House Rental Management System

Dip Karmakar (210201010), Prapti Nandi (210201006), Md. Abdul High Sharif (210201008),

Rownak E Ikram (210201035)

Course Code: CSE 2206 **Course Title:** Database Management Systems Sessional

Department of Computer Science and Engineering, Bangladesh Army University of Science and Technology (BAUST)

1. Introduction

HOUSE RENTAL MANAGEMENT SYSTEM project represents a concerted effort to address the complexities inherent in property rental management. As the demand for housing continues to surge, both landlords and tenants are faced with an array of intricate processes, including property listings, tenant screening, rent collection, and maintenance tracking, Our project endeavors to streamline these operations, enhance transparency, and foster seamless communication between all stakeholders involved.

2. Problem Definition

Manual Processes:

Traditional rental processes are often manual, leading to inefficiencies and time-consuming tasks for both landlords and tenants.

Lack of Transparency:

Existing systems may lack transparency, causing confusion and misunderstandings between landlords and tenants.

Security Concerns:

Data security is a concern in some systems, potentially putting sensitive information at risk.

Limited Accessibility:

Some rental management systems may not be easily accessible or user-friendly, creating barriers for users, especially those less familiar with technology.

Identifying and addressing these issues is crucial for the successful development and implementation of our House Rental Management System.

3. Available Applications

Projects	Features	Limitations
1. Zillow	Use agencies for better coverage.	Missing Local Listings.
	Missing listing network	Limited property options.
2. Trulia	Maintenance & Update.	Lack of Real-Time Updates.
	Comprehensive pricing.	Lack of accurate pricing
3. Rent	Rent negotiation platform	Limited Budget Options
	Enhanced search filter	No area based search option.
4. Airbnb	Detailed description about property.	Not used accurate property images.
	Regular verification & user rating.	Outdated listing.
5.Craigslist	Enhanced data security.	Hassle to fill-up lots of information.

4. Stakeholders

Landlords:

Primary users entrusting the system to manage and streamline their property rental processes.

Tenants:

End-users benefiting from the user-friendly features and improved rental experience provided by the system.

Development Team:

The team responsible for designing, coding, and maintaining the House Rental Management System.

Customer Support:

The support team ensuring assistance and guidance for both landlords and tenants in using the system.

Management:

Decision-makers overseeing the project's progress, providing resources, and ensuring alignment with organizational goals.

5. Issues Encountered

As PHP is a new language for us, first we had to learn about the fundamentals of PHP and then we took a course (tutorial) for PHP and also learn MYSQL basic for completing backend of this project. Once we got our fundamentals right, we started implementing. As we go further into the project we faced multiple errors and had to take help from the W3 school and mysqltutorial.org to solve this problems. Some of the problems had to do with managing libraries and find the specific library to use. So to find the solution we took some tutorials and learn about the specific library and used it to work for our project. We face some problem during form validation, we use sanitization to validate the form after researching some websites. So once we correctly configure the hardhat config we successfully deploy our smart contract.

6. System Architecture

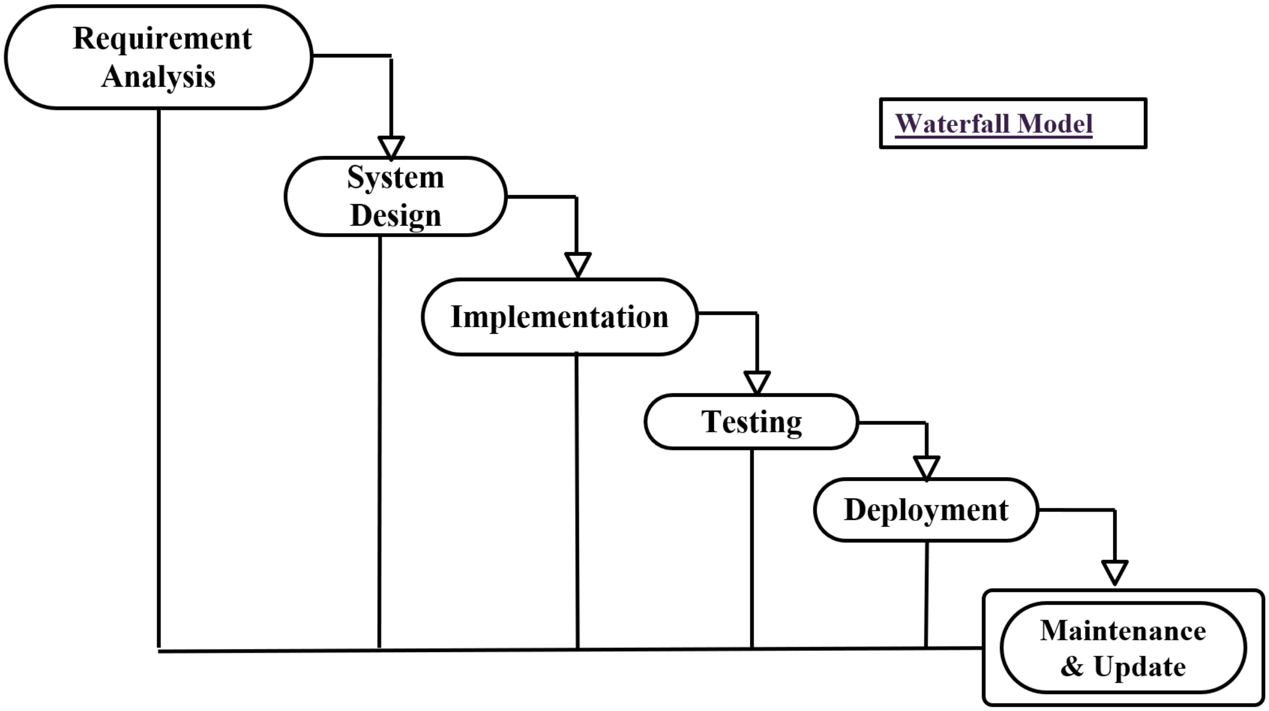


Fig:1 Methodology diagram of our project

In developing our House Rental Management System, we embraced the Waterfall Model, a sequential approach guiding us through key phases. Initially, we meticulously gathered stakeholder requirements, establishing a solid foundation. Subsequently, the system's architecture took shape in the design phase, providing a blueprint for its structure. The development team then translated these designs into functional code during the implementation phase. Rigorous testing followed, addressing any bugs or issues to ensure a robust system.

Upon successful testing, the system was deployed, marking its transition to active use. Post-deployment, a structured maintenance and support system were established to address user feedback and ensure ongoing smooth operation. The Waterfall Model's step-by-step progression ensured a comprehensive understanding of user needs, resulting in the creation of a reliable and user-friendly House Rental Management System.

7. Database Design

ER Diagram

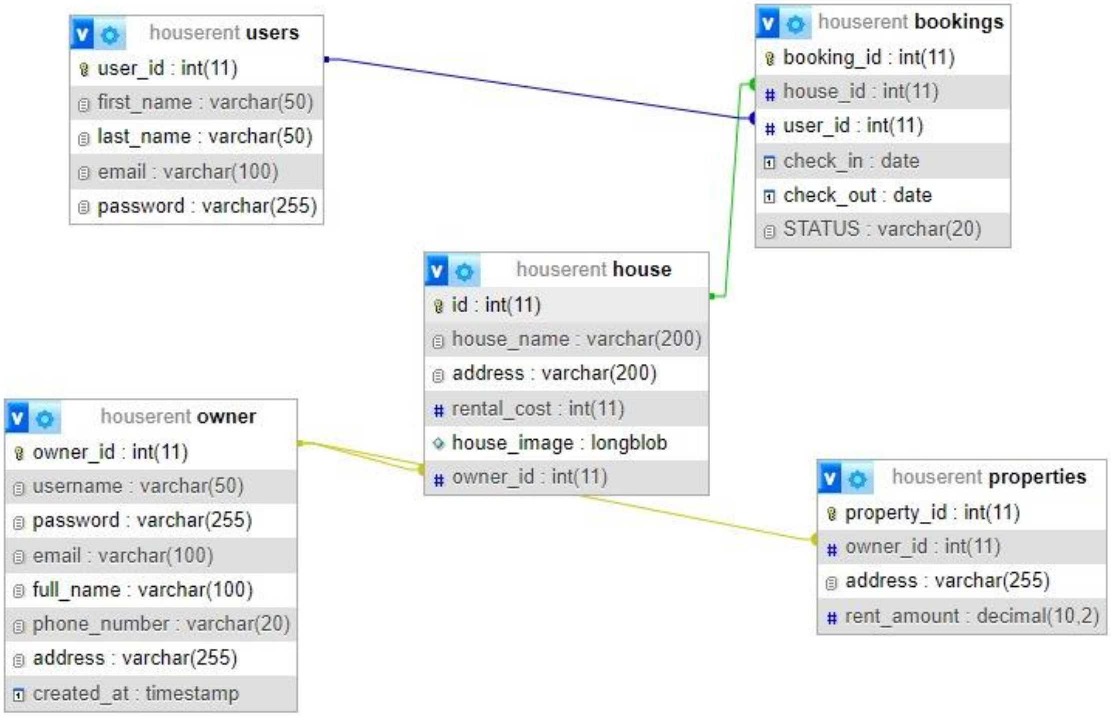


Fig:2 Entity relationship diagram of our project

Database Schema

Database: ‘houserent’ for the web-page

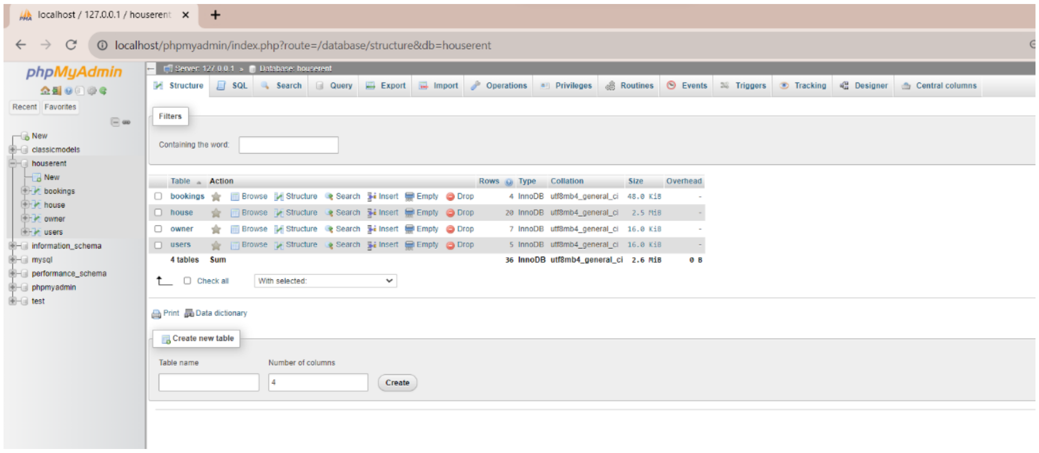
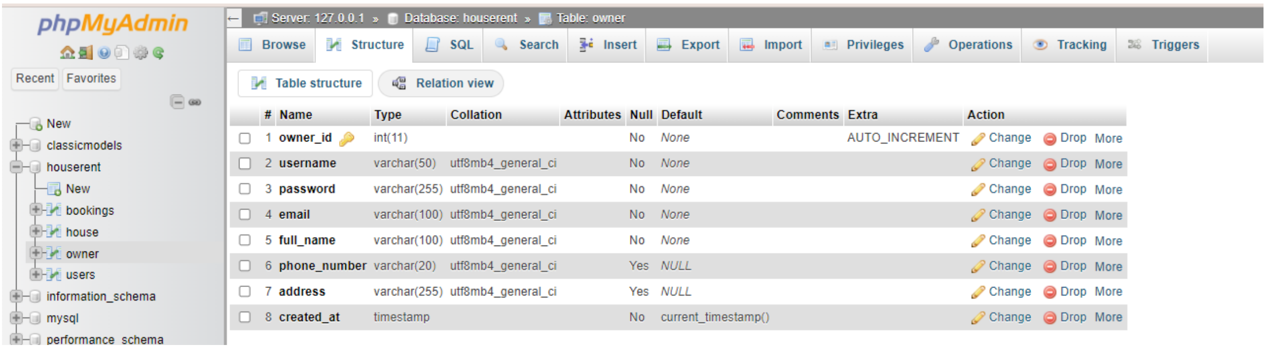

localhost / 127.0.0.1 / houserent
localhost/phpmyadmin/index.php?route=/database/structure&db=houserent
phpMyAdmin
Recent Favorites
New
classicmodels
houserent
New
bookings
house
owner
users
information_schema
mysql
performance_schema
phpmyadmin
test
Server: 127.0.0.1 » Database: houserent
Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers Tracking Designer Central columns
Filters
Containing the word:
Check all
With selected:
Print Data dictionary
Create new table
Table name
Number of columns
4
Create

Table	Action	Rows	Type	Collation	Size	Overhead
bookings	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	48.0 KiB	-
house	Browse Structure Search Insert Empty Drop	20	InnoDB	utf8mb4_general_ci	2.5 MiB	-
owner	Browse Structure Search Insert Empty Drop	7	InnoDB	utf8mb4_general_ci	16.0 KiB	-
users	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	16.0 KiB	-
4 tables	Sum	36	InnoDB	utf8mb4_general_ci	2.6 MiB	0 B

Fig:3

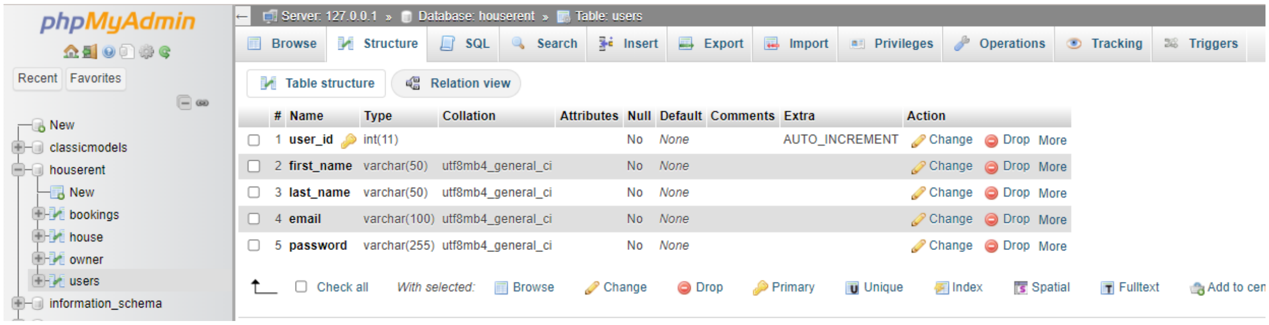
Table: “owner” for owner-reg & owner login



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	owner_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
☐	2	username	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
☐	3	password	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
☐	4	email	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
☐	5	full_name	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
☐	6	phone_number	varchar(20)	utf8mb4_general_ci		Yes	NULL			Change Drop More
☐	7	address	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
☐	8	created_at	timestamp			No	current_timestamp()			Change Drop More

Fig: 3.1

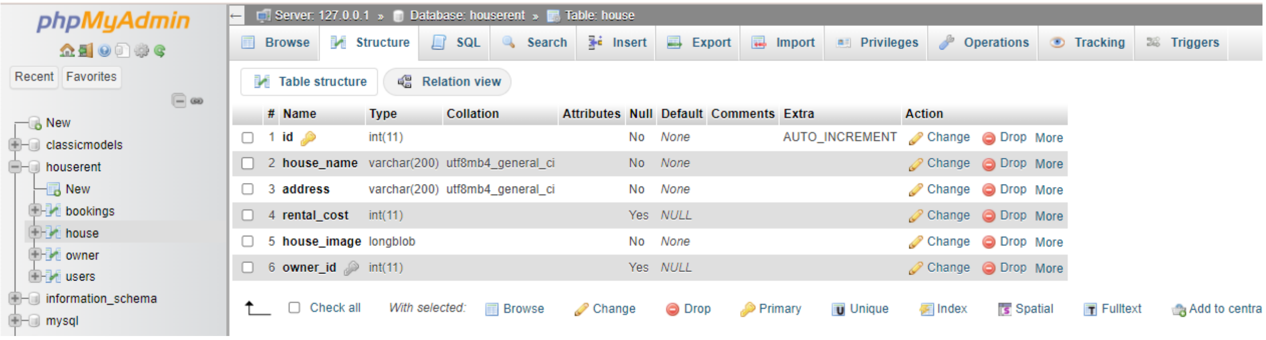
Table: “user” for user information



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	user_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
☐	2	first_name	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
☐	3	last_name	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
☐	4	email	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
☐	5	password	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More

Fig:3.2

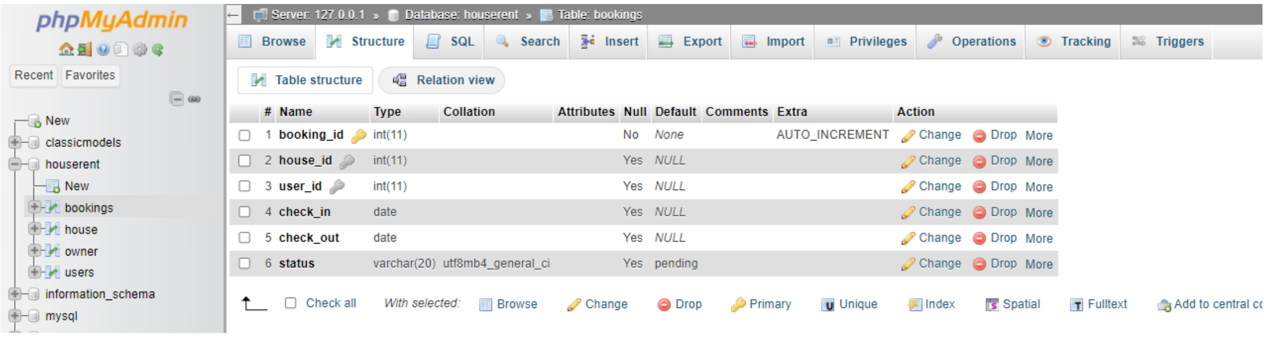
Table: “house” for store house information



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
☐	2	house_name	varchar(200)	utf8mb4_general_ci		No	None			Change Drop More
☐	3	address	varchar(200)	utf8mb4_general_ci		No	None			Change Drop More
☐	4	rental_cost	int(11)			Yes	NULL			Change Drop More
☐	5	house_image	longblob			No	None			Change Drop More
☐	6	owner_id	int(11)			Yes	NULL			Change Drop More

Fig:3.3

Table: “bookings” for book house



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	booking_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	house_id	int(11)			Yes	NULL			Change Drop More
3	user_id	int(11)			Yes	NULL			Change Drop More
4	check_in	date			Yes	NULL			Change Drop More
5	check_out	date			Yes	NULL			Change Drop More
6	status	varchar(20)	utf8mb4_general_ci		Yes	pending			Change Drop More

Fig:3.4

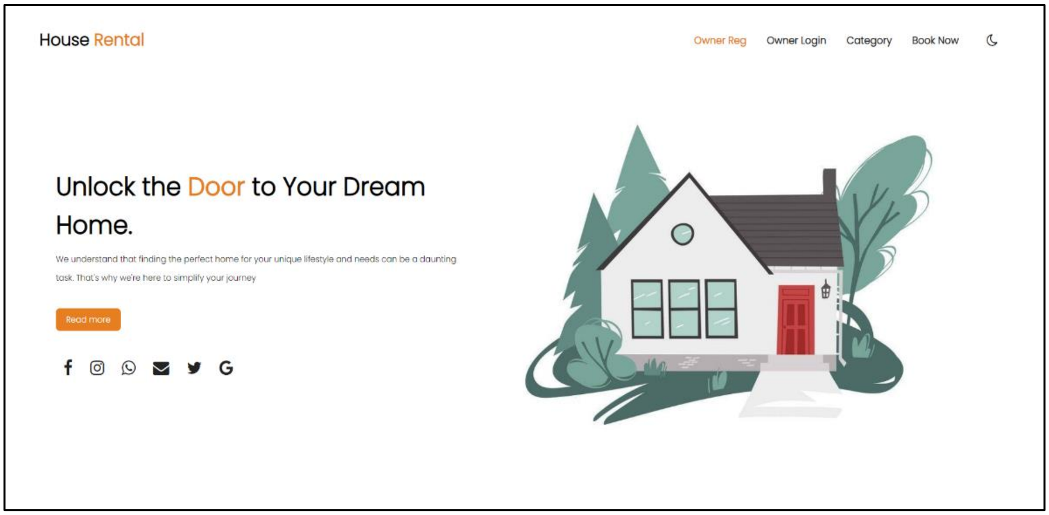
8. Tools and Technologies:

All the Languages we have used in this project
PHP, HTML, CSS, Bootstrap 5, JavaScript, MYSQL

Software we have used:
XAMPP, Visual Studio Code

9. Implementation

Home page: Navigation bar, Owner reg, Owner login, Category, Book now

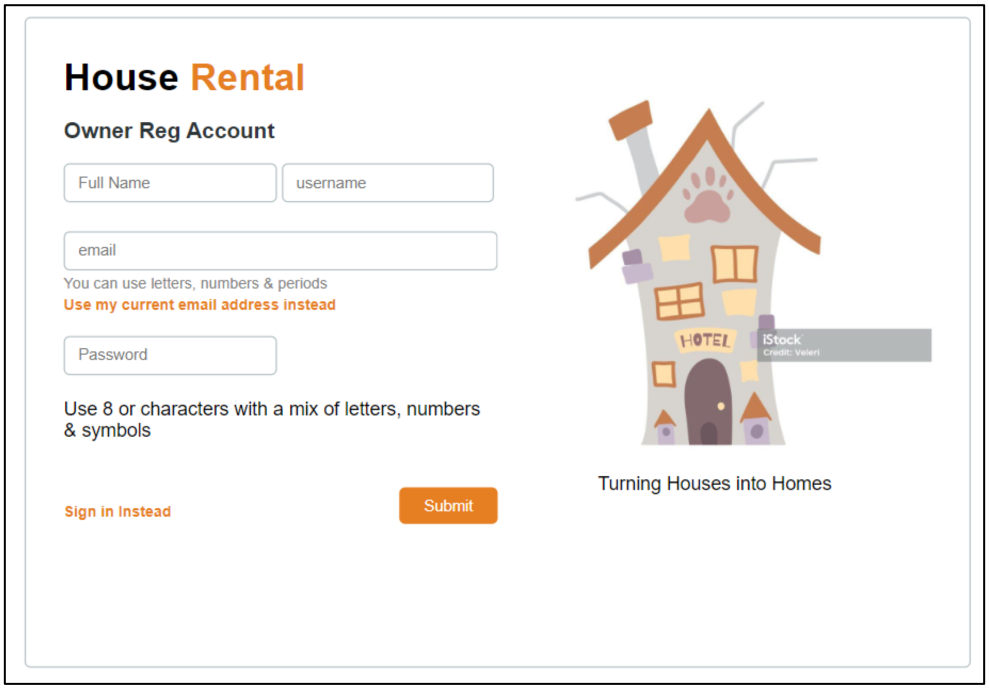


Unlock the Door to Your Dream Home.

We understand that finding the perfect home for your unique lifestyle and needs can be a daunting task. That's why we're here to simplify your journey

Fig:4

Owner Registration: Insert all information for owner



House Rental

Owner Reg Account

You can use letters, numbers & periods

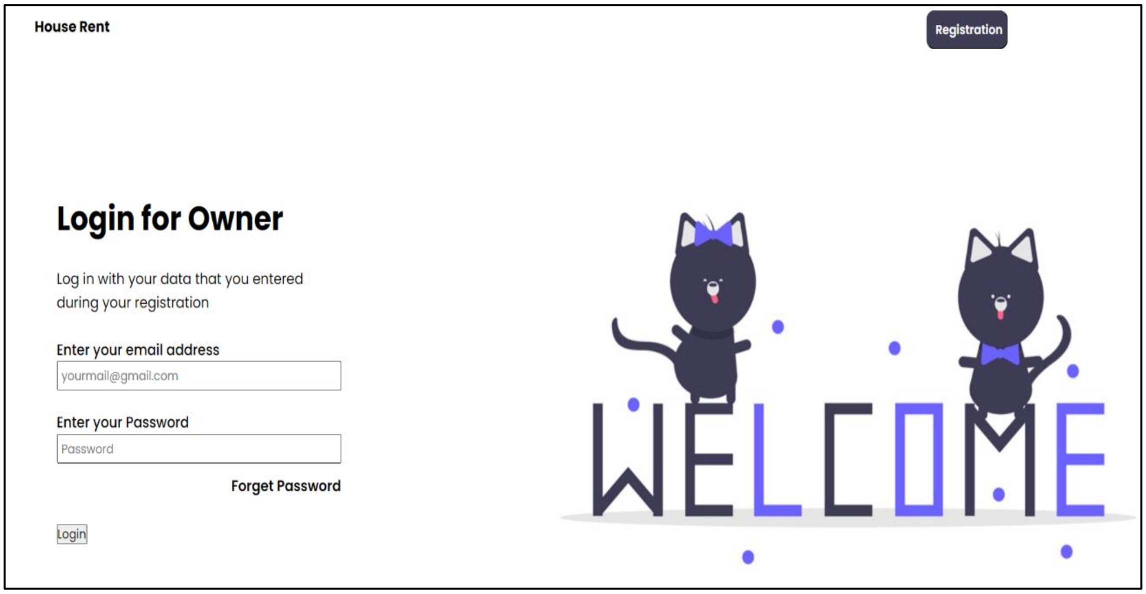
Use my current email address instead

Use 8 or characters with a mix of letters, numbers & symbols

Turning Houses into Homes

Fig:4.1

Owner login: Insert email & password for owner



Login for Owner

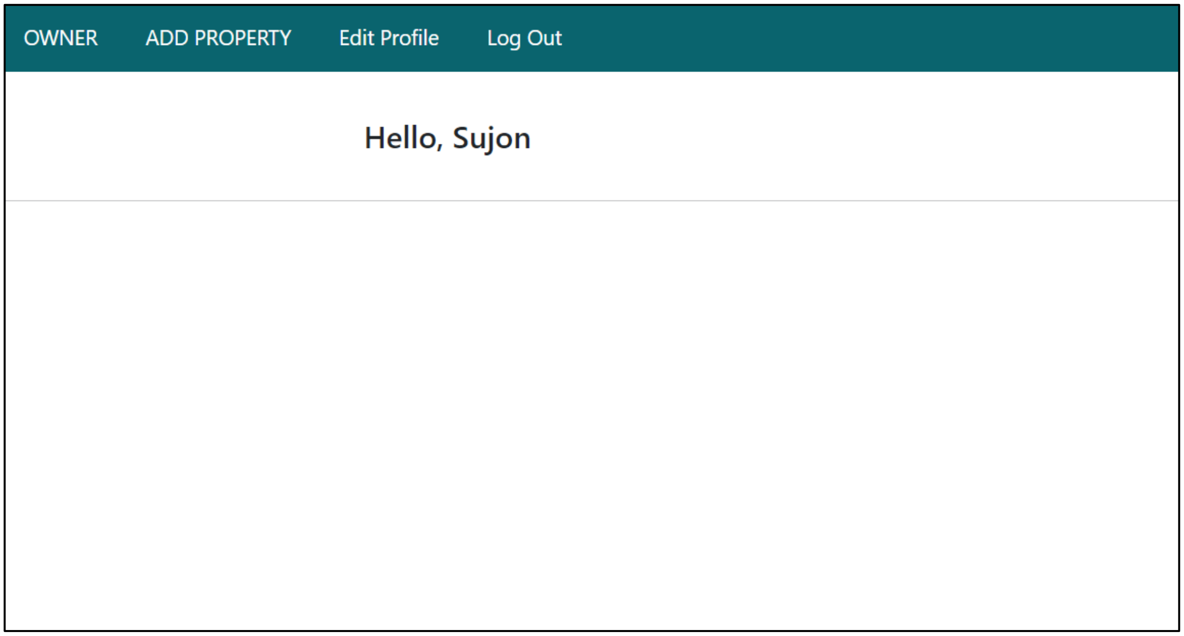
Log in with your data that you entered during your registration

Enter your email address

Enter your Password

Fig:4.2

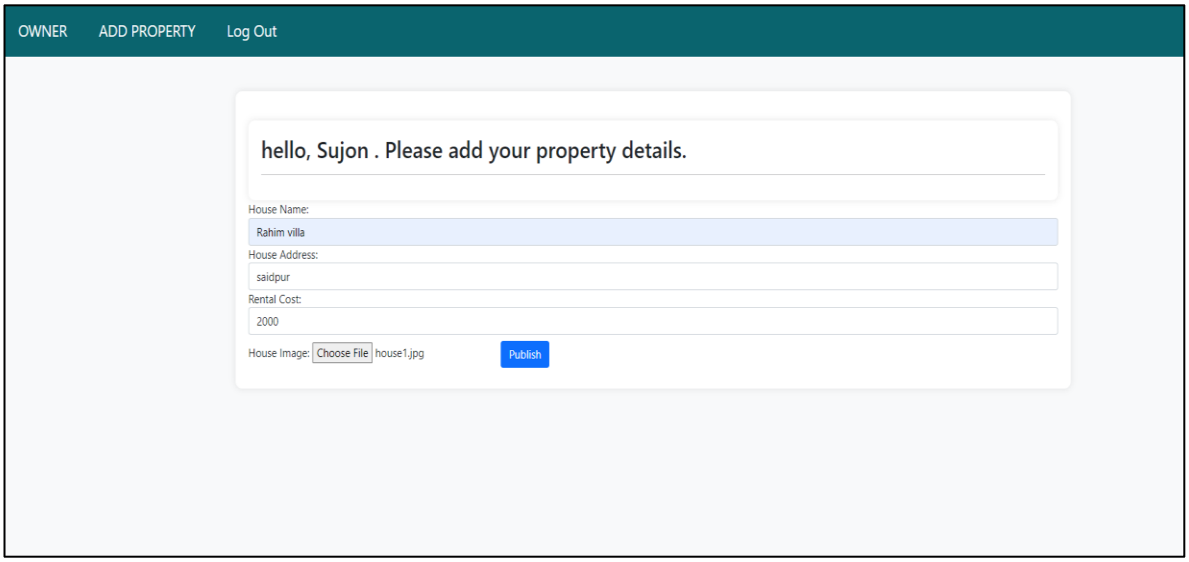
Owner panel: Add property, Owner profile, Edit profile, Log out



Hello, Sujon

Fig:5

Add property: Insert house information



hello, Sujon . Please add your property details.

House Name: Rahim villa

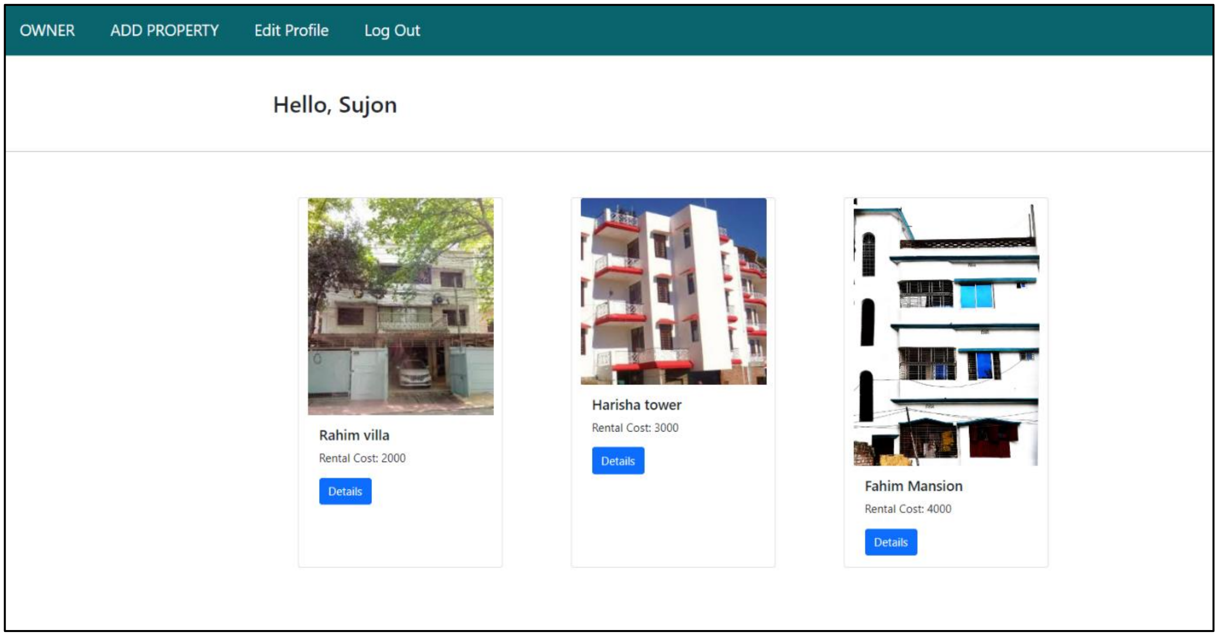
House Address: saidpur

Rental Cost: 2000

House Image: house1.jpg

Fig:5.1

Owner profile: House list, details



Hello, Sujon

Rahim villa

Rental Cost: 2000

Harisha tower

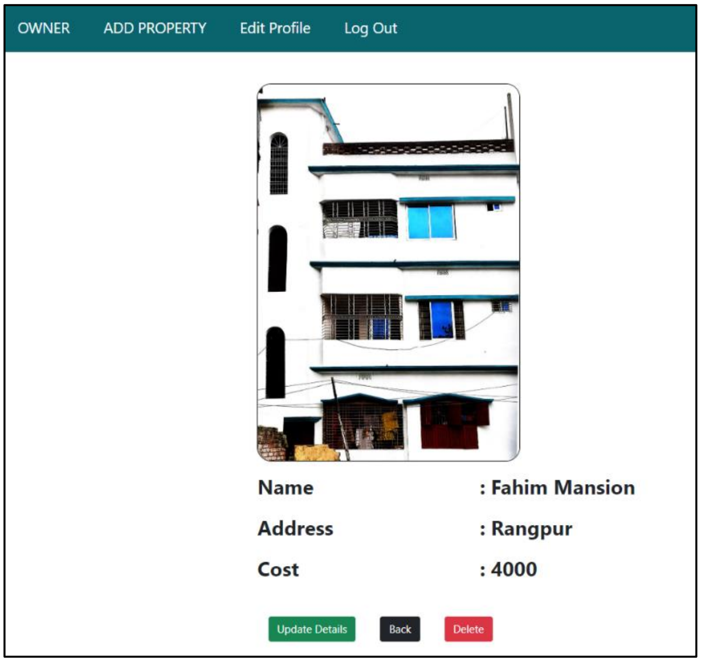
Rental Cost: 3000

Fahim Mansion

Rental Cost: 4000

Fig:5.2

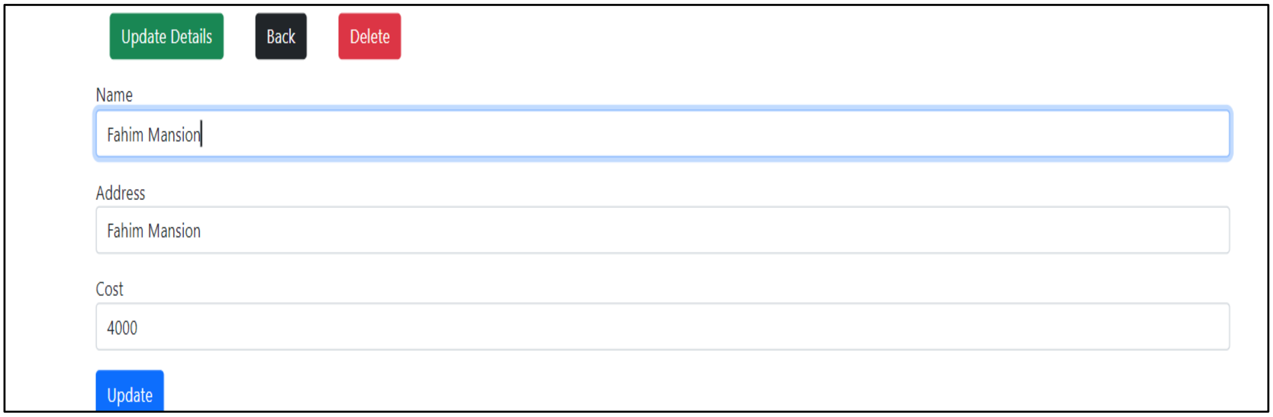
Owner profile: Update and delete details



Name	: Fahim Mansion
Address	: Rangpur
Cost	: 4000

Fig:5.3

Update details: Edit



Name

Fahim Mansion

Address

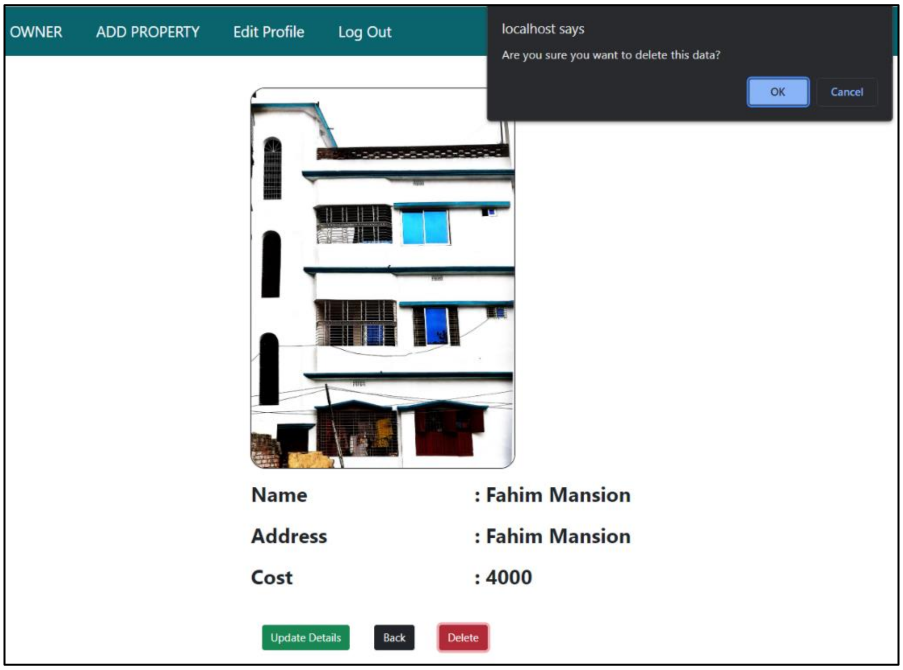
Fahim Mansion

Cost

4000

Fig:5.4

Delete details: Confirm delete



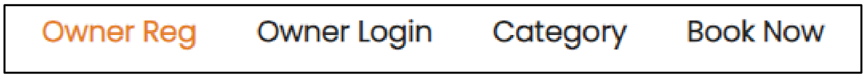
localhost says

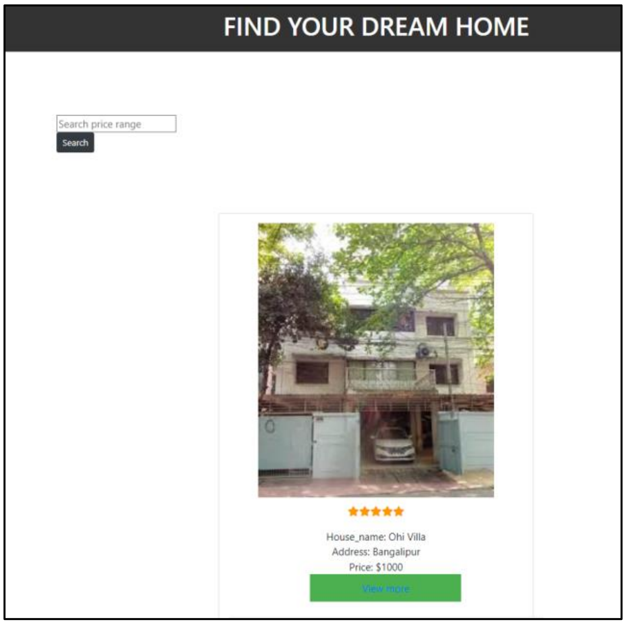
Are you sure you want to delete this data?

Name	: Fahim Mansion
Address	: Fahim Mansion
Cost	: 4000

Fig:5.5

Category: Search, View more

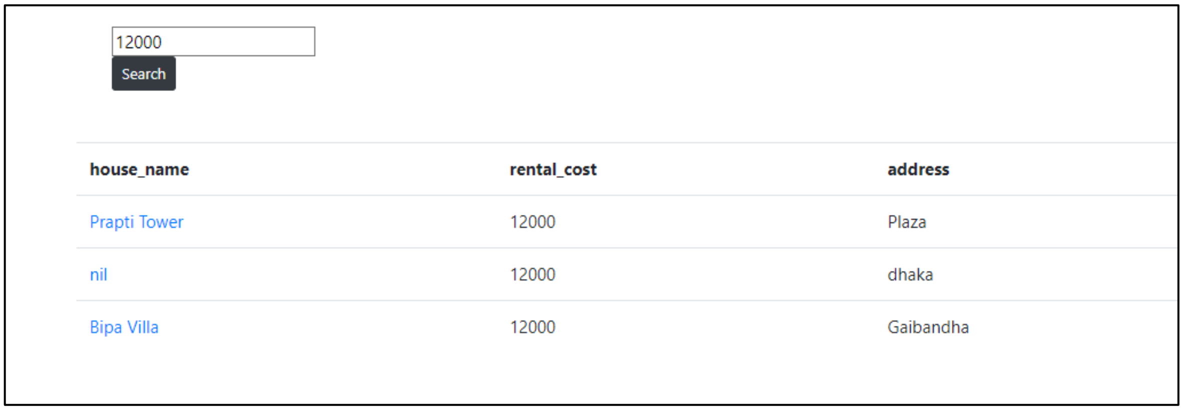




House_name: Ohi Villa
Address: Bangalipur
Price: $1000

Fig:6

Search: Search price range

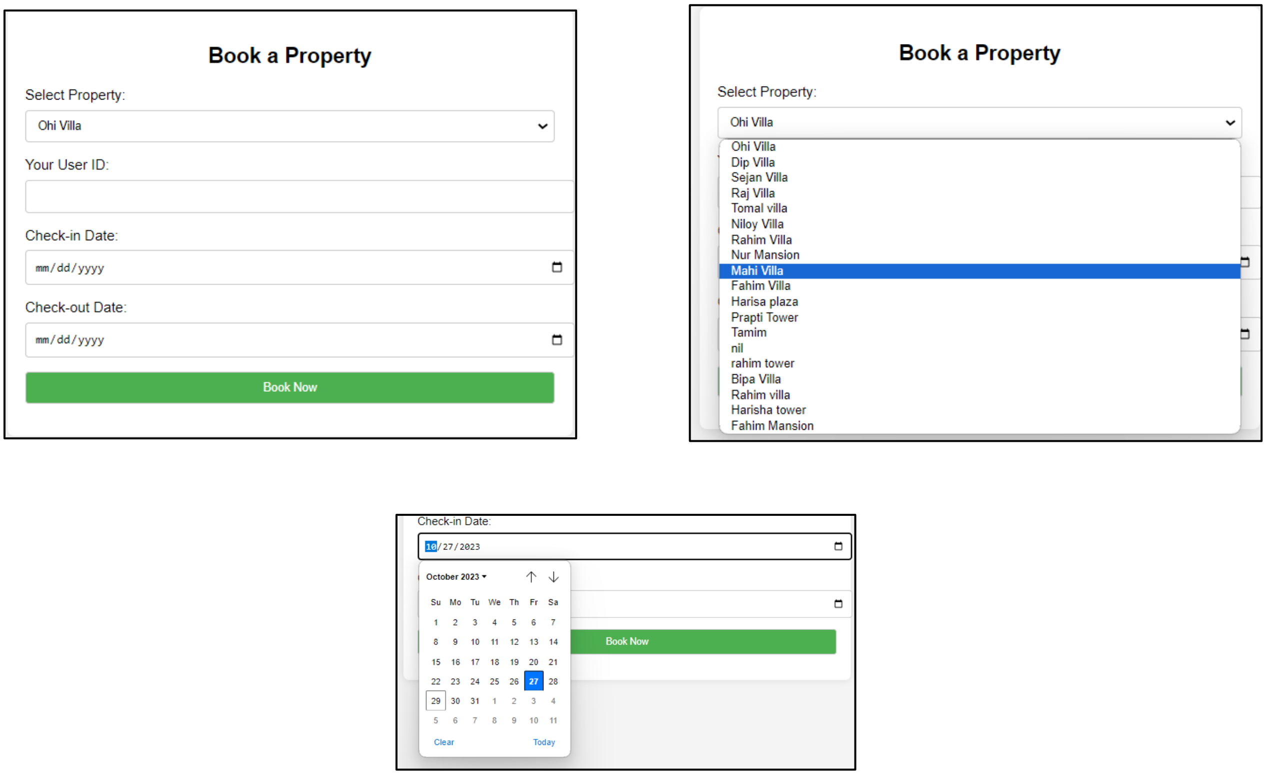


12000

house_name	rental_cost	address
Prapti Tower	12000	Plaza
nil	12000	dhaka
Bipa Villa	12000	Gaibandha

Fig:6.1

Book now: Insert information for book a house



Book a Property

Select Property:

Ohi Villa

Your User ID:

Check-in Date:

mm/dd/yyyy

Check-out Date:

mm/dd/yyyy

Book a Property

Select Property:

Ohi Villa

- Ohi Villa
- Dip Villa
- Sejan Villa
- Raj Villa
- Tomal villa
- Niloy Villa
- Rahim Villa
- Nur Mansion
- Mahi Villa
- Fahim Villa
- Harisa plaza
- Prapti Tower
- Tamim
- nil
- rahim tower
- Bipa Villa
- Rahim villa
- Harisha tower
- Fahim Mansion

Check-in Date:

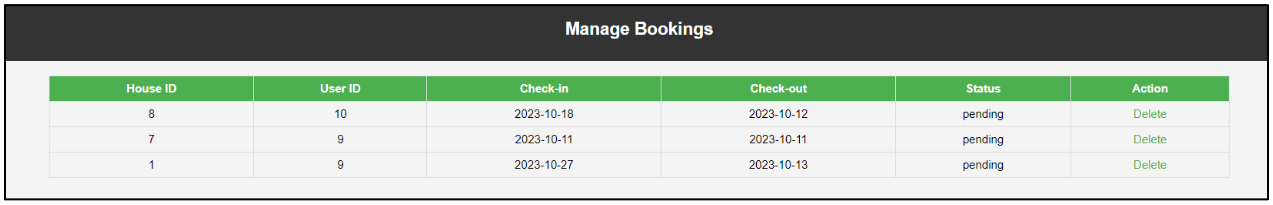
10/27/2023

October 2023

Su	Mo	Tu	We	Th	Fr	Sa
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	6	7	8	9	10	11

Fig:7

Manage bookings: Show bookings, Delete bookings

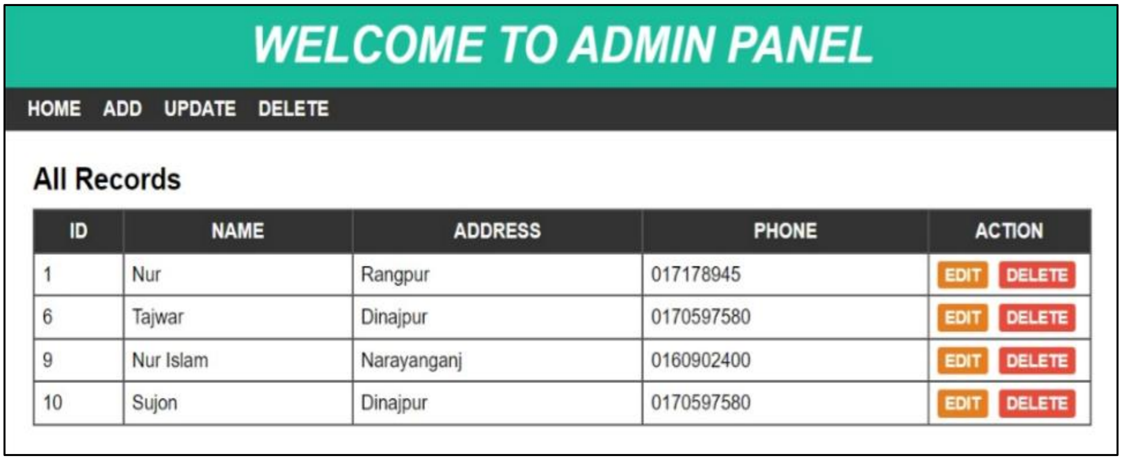


Manage Bookings

House ID	User ID	Check-in	Check-out	Status	Action
8	10	2023-10-18	2023-10-12	pending	Delete
7	9	2023-10-11	2023-10-11	pending	Delete
1	9	2023-10-27	2023-10-13	pending	Delete

Fig:7.1

Admin panel: Add, update and delete records



All Records

ID	NAME	ADDRESS	PHONE	ACTION
1	Nur	Rangpur	017178945	EDIT DELETE
6	Tajwar	Dinajpur	0170597580	EDIT DELETE
9	Nur Islam	Narayanganj	0160902400	EDIT DELETE
10	Sujon	Dinajpur	0170597580	EDIT DELETE

Fig:8

Add panel: Add, update, delete

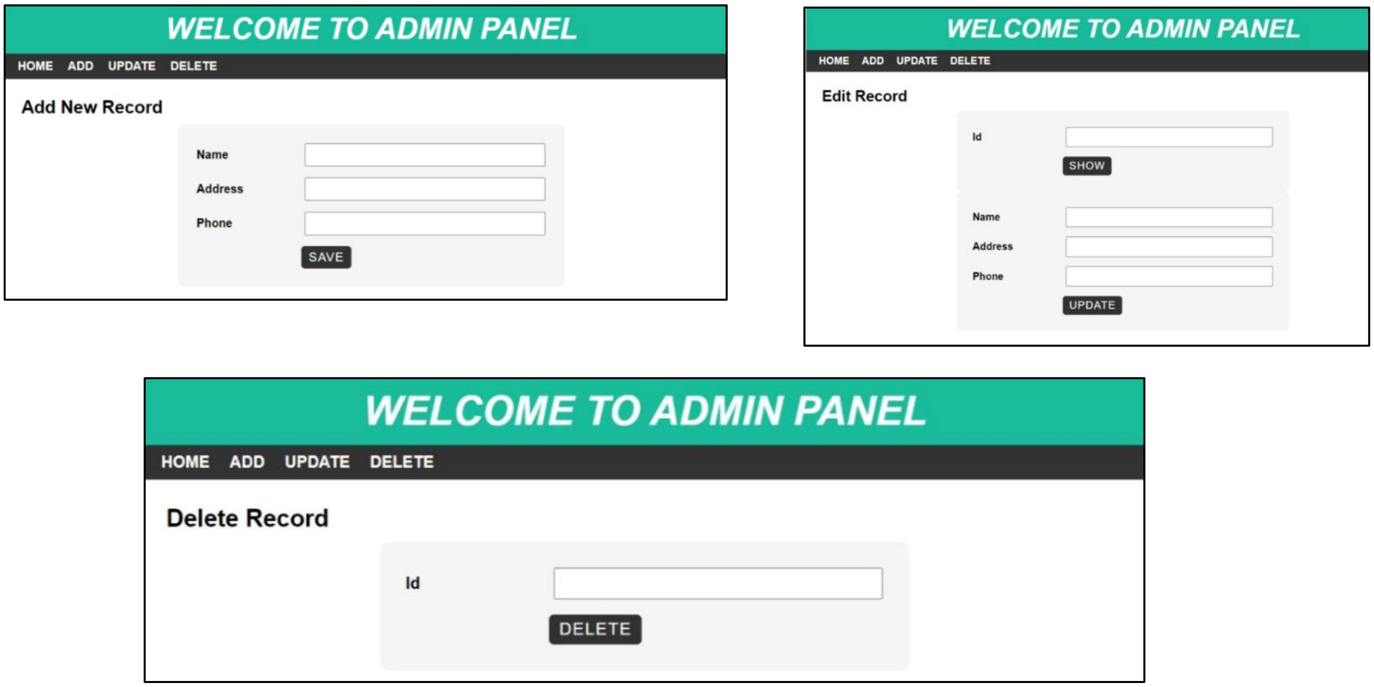


Fig:8.1

10. Conclusion and Limitations

In conclusion, the House Rental Management System project demonstrates a significant step towards streamlining property management processes, enhancing user experience, and promoting efficient communication between landlords and tenants. The implementation of features such as automated rent payments, maintenance requests, and document sharing contributes to a more organized and user-friendly platform.

However, it's essential to acknowledge certain limitations. The system's effectiveness heavily relies on user compliance and the availability of reliable internet connectivity. Additionally, while efforts have been made to ensure security, the system may still be susceptible to unforeseen vulnerabilities. Ongoing updates and user feedback will be crucial in addressing these limitations and ensuring the continual improvement of the House Rental Management System.

References:

www.google.com, www.mysqltutorial.org, www.youtube.com, www.w3schools.com, www.tutorialspoint.com, www.codeacademy.com, www.researchgate.net.

Appendix

Attainment of Complex Engineering Problem (CP)

S.L.	CP No.	Attainment	Remarks
1.	P1: Depth of Knowledge Required	Yes	K3 (Engineering Fundamentals): Require knowledge of database design (Sec. 7).
			K4 (Engineering Specialization): Require Knowledge of Bootstrap (Sec. 8).
			K5 (Design): Flow Chart of Methodology shows solution design of the problem (Sec. 6).
			K6 (Technology): XAMPP server, PHP, MySQL, Bootstrap, etc. (Sec. 8).
			K8 (Research): Studied related application to find limitation (Sec. 3).
2.	P2: Range of Conflicting Requirements	Yes	House Rental (House Rental Management Sec. 1), Web technologies (PHP, MySQL, Bootstrap, etc. Sec. 8), Database Management System (Sec. 7).
3.	P3: Depth of Analysis Required	No	

4.	P4: Familiarity of Issues	Yes	Working with House Rental as a CSE student (Sec. 1).
5.	P5: Extent of Applicable Codes	Yes	Use Waterfall software development model (Sec. 6).
6.	P6: Extent of Stakeholder Involvement and Conflicting Requirements	Yes	Involves owner, tenants and admin (Sec. 4)
7.	P7: Interdependence	Yes	Involve login, house listing, booking management, category, etc (Sec. 9).

Mapping of Complex Engineering Activities (CA)

S.L.	**CA No.**	**Attainment**	**Remarks**
1.	A1: Range of resources	Yes	Involves house owner, tenants and admin, Computer Engineers, Technologies: PHP, MySQL, Bootstrap etc. (Sec. 4 and Sec. 8).
2.	A2: Level of interaction	Yes	Solve problem arises from various conflicting and other issues (Sec. 5).
3.	A3: Innovation	No	
4.	A4: Consequences for Society and the Environment	No	
5.	A5: Familiarity	No	