CSE302: Database Systems (Section No. 10) [FALL - 2025]

Project Report

EWU Hostel Management

Project URL

Ashfaq Ahmed Abir: LINK

Peu Mondol Sharna: LINK

Mahima Begum Mitu:

Submitted by:

Student ID	**Student Name**	**Contribution**
2024-1-60-088	Ashfaq Ahmed Abir	40%
2024-1-60-170	Mahima Begum Mitu	30%
2024-1-60-276	Peu Mondol Sharna	30%

1. Project Description

The **Hostel Management System** is a web-based application developed to make hostel management easier and more organized. This project simulates a real-life hostel system where students live, take meals, pay expenses, and book rooms.
The main purpose of this system is to reduce manual work and manage hostel activities digitally. Instead of using paper records or spreadsheets, all information is stored and handled through the system in a structured way.

This system handles different types of data, such as:

- Member information (name, email, role, contact details)
- Room and seat information
- Meal details and payments
- Expense and deposit records

The system supports different users like **Admin**, **Manager**, and **Member**, and each user can only access the features related to their role. Overall, this project helps in managing hostel operations efficiently, accurately, and securely in a real-world hostel environment.

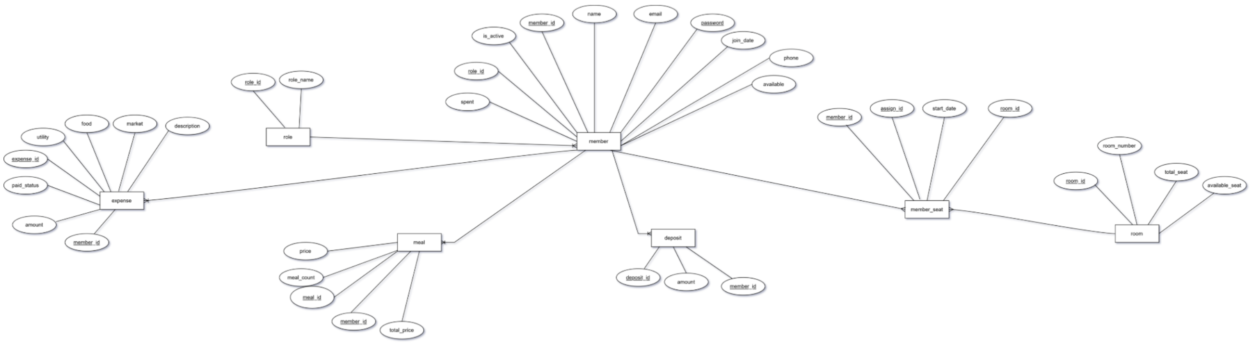
2. Key Features

The Hostel Management System provides several important features to manage hostel activities smoothly. These features are designed to be simple and user-friendly for all types of users.

- User login system with role-based access (Admin, Manager, Member)
- Admin can add, edit, and delete members
- Email duplication is prevented to avoid login issues
- Secure password storage using encryption
- Room and seat management with automatic seat availability updates
- Members can book only one seat at a time
- Meal management system with meal count and payment tracking
- Members can pay for meals one by one
- Automatic update of total meal price after each payment
- Expense management for utilities, food, and market costs
- Admin and Manager can add and update expenses
- Deposit management to track member payments
- Members can view their own expenses, deposits, and meal details
- Clean and easy-to-use interface for better user experience

3. Database Design

3.1 E–R Model



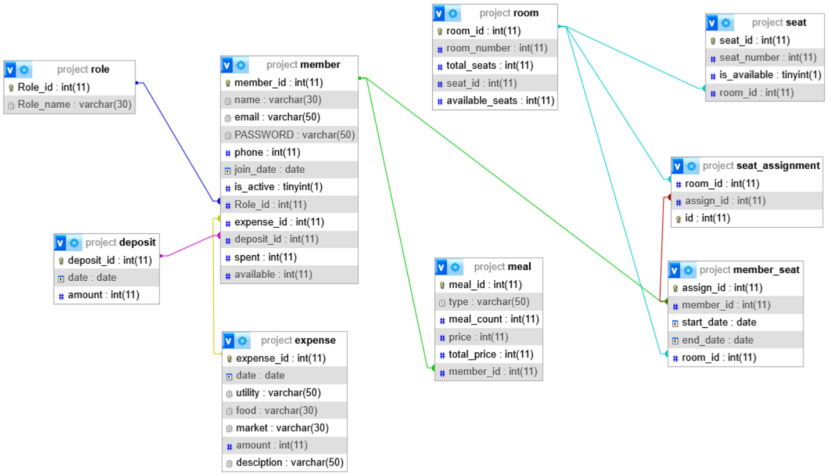
The Entity–Relationship (E–R) model represents the logical structure of the Hostel Management System. It shows the main entities involved in the system, their attributes, and the relationships between them.

The key entities in the system include **Role**, **Member**, **Room**, **Member_Seat**, **Meal**, **Expense**, and **Deposit**. Each entity represents a real-world object in a hostel environment.

- A **Role** entity defines different types of users such as Admin, Manager, and Member.
- A **Member** entity stores user-related information like name, email, password, phone number, role, and account balances.
- A **Room** entity contains room details such as room number, total seats, and available seats.
- The **Member_Seat** entity connects members with rooms and ensures that one member can be assigned only one seat.
- The **Meal** entity stores meal-related information including meal type, meal count, price, and total price for a member.
- The **Expense** entity records hostel expenses such as utility, food, and market costs.
- The **Deposit** entity stores deposit information made by members.

The relationships between entities are mostly **one-to-many**. For example, one role can be assigned to many members, one room can be associated with multiple seat assignments, and one member can have multiple meal, expense, and deposit records. Primary keys and foreign keys are used to maintain data integrity.

3.2 Relational Data Model



The relational data model is created by converting the E–R model into relational tables. Each entity from the E–R diagram is transformed into a table, and relationships are represented using foreign keys.

- The **Role** table uses `role_id` as the primary key.
- The **Member** table uses `member_id` as the primary key and includes `role_id` as a foreign key referencing the Role table.
- The **Room** table uses `room_id` as the primary key.
- The **Member_Seat** table uses `assign_id` as the primary key and contains `member_id` and `room_id` as foreign keys to link members and rooms.
- The **Meal** table uses `meal_id` as the primary key and includes `member_id` as a foreign key to associate meals with members.
- The **Expense** table uses `expense_id` as the primary key and includes `member_id` as a foreign key.
- The **Deposit** table uses `deposit_id` as the primary key and includes `member_id` as a foreign key.

4. Tools & Technologies Used

This project was developed using commonly available and easy-to-use tools and technologies suitable for academic and real-world applications.

Software and Tools Used:

- **DBMS:** MySQL (using phpMyAdmin for database management)
- **Server Environment:** XAMPP (Apache server with MySQL)

- **Programming Editor:** Visual Studio Code
- **Web Browser:** Google Chrome (for testing and debugging)
- **Operating System:** Windows

Programming and Query Languages:

- **PHP:** Used for server-side logic and handling user requests
- **SQL:** Used for database creation, table design, and data manipulation
- **CSS & Bootstrap:** Used for styling and responsive user interface design

These tools together helped in building a complete, functional, and user-friendly Hostel Management System.

MySQL Queries

```
DROP DATABASE IF EXISTS PROJECT;

CREATE DATABASE PROJECT;
USE PROJECT;

CREATE TABLE role (
   role_id INT AUTO_INCREMENT PRIMARY KEY,
   role_name VARCHAR(30)
);

INSERT INTO role (role_name) VALUES
('Admin'),
('Member'),
('Manager');

CREATE TABLE member (
   member_id INT AUTO_INCREMENT PRIMARY KEY,
   name VARCHAR(30),
   email VARCHAR(50),
   password VARCHAR(50),
   phone VARCHAR(20),
   join_date DATE,
   is_active BOOLEAN DEFAULT 1,
   role_id INT,
   spent INT DEFAULT 0,
   available INT DEFAULT 0,
   FOREIGN KEY(role_id) REFERENCES role(role_id)
);

ALTER TABLE member ADD UNIQUE (email);
```

```
UPDATE member SET password = MD5(password);

INSERT INTO member (name, email, password, phone, join_date, role_id, spent, available)
VALUES ('Admin', 'admin@admin.com', '123', '01700000000', CURDATE(), 1, 0, 0);

CREATE TABLE room (
   room_id INT AUTO_INCREMENT PRIMARY KEY,
   room_number INT,
   total_seats INT,
   available_seats INT
);

CREATE TABLE member_seat (
   assign_id INT AUTO_INCREMENT PRIMARY KEY,
   member_id INT,
   room_id INT,
   start_date DATE,
   FOREIGN KEY(member_id) REFERENCES member(member_id),
   FOREIGN KEY(room_id) REFERENCES room(room_id)
);

CREATE TABLE expense (
   expense_id INT AUTO_INCREMENT PRIMARY KEY,
   utility VARCHAR(50),
   food VARCHAR(30),
   market VARCHAR(30),
   amount INT,
   description VARCHAR(100),
   member_id INT,
   paid_status VARCHAR(10) DEFAULT 'No',
   FOREIGN KEY(member_id) REFERENCES member(member_id)
);

CREATE TABLE deposit (
   deposit_id INT AUTO_INCREMENT PRIMARY KEY,
   amount INT,
   member_id INT,
   FOREIGN KEY(member_id) REFERENCES member(member_id)
);

CREATE TABLE meal (
   meal_id INT AUTO_INCREMENT PRIMARY KEY,
   type VARCHAR(50),
   meal_count INT,
   price INT,
   total_price INT,
   member_id INT,
   FOREIGN KEY(member_id) REFERENCES member(member_id)
);
```

5. Role Assignment

The Hostel Management System uses a role-based access control approach to ensure security and proper system usage. Each user is assigned a specific role, and system access is controlled based on that role.

Admin

- Has full control over the system.
- Can add, edit, and delete members.
- Can assign or change user roles.
- Can manage rooms, seats, meals, expenses, and deposits.
- Can view all system data and perform all actions.

Manager

- Assists the admin in managing hostel operations.
- Can add and update rooms, meals, expenses, and deposits.
- Can edit member information except changing user roles.
- Cannot delete members or access admin-only settings.

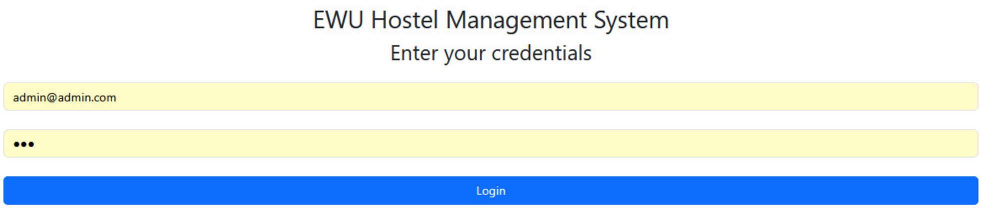
Member

- Represents regular hostel residents.
- Can view personal information, meals, expenses, and deposits.
- Can book one seat if available.
- Can pay for meals and view payment status.
- Cannot add, edit, or delete system data.

This role-based design ensures that users can only access features relevant to their responsibilities, improving both system security and usability.

6. GUI Screenshot

Login Page



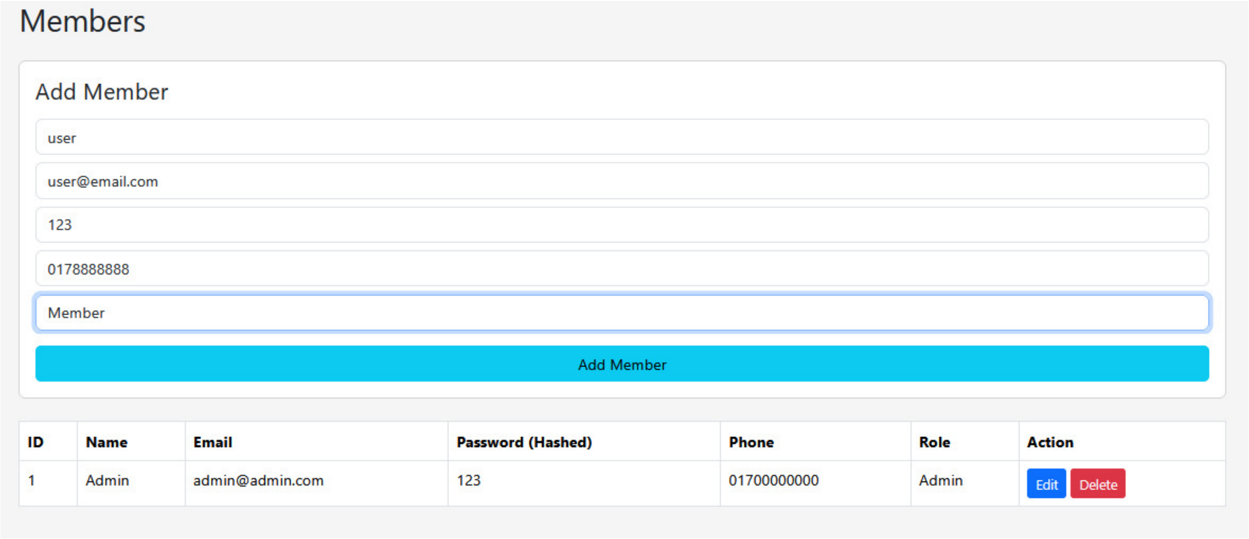
EWU Hostel Management System

Enter your credentials

admin@admin.com

•••

Members Page



Members

Add Member

user

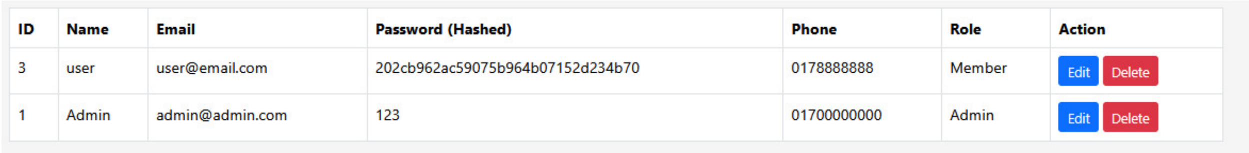
user@email.com

123

0178888888

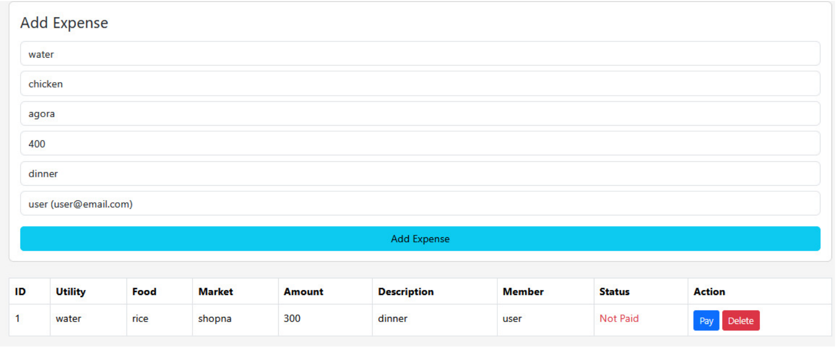
Member

ID	Name	Email	Password (Hashed)	Phone	Role	Action
1	Admin	admin@admin.com	123	01700000000	Admin	Edit Delete



ID	Name	Email	Password (Hashed)	Phone	Role	Action
3	user	user@email.com	202cb962ac59075b964b07152d234b70	0178888888	Member	Edit Delete
1	Admin	admin@admin.com	123	01700000000	Admin	Edit Delete

Expense Page



Add Expense

water

chicken

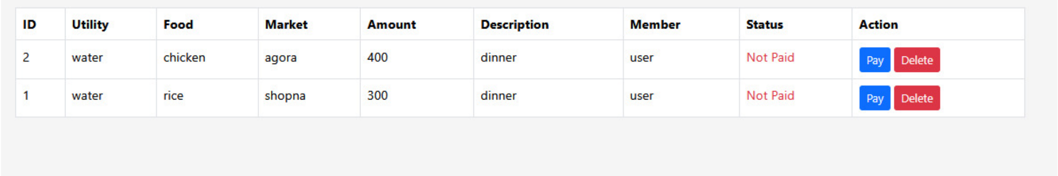
agora

400

dinner

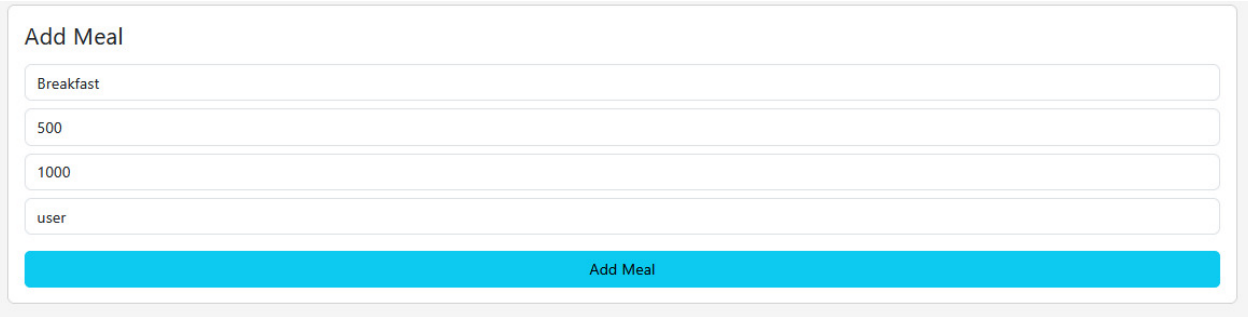
user (user@email.com)

ID	Utility	Food	Market	Amount	Description	Member	Status	Action
1	water	rice	shopna	300	dinner	user	Not Paid	Pay Delete



ID	Utility	Food	Market	Amount	Description	Member	Status	Action
2	water	chicken	agora	400	dinner	user	Not Paid	Pay Delete
1	water	rice	shopna	300	dinner	user	Not Paid	Pay Delete

Meal Page



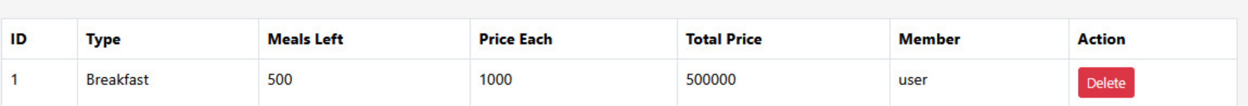
Add Meal

Breakfast

500

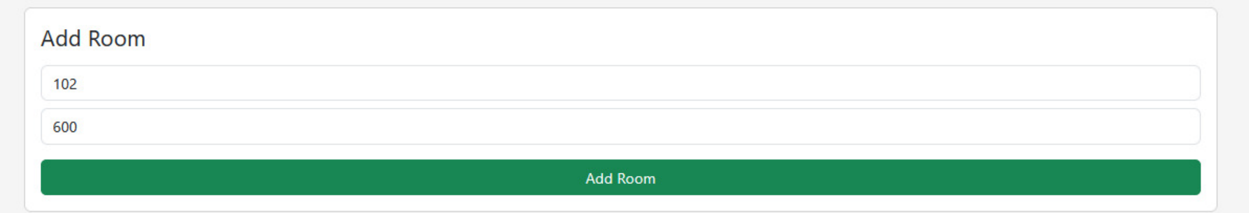
1000

user



ID	Type	Meals Left	Price Each	Total Price	Member	Action
1	Breakfast	500	1000	500000	user	Delete

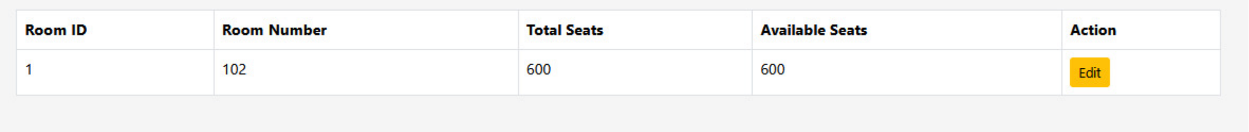
Room page



Add Room

102

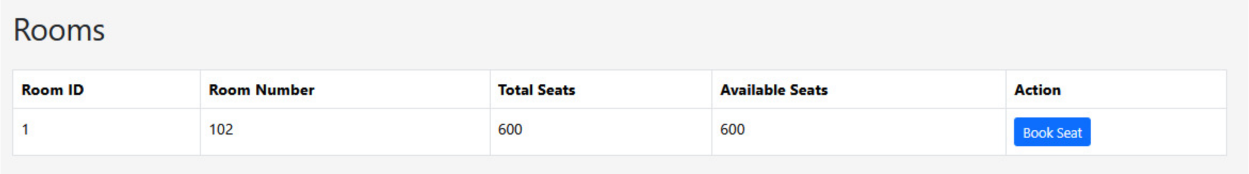
600



Room ID	Room Number	Total Seats	Available Seats	Action
1	102	600	600	Edit

Room page (from user end)

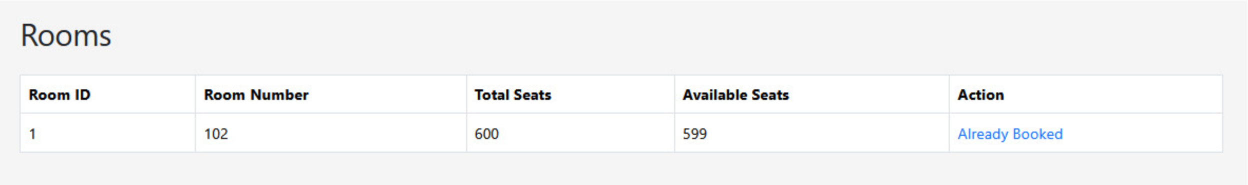
- Before Booking



Rooms

Room ID	Room Number	Total Seats	Available Seats	Action
1	102	600	600	Book Seat

- After Booking



Rooms

Room ID	Room Number	Total Seats	Available Seats	Action
1	102	600	599	Already Booked

7. Conclusion

This project helped us gain practical experience in designing and developing a complete web-based system using **PHP and MySQL**. Through the Hostel Management System, we learned how to implement role-based access control, manage relational databases, and connect frontend interfaces with backend logic. The project also improved our understanding of real-world system requirements such as data consistency, automation, and security.

During development, we faced several challenges. Designing the database relationships correctly, especially for room-seat allocation and meal tracking, required careful planning. Handling role-based permissions and preventing data conflicts such as duplicate emails or multiple seat bookings was also challenging. Additionally, implementing automatic updates for balances, seat availability, and meal counts needed thorough testing to ensure accuracy.

For future improvements, the system can be enhanced by using stronger password security methods such as `password_hash()` instead of MD5. Adding features like payment gateway integration, notification systems, and detailed reports would make the system more practical. The user interface can also be improved further for better usability. Overall, this project provided valuable hands-on experience and can be extended into a more advanced hostel management solution.

8. References

- MySQL Tutorial (w3schools.)
- DBMS Course By Gate Smashers
- Course materials provided by Faculty (Google Classroom)