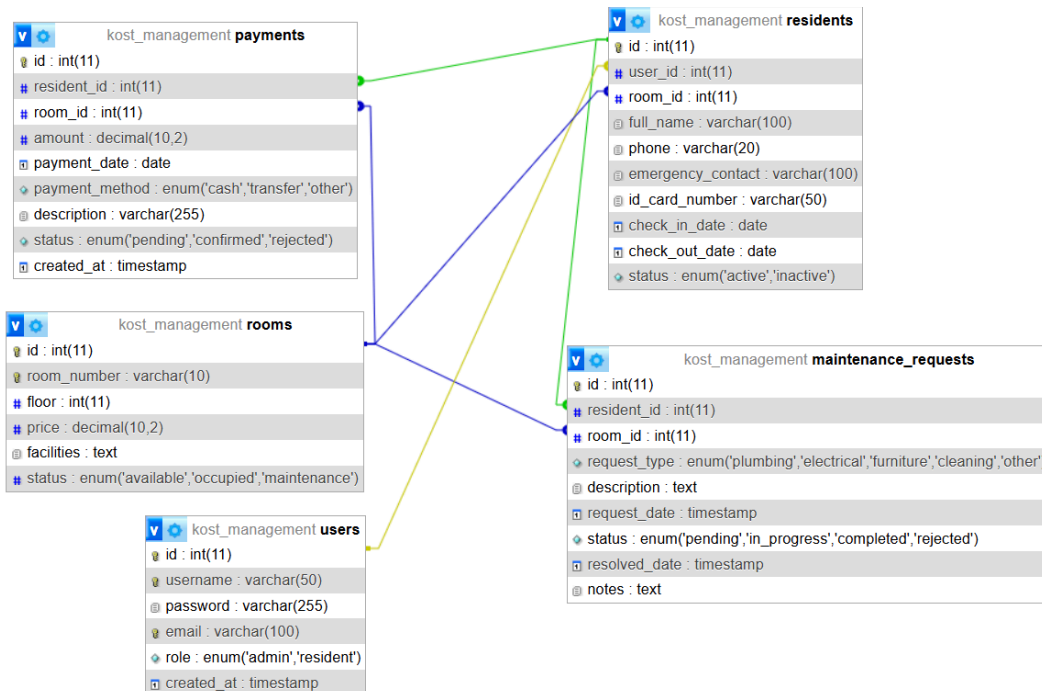
Maintenance\_Requests (Request\_ID PK, Resident\_ID FK, Room\_ID FK, Issue\_Description, Status, Request\_Date)

# CHAPTER D

## IMPLEMENTATION

### D.1 The Database

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> maintenance_requests	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	48.0 KiB	-
<input type="checkbox"/> payments	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	48.0 KiB	-
<input type="checkbox"/> residents	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	48.0 KiB	-
<input type="checkbox"/> rooms	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	32.0 KiB	-
<input type="checkbox"/> users	★ Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	48.0 KiB	-
<b>5 tables</b>	<b>Sum</b>	<b>10</b>	<b>InnoDB</b>	<b>utf8mb4_general_ci</b>	<b>224.0 KiB</b>	<b>0 B</b>



### D.2 The Data

#### - The data of user

	id	username	password	email	role	created_at
<input type="checkbox"/>	3	shant00	\$2y\$10\$P.ETp8XvS.JsFC3wxNSJ9jONTixiYelnRf4aHxDhAHlv...	shanty.kh2@gmail.com	resident	2025-04-15 15:27:45
<input type="checkbox"/>	7	admin2	\$2y\$10\$xDrl.Sw4pl.bUEOdbSCI6CegCOF4s/1dSVHx5WnG6PWZ...	admin@gmail.com	admin	2025-04-15 17:00:20
<input type="checkbox"/>	8	kale	\$2y\$10\$zXG4fXQhFEQmATgaDeZsb.yJF9weFpLMM5jNcPEYm6a...	kale@gmail.com	resident	2025-05-06 23:49:27
<input type="checkbox"/>	9	john	\$2y\$10\$ZlqcjpszyYfgSteu3pPGOUndQJZGwo.62dGkhYCKYo...	john@gmail.com	resident	2025-05-06 23:50:48

- **The data of rooms**

	id	room_number	floor	price	facilities	status
<input type="checkbox"/> Edit Copy Delete	1	101	1	1000000.00	Private room, Bed, Air Conditioner	occupied

- **The data of residents**

	id	user_id	room_id	full_name	phone	emergency_contact	id_card_number	check_in_date	check_out_date	status
<input type="checkbox"/> Edit Copy Delete	1	3	NULL	Shanty	081285672878		0012300	NULL	NULL	active
<input type="checkbox"/> Edit Copy Delete	2	8	1	Michelle	+6285262270970	085262270970	001202400017	2024-08-24	2025-08-24	active

- **The data of payments**

	id	resident_id	room_id	amount	payment_date	payment_method	payment_proof	description	status	created_at
<input type="checkbox"/> Edit Copy Delete	1	2	1	1000000.00	2025-04-15	transfer		Monthly rent for Room 23	confirmed	2025-04-15 17:15:35
<input type="checkbox"/> Edit Copy Delete	2	2	1	1000000.00	2025-04-28	cash		Monthly rent for Room 23	pending	2025-04-28 17:15:25

- **The data of maintenance\_request**

	id	resident_id	room_id	request_type	description	request_date	status	resolved_date	notes
<input type="checkbox"/> Edit Copy Delete	1	2	1	furniture	keropos	2025-04-15 17:16:11	completed	2025-04-15 12:19:32	

### D.3 The Queries

1. **Display All Active Residents**

```
SELECT full_name, phone, room_id
FROM residents
WHERE status = 'active';
```

full_name	phone	room_id
Shanty	081285672878	
Michelle	085262270970	

2. **Show Confirmed Payment History**

```
SELECT r.full_name, p.amount, p.payment_date, p.payment_method
FROM payments p
JOIN residents r ON p.resident_id = r.id
WHERE p.status = 'confirmed';
```

full_name	amount	payment_date	payment_method
Shanty	1000000.00	2025-04-15	transfer
Shanty	1000000.00	2025-04-26	cash
Michelle	1500000.00	2025-05-06	other
Michelle	1500000.00	2025-05-06	cash

### 3. List All Occupied Rooms

```
SELECT room_number, floor, price, status
FROM rooms
WHERE status = 'occupied';
```

room_number	floor	price	status
101	1	1000000.00	occupied
23	2	1500000.00	occupied

### 4. View Ongoing Maintenance Requests

```
SELECT r.full_name, m.request_type, m.description, m.status
FROM maintenance_requests m
JOIN residents r ON m.resident_id = r.id
WHERE m.status IN ('pending', 'in_progress');
```

full_name	request_type	description	status
Michelle	electrical	bzzttt	pending
Michelle	cleaning	pintunya berjamur	pending

### 5. View Completed Maintenance Requests

```
SELECT r.full_name, m.request_type, m.description, m.status
FROM maintenance_requests m
JOIN residents r ON m.resident_id = r.id
WHERE m.status IN ('completed');
```

full_name	request_type	description	status
Shanty	furniture	AC bocor	completed
Michelle	electrical	lampunya jedag jedug	completed
Michelle	other	air ac netes	completed

**6. Calculate Total Confirmed Payments per Resident**

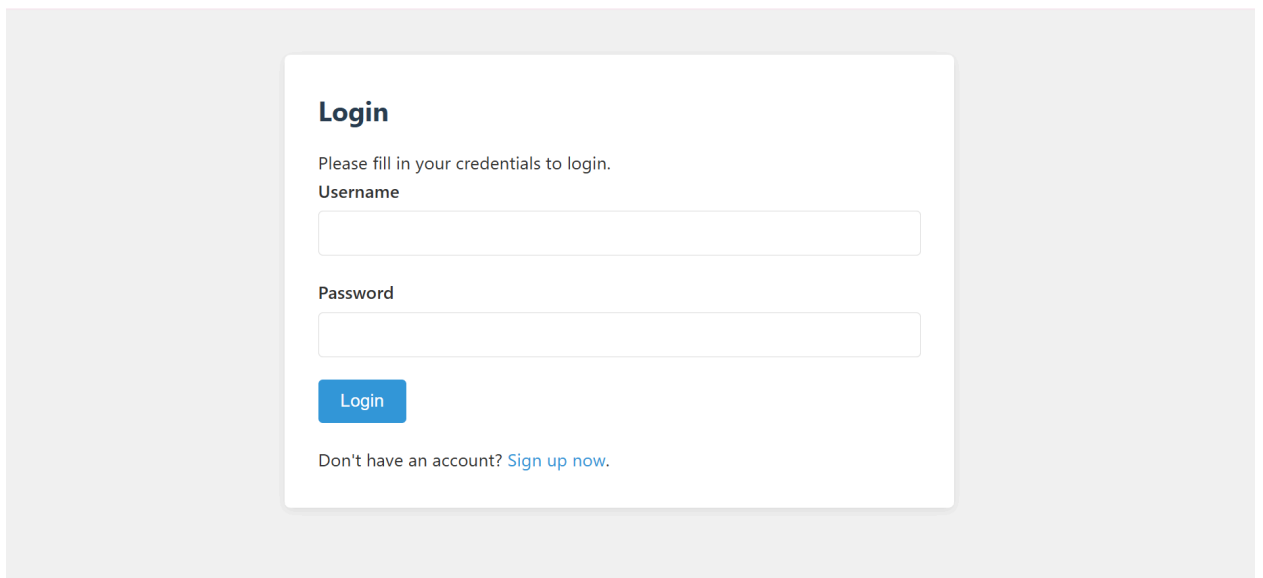
```
SELECT r.full_name, SUM(p.amount) AS total_paid
FROM payments p
JOIN residents r ON p.resident_id =r.id
WHERE p.status = 'confirmed' GROUP BY r.full_name;
```

full_name	total_paid
Michelle	3000000.00
Shanty	2000000.00

## CHAPTER E

### CREATE PHP BASED DATABASE APPLICATION

- This is the login page



**Login**

Please fill in your credentials to login.

Username

Password

Login

Don't have an account? [Sign up now.](#)

- This is the register page

### Register

Please fill this form to create an account.

Username

Email

Password

Confirm Password

Full Name

Phone Number

ID Card Number

Emergency Contact (Optional)

Already have an account? [Login here.](#)

- This is the dashboard for the user

## Kost Management System

Dashboard Payments Maintenance Logout

### Welcome, Michelle

#### Resident Information

**Full Name:** Michelle  
**Phone:** +6285262270970  
**ID Card:** 001202400017  
**Emergency Contact:** 085262270970  
**Check-in Date:** 2024-08-24  
**Check-out Date:** 2025-08-24

#### Room Information

**Room Number:** 23  
**Floor:** 2  
**Monthly Rent:** Rp 1.000.000

#### Payment Information

**Last Payment:** 2025-04-28  
**Amount:** Rp 1.000.000  
**Status:** Pending

#### Recent Maintenance Requests

Type	Description	Date	Status
Furniture	keropos	2025-04-15 17:16:11	Completed

- This is the payment page

### Payments

**Make a Payment**

Your monthly rent is **Rp 1.000.000**.

[Make a Payment](#)

**Payment History**

ID	Room	Amount	Payment Date	Method	Description	Status
3	101	Rp 1.000.000	2025-05-06	Transfer	Monthly rent for Room 101	Pending
4	101	Rp 1.000.000	2025-05-06	Transfer	Monthly rent for Room 101	Pending
2	101	Rp 1.000.000	2025-04-26	Cash	Monthly rent for Room 101	Confirmed
1	101	Rp 1.000.000	2025-04-15	Transfer	Monthly rent for Room 101	Confirmed

- This is the page if we want make a payment

### Kost Management System

Dashboard Payments Maintenance Logout

#### Submit Payment

Room

Amount (Rp)

Default amount is your monthly rent.

Payment Date

Payment Method

Description

Eg. "Monthly rent for April 2023"

[Submit Payment](#) [Cancel](#)

- This is the message if we already make our payment

## Payments

Payment submitted successfully. It will be reviewed by the administrator.

- This is the maintenance page

**Kost Management System** [Dashboard](#) [Payments](#) [Maintenance](#) [Logout](#)

## Maintenance Requests

Your Maintenance Requests [New Request](#)

ID	Room	Type	Description	Request Date	Status	Resolved Date	Notes
1	23	Furniture	keropos	2025-04-15 17:16:11	Completed	2025-04-15 12:19:32	N/A

- This is if we wanna make a maintenance request

**Kost Management System** [Dashboard](#) [Payments](#) [Maintenance](#) [Logout](#)

### Submit New Maintenance Request

Request Type

Description

Please describe the issue in detail and provide any relevant information.

[Submit Request](#) [Cancel](#)

- **This is the type of request**

**Request Type**

Select request type ▼

Select request type

- Plumbing
- Electrical
- Furniture
- Cleaning
- Other