Project Report on

Hostel Facility Management System

By

Soumyadip Chowdhury (202116009)

Arnab Roy (202116002)

Santa Bir Limboo (202116053)

Sourav Kumar Sharma (202116052)

In partial fulfillment of requirements for the award of degree in

Bachelor of Computer Application

(2021-2024)

Under the Project Guidance of

Dr. Mousumi Gupta
Associate Professor
Department of Computer Applications
Sikkim Manipal institute of Technology
Majhitar, Rangpo, East Sikkim

LIST OF CONTENTS

***Abstract*:** Hostel Facility Management system (HFMS) has been developed for SMIT hostels. This project deals with the development of all activities of hostel along with the student report for those who all are staying in the hostel. This is useful to avoid the manual data entry and is easy to access the data about the hostels. Thereby the management of the hostel becomes easier to maintain and access. The application is provided with an easily understandable and user-friendly interface. The main objectives of the project are to reduce the manual effort for hostel management and processing time, in maintaining the student and staff information. This project provides easy access for the hostel data. The various activities in the hostel management include the management of student data, updating their daily activities report and their mobility information.

***Keywords*:** *User-Friendly, Feedback, HFMS- Hostel Facility Management System*

1. INTRODUCTION

Managing hostels in SMIT is tough because we have to deal with a lot of problems. Things like doing paperwork by hand and not being able to communicate well can all cause issues. The current way of managing hostels isn't great because it's slow and mistakes are made. We have seen many institutions use advanced technology to automate tasks and make things easier. There are several applications developed by well grouped teams. Emmanuel Adu Baffoe developed a hostel management system that deals with similar issues [2]. Sneha Agarwal and her team implements a Cloud based Hostel Facility Management System to automate hostel activities for school children [4].

That's why we want to suggest a similar system called the Hostel Facility Management System. It helps them from the manual work from which it is very difficult to find the record of the students. This system gives an idea about how student registration, room allocation and attendance are maintained in a better way. The hostel management system will also contain special features like how many students are in a room, student's id and free rooms or space available.

1.1. Aim and Objective

Aim:

The primary aim of the HFMS is to make a web application that helps manage and run hostels in SMIT better. It wants to use technology to make things like paperwork and communication easier and faster, so students get better service overall.

Objectives:

1. To enhance the management of hostel facilities in SMIT.
2. To improve efficiency in administrative tasks such as student registration, room allocation, and attendance.
3. To improve communication channels between hostel authorities and students.
4. To reduce the use of manpower and paperwork and make things easier.
5. To have a user-friendly interface.

1.2. Feasibility Study

1. Economic: Open–source and free software are used which may result in low to no cost in developing this project.

2. Technical: React JS, Node JS, Express JS, and MySQL are being used to develop this project which our team members are comfortable with.

3. Schedule: A schedule has been developed so that the project can be completed within the desired timeframe.

1.3. Literature Survey

Developers	Name of the application	Work done	Languages Used	Limitations
Prof. Deepali Narkhede, Rutuja Bamgude, Mayuri Sonawane, Mandar Shevade	Hostel Management System (HMS)	Created an automated system that allows hostel officer to manage the affairs of the hostel.	HTML, PHP, JavaScript, Ajax, CSS, J-FRAME, MySQL and XAMPP server.	Entry/ Exit logs could be added for students going and coming back from holidays.
Emmanuel Adu Baffoe	Hostel Management System	Created an error free, secure, reliable, and fast management system	JavaScript, PHP, MySQL, Apache.	Hostel, Student and room data is not visible.
Adam Brown	Hostel Admin	Developed a user-friendly hostel	Python, SQLite.	Limited scalability for large hostels.

		management system with features for room allocation, fee tracking, and staff management.		
Sarah Lee	HostelGenius	Created a cloud-based hostel management system with multi-tenancy support and automated notifications.	JavaScript, Node.js, MongoDB	Limited customization options, occasional downtime during updates.
Michael Chen	Hostel Tracker	Designed a hostel management system with real-time occupancy tracking and guest feedback management.	Java, MySQL	Basic reporting capabilities, manual data entry for some functionalities.
David Garcia	Hostel Pro +	Enhanced an existing hostel management system with advanced features such as	C#, ASP.NET, SQL Server	High initial setup costs, potential security vulnerabilities

		predictive analytics for demand forecasting, integration with IoT devices for energy management, and support for mobile platforms.		associated with IoT integration.
Emily Johnson	Hostel Ease	Developed a user-friendly hostel management system with a focus on intuitive interface design and accessibility. Implemented features for online payments, maintenance requests, and real-time occupancy tracking.	Ruby on Rails, MongoDB	Limited customization options, occasional performance issues during peak usage times.

1.4. Problem Definition

In our college, managing hostel facilities comes with several challenges. Firstly, the current system is time-consuming and inefficient, leading to delays and errors in tasks like room allocation and student registration. Secondly, there's often a communication gap between hostel staff and students, which can lead to misunderstandings and dissatisfaction. Lastly, the reliance on paperwork and manpower for tasks like record-keeping and maintenance adds to the workload and increases the chances of errors. Overall, there is a need for a more smooth and efficient approach to managing hostel facilities to enhance the living experience for students.

1.5 SRS

1.5.1 Functional Requirements

F1: Login process

- Description - Users are required to provide their login credentials.

F1.1: Admin login

- Input - Hostel staff's admin password.
- Output - Staff will be redirected to admin homepage.
- Processing - Selecting as admin login and providing the admin password, it will be verified from the backend. If valid, the admin homepage will be opened, else wrong password warning will be displayed.

F2: Admin module

- Description - The hostel staffs i.e. the administrator will have full control of the site maintenance. They will have access like modify the student data, take attendance, discard complaints against fulfilment and allot rooms to students.

F2.1: View student data

- Input - Student registration no., name, or room no.

- Output - Input matched in the database corresponding to the student's data will be displayed.
- Processing - input data will be searched in the database and matching data will be sent back to frontend.

F2.2: Attendance

- Description - Students list will be displayed to mark attendance along with student search can also be done for a specific student's attendance.
- Input - Registration no., Name, Room no.
- Output - Students' details will be displayed for marking attendance.
- Processing - Each attendance marked will be sent and stored in the database with the current date.

F2.3: Room Renovation/Complaints

- Input - Registration no., name, room no., phone no., type, description of the complaint by the student.
- Output - The complaint raised will be displayed in the renovation/complaint section in the admin module.
- Processing - Students input will be stored in the database temporarily and will be removed by the admin when resolved.

F2.4: Room allotment

- Description - All room data will be displayed with details like occupancy status (vacant, semi-vacant or occupied)
- Input – Reg no., name, phone no., address, parents phone no.
- Output - Room successfully allotted message.
- Processing - On selecting a room, a form will be filled, and the data will be stored in the database with the student's details to the allotted room.

1.5.2 Non-Functional Requirements

- **Reliability:** The website should be reliable and should hardly ever breakdown and even if it does it should recover very quickly without losing any data.

- **User-friendly:** The website should be easy to use with an user-friendly interface.
- **Compatibility**: The system should work on different computers and web browsers, like Chrome or Safari.
- **Stress Handling:** Even if lots of people use the system at once, like during registration, it should still work without any problems.

1.6 Solution Strategy

Our solution strategy focuses on developing a Hostel Facility Management System (HFMS) to address the challenges faced in managing hostel facilities within our college.

1. **Digital Transformation:** We'll transition all hostel management tasks to digital platforms. This includes student registration, room allocations and attendance. By going digital, we simplify tasks, reduce paperwork, and make information more accessible.
2. **Complaints Tab:** We'll implement a dedicated complaints tab within our digital platform. This feature allows students to easily submit complaints or suggestions, leading to better communication between students and hostel staff. Staff then can deal with the issue accordingly.
3. **Digitalization of Records:** We'll digitize all hostel records, including student information, attendance records, and maintenance logs. This not only saves physical storage space but also makes information retrieval quicker and more efficient.
4. **Regular Evaluation:** We'll establish a system for regular evaluation of our solution. This involves monitoring key metrics such as complaint resolution times, user satisfaction levels, and efficiency gains from digitalization. Based on this feedback, we'll make necessary adjustments to optimize our systems continuously.

2. PROJECT PLAN

2.1 Hardware and Software Requirements

2.1.1 Developmental Requirements

Hardware Requirements:

- Processor: Intel Core i5 or Ryzen 5 processor or higher
- RAM: 8 GB of RAM
- Storage: 50 GB

Software Requirements:

- Frontend: React JS
- Backend: Node JS, Express JS and MySQL

2.1.2 Minimum Requirement

Hardware Requirements:

- Will run on any device such as desktop, laptop, mobile phone, tablet etc. as long as it can run a browser.

Software Requirements:

- Any Browser such as Google Chrome, Microsoft Edge, Firefox etc.

2.2 Team Structure

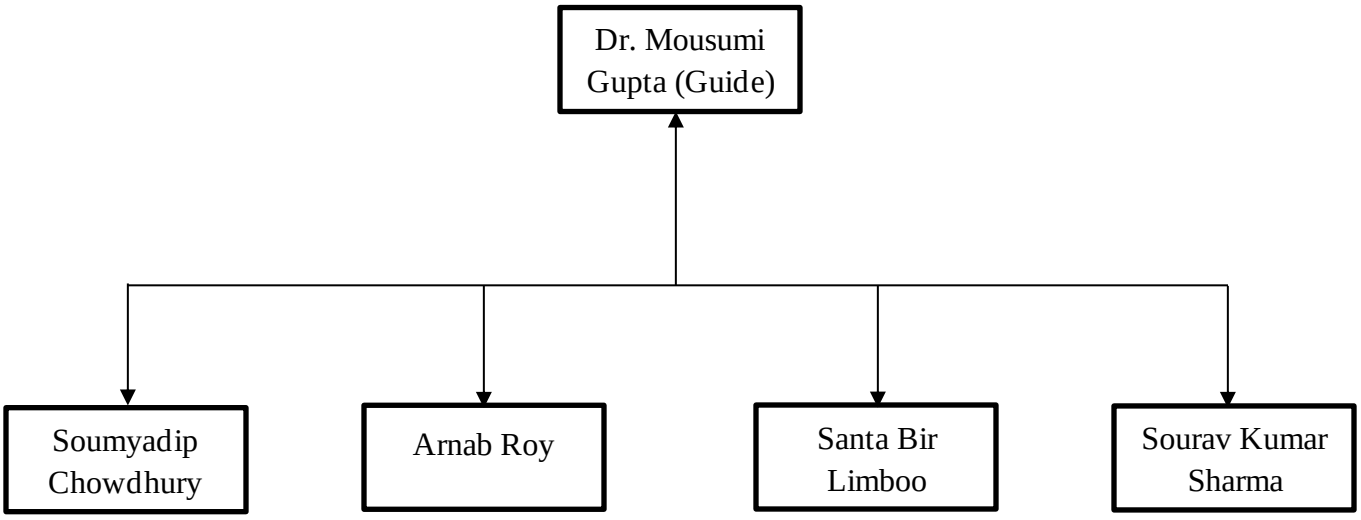

Figure 1: Software Development Life cycle

Project Guide: Dr. Mousumi Gupta

Members:

- Soumyadip Chowdhury
- Arnab Roy
- Santa Bir Limboo
- Sourav Kumar Sharma

2.3 Software Development Life Cycle

In our Hostel Management System, Iterative Waterfall model has been proposed for software development. In practical development environments, as developers are tend to make a large number of errors in each of the phases of the life cycle model so in order to prevent the error we need to correct it as per when necessary. The source of the defects can be many, which usually get detected much later in the life cycle. Once a defect is detected,

we need to go back to the phase where the error occurred in order to correct it. Iterative waterfall model provides the flexibility to do so as shown in the figure below.

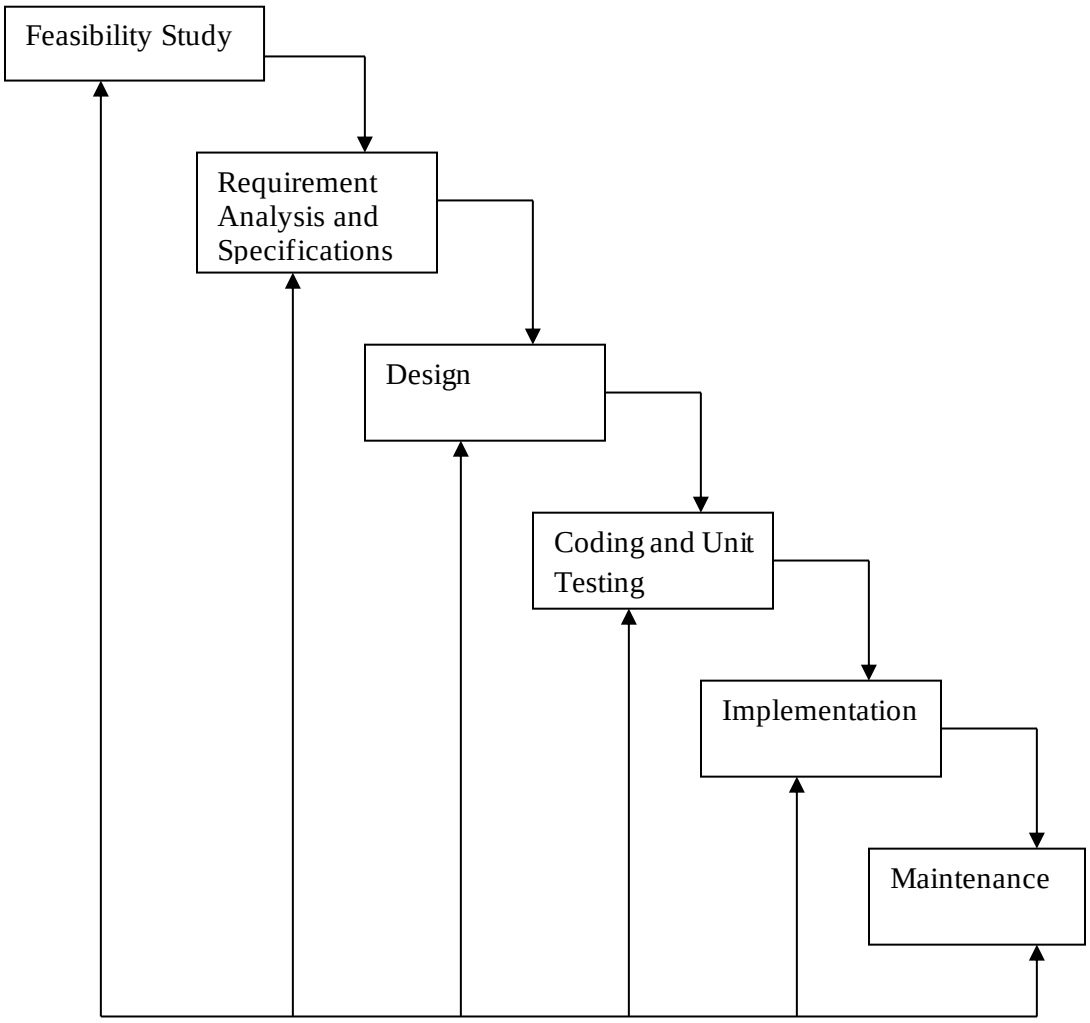

Figure 2: Iterative Waterfall Model

2.4 Gantt-Chart

Activities	18 Jan – 25 Jan	28 Jan – 10 Feb	12 Feb – 19 Feb	26 Feb - 10 March	14 March – 10 April	12 April – 26 April
Literature Survey						
Problem Definition						
Design and Development						
Testing and Validation						
Documentation						

Figure 3: Gantt Chart

3. DESIGN STRATEGY FOR THE SOLUTION

3.1 Data Flow Diagram

Level 0

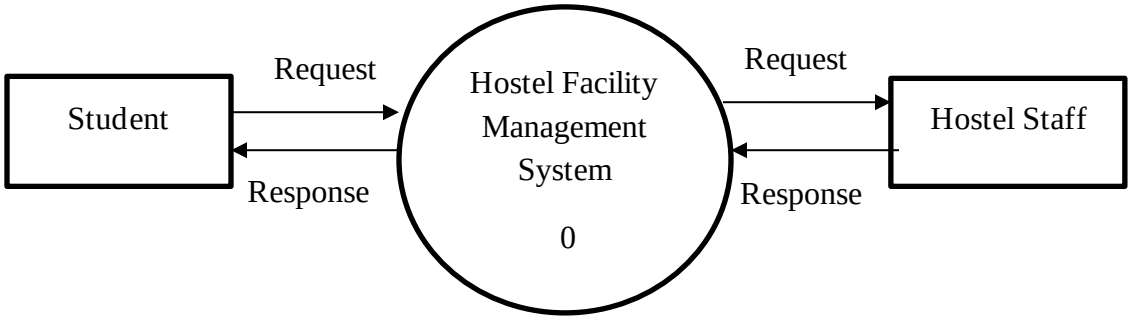

Figure 4: DFD Level 0

Level 1

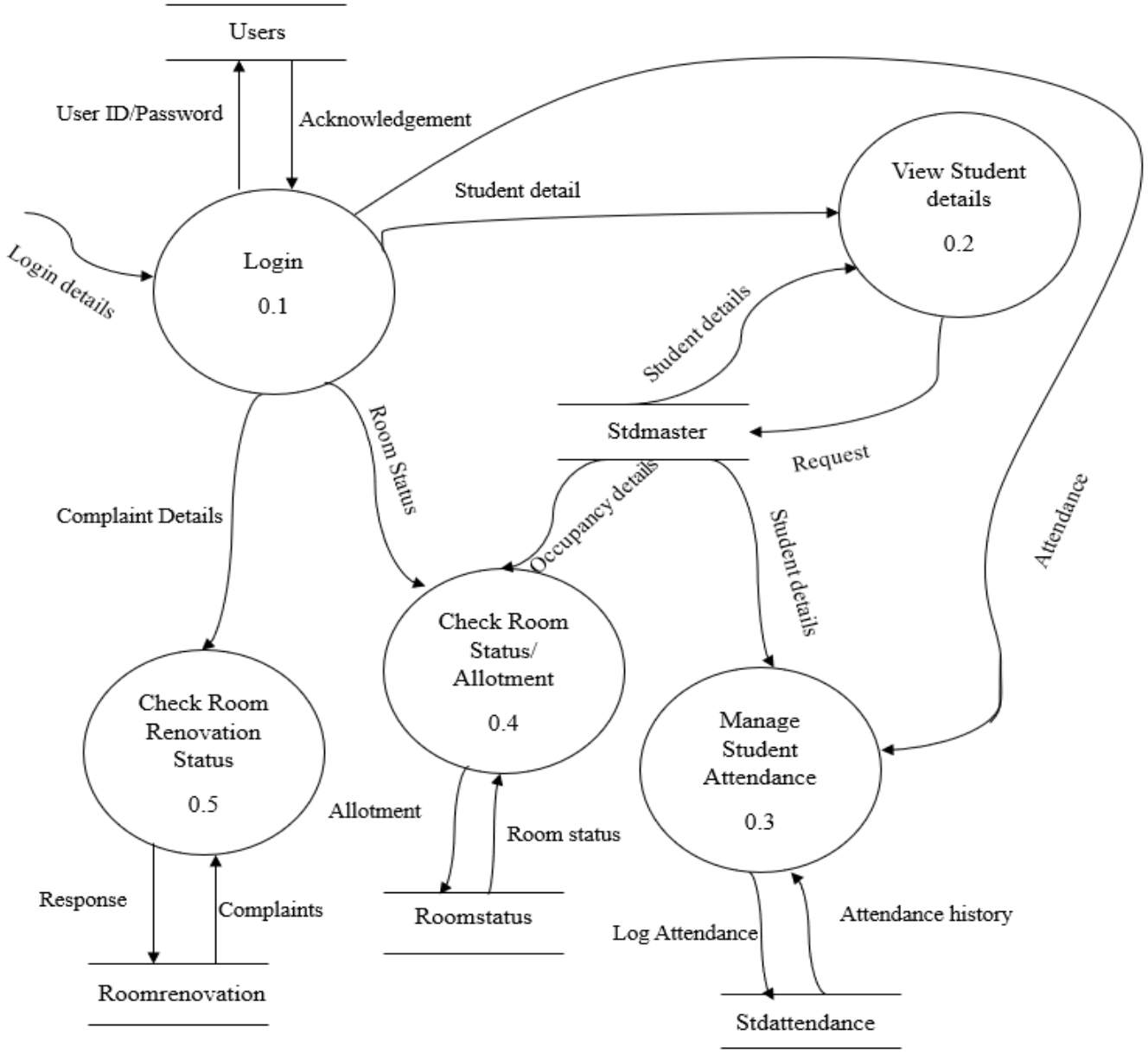

Figure 5: DFD Level 1

3.2 Entity Relationship Diagram

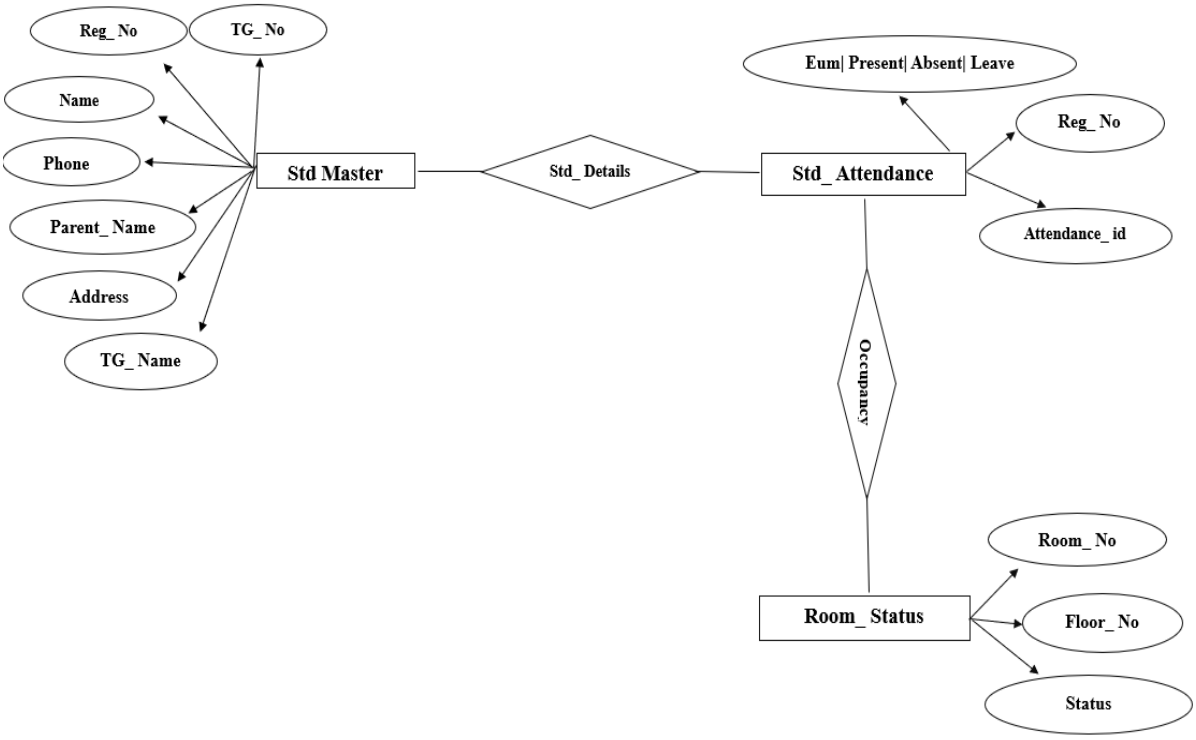

Figure 6: ER Diagram

4. IMPLEMENTATION DETAILS

4.1 Pseudo Code

4.1.1 Attendance

```
function updateStatus(button, regNo) {
        if (button.backgroundColor === 'green') {
        updatedStatus = 'Present';
    } else if (button.backgroundColor === 'yellow') {
        updatedStatus = 'On leave';
    } else {
        updatedStatus = 'Absent';
    }
    fetch_request('/update-attendance') {
        method: 'POST',
        body: { status: updatedStatus, regNo: regNo }
    }
}
```

4.1.2 Renovation

```
function deleteRow(rowId){
  fetch_request('/delete-reno-row'){
     method: 'POST',
     body: { rowId: rowId }
  }
}
```

4.1.3 Renovation Logs

```
function filterLogs(button){
    if (button === 'All'){
        fetchAllLogs();
    }
    else {
        fetch_request('/filter-logs'){
            method: 'POST',
            body: { type: button }
        }
        }
}
```

4.1.4 Room Status

```
function RoomStatus(){
    const [roomsData, setRoomsData] = useState([]);
    fetch_request('http://localhost:5000/room-status', {
        method: 'GET'
    }
    .then(data){
        setRoomsData(data);
    }
}
```

4.1.5 Student Details

```
function StudentDetails(){
    const [studentDetailsData, setStudentDetailsData] = useState([]);
    fetch_request('http://localhost:5000/student-details', {
        method: 'GET',
    })
    .then(data){
```

```
        setStudentDetailsData(data);
    }
}
```

4.1.6 New Student Allotment

```
function handleSubmit(){
    fetch_request('/student-entry'){
        method: 'POST',
        body:
            {
            name: name,
            regNo: regNo,
            pNo: pNo,
            roomNo: roomNo,
            parentNo: parentNo,
            address: address
        }
    }
    .then(data){
        alert(data.message);
    }
}
```

5. RESULT AND DISCUSSION

5.1 Snapshots

5.1.1 Login Page

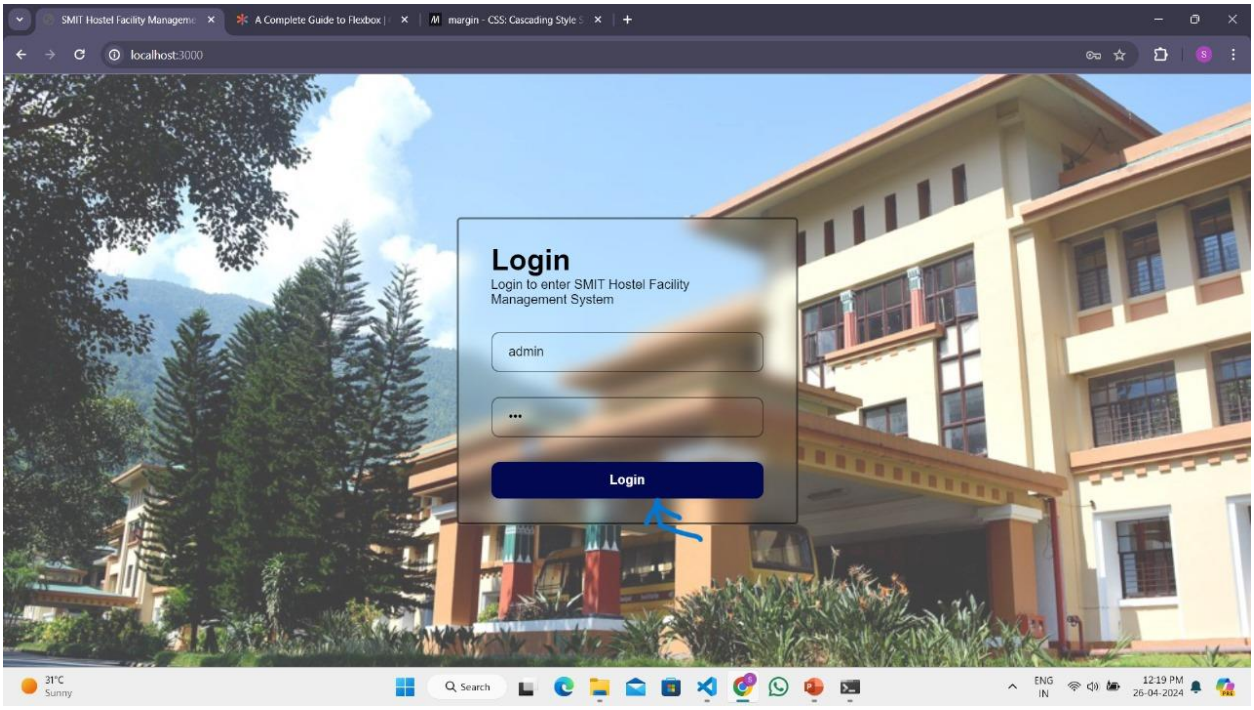

Figure 7: Login Page

Description- Here users will be able to login with their credentials.

5.1.2 Renovation Status

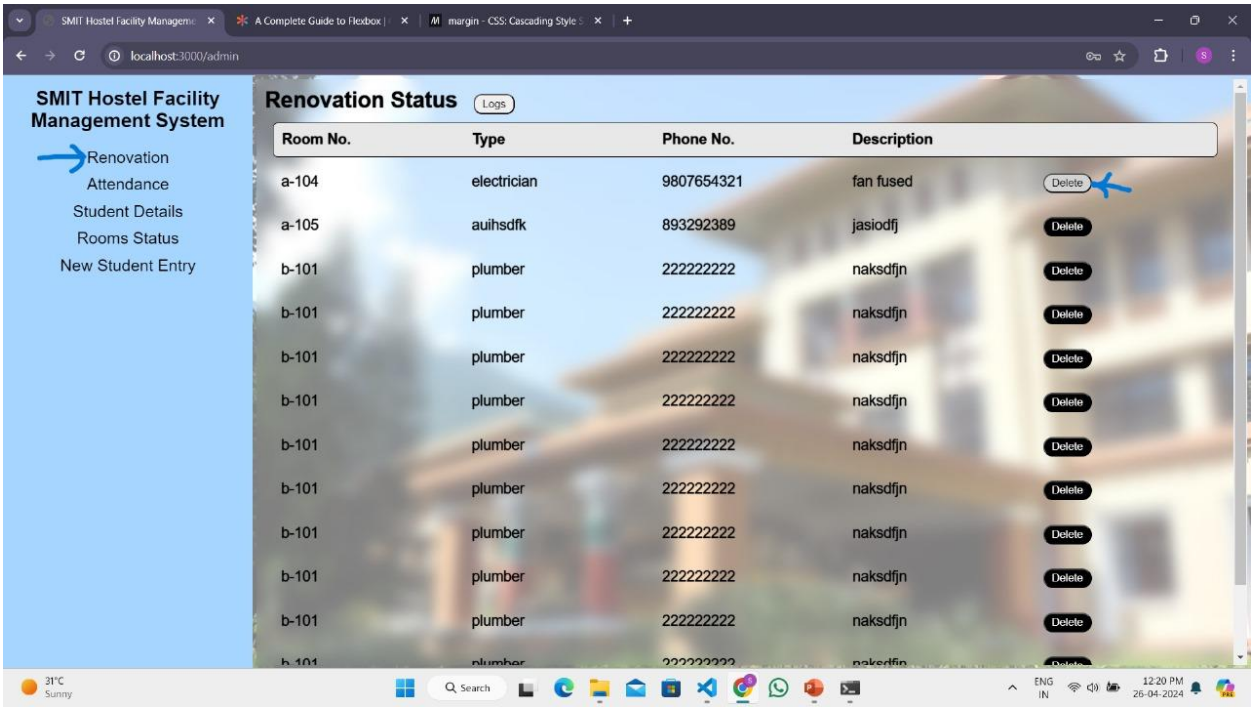

Figure 8: Renovation Status

Description- In the Renovation Section, the staffs will be able to see the list of room complaints so that they can act accordingly.

After a complaint has been resolved the staff can click on the delete button on that particular complaint and that will move to the renovation log.

5.1.3 Renovation Log

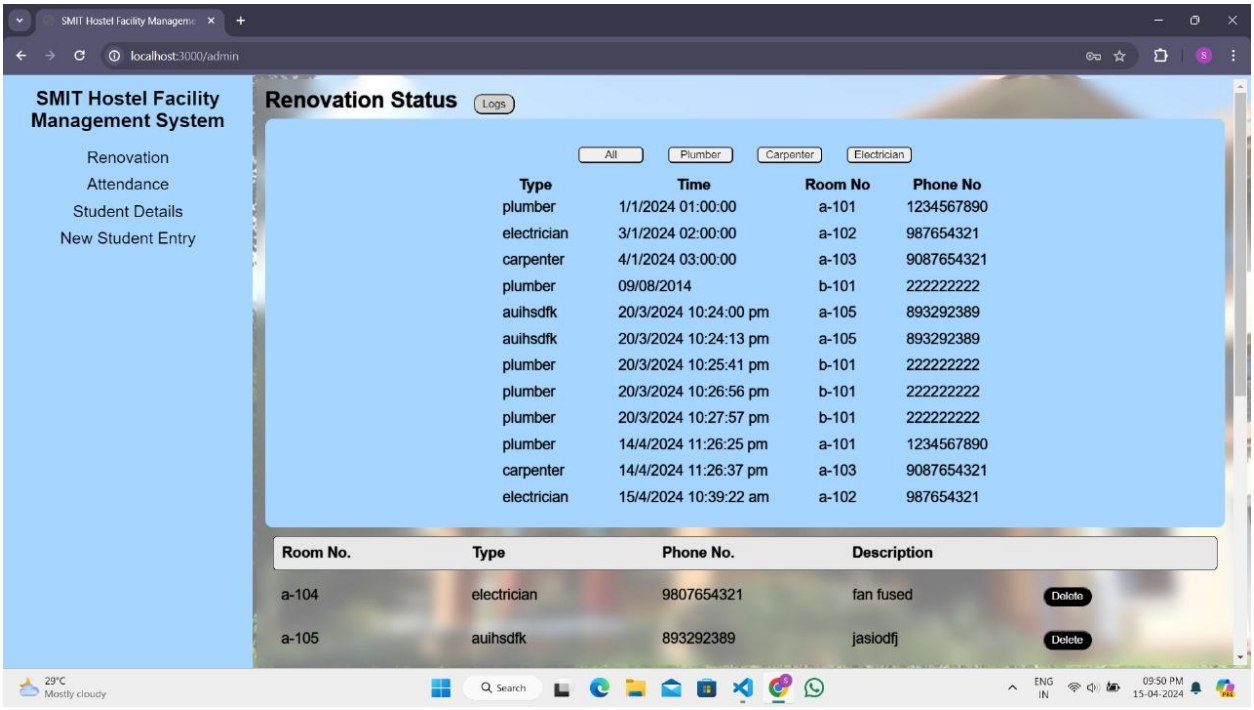

Figure 9: Renovation Log

Description- Renovation log will display all the previous complains that have been resolved.

5.1.4 Room Status

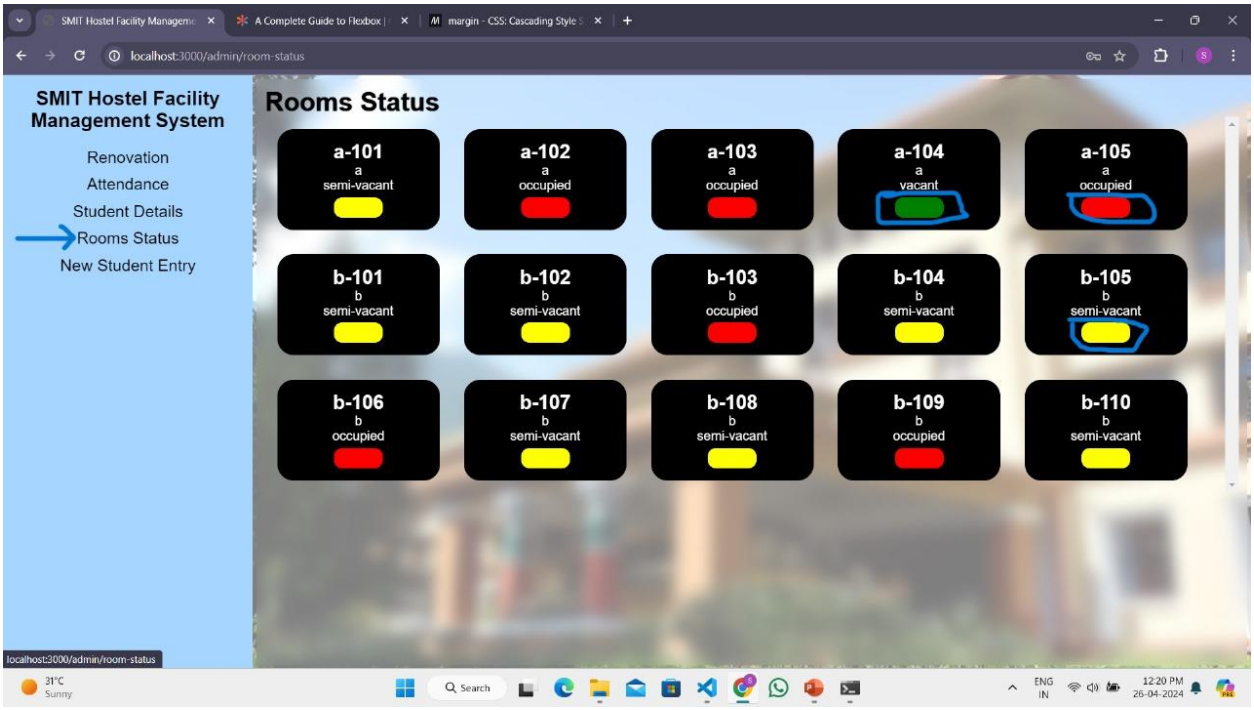

Figure 10: Room Status

Description- The Room occupancy Status will show all the rooms and its occupancy status with the help of these colours.

Green- This means that the room is fully vacant and new students can be allotted to that.

Yellow- This means that the room is semi vacant with only a single student and one more student can be allotted.

Red- This means that the room is completely occupied, and no new students can be allotted.

5.1.5 New Student Entry

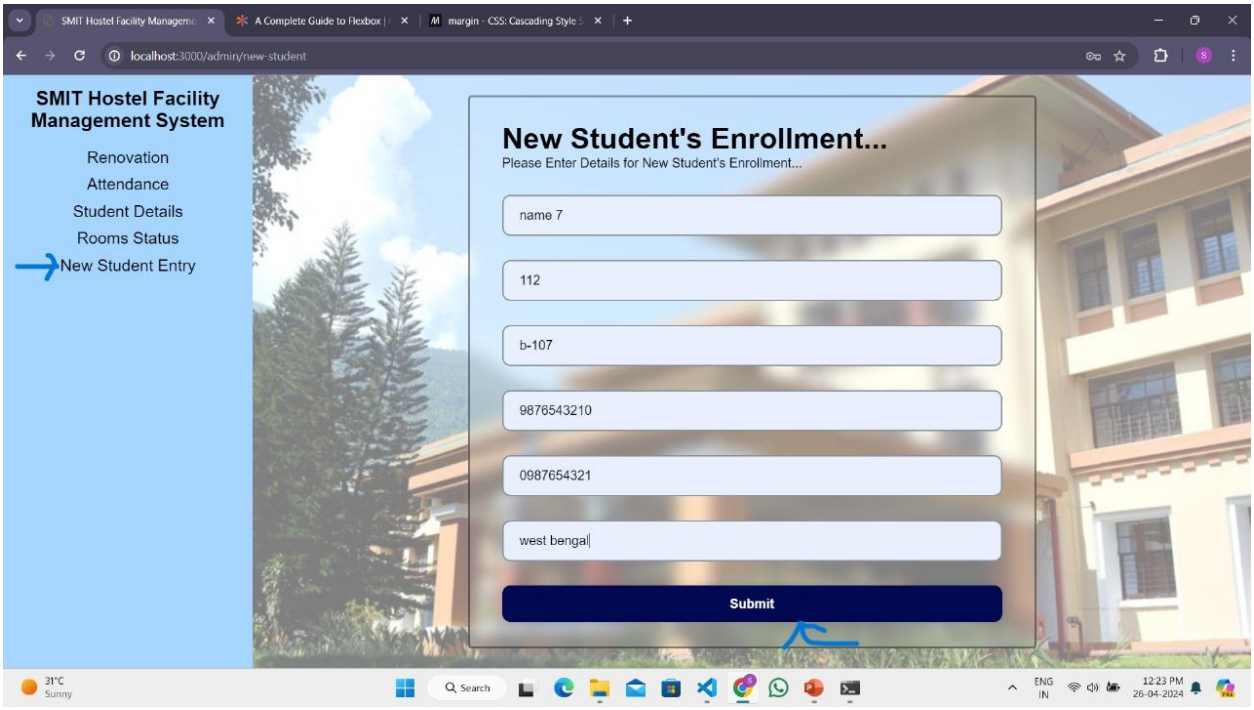

Figure 11: New Student Entry

Description- New students planning to stay in hostel can be registered in this section by providing their details.

5.1.6 New Student Entry Completed

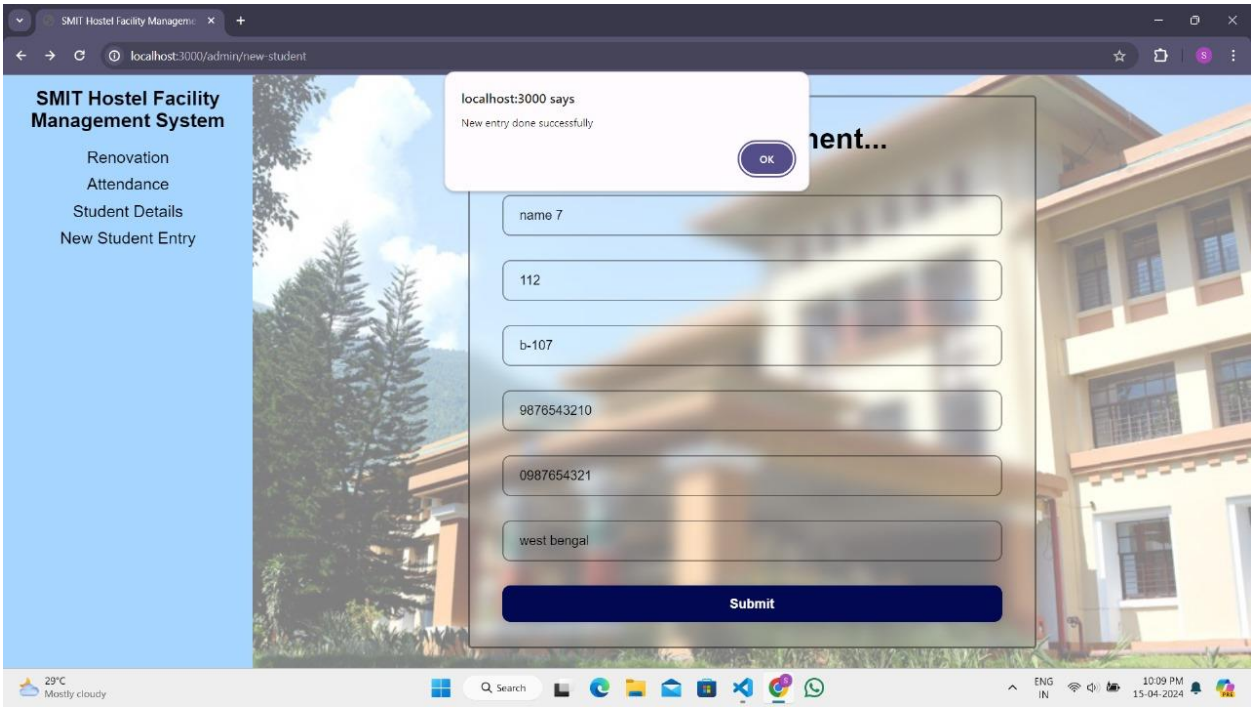

Figure 12: New Student Entry Completed

Description- After submitting the New Student Entry form, a dialogue box will appear stating that the entry has been done successfully.

5.1.7 Student Details

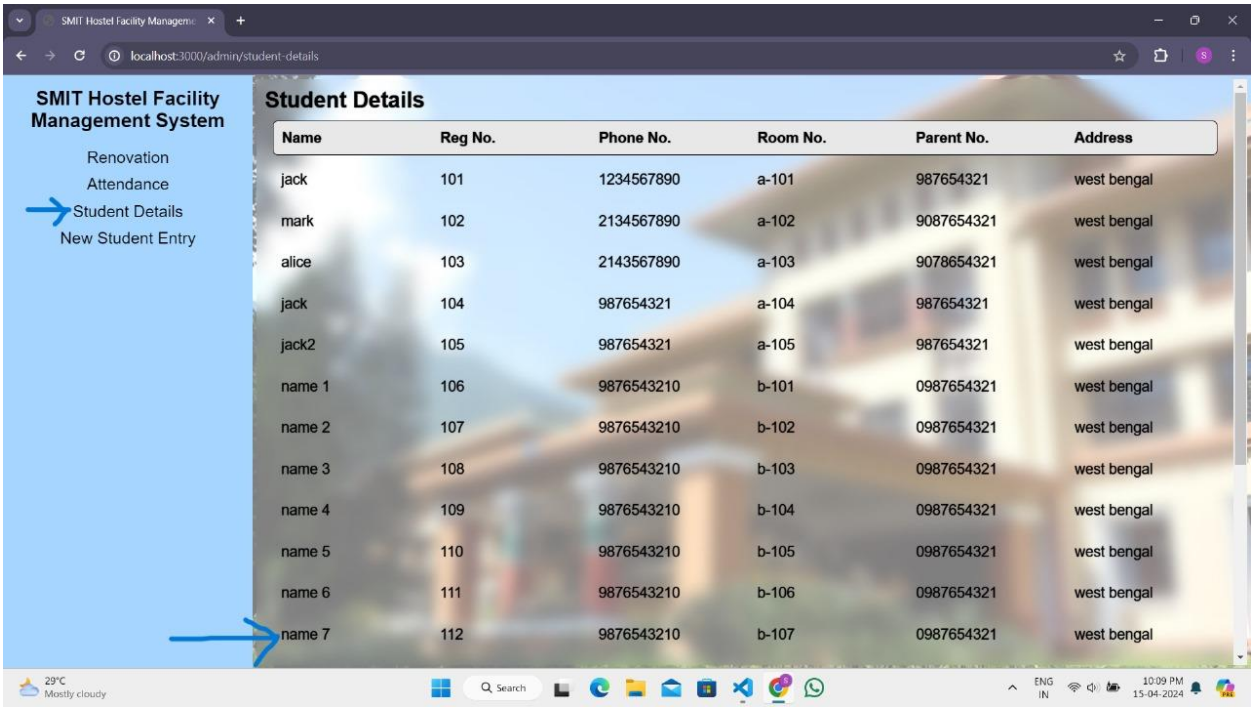

Figure 13: Student Details

Description- Student details section will display a list of all the students staying in the hostel and their details.

5.1.8 Attendance

Figure 14: Attendance

Description- In the Attendance section, a list of students will be displayed along with options to mark their attendance.

Green- This means that the student is present.

Yellow- This means that the student is on leave.

Red- This mean that the student is absent.

We have also included an option for marking every student present so that the staff doesn't have to mark students present individually. In Case any student is absent or on leave they can then mark them that individually.

We also have an option for All on leave for during the vacations.

5.1.9 Search Student

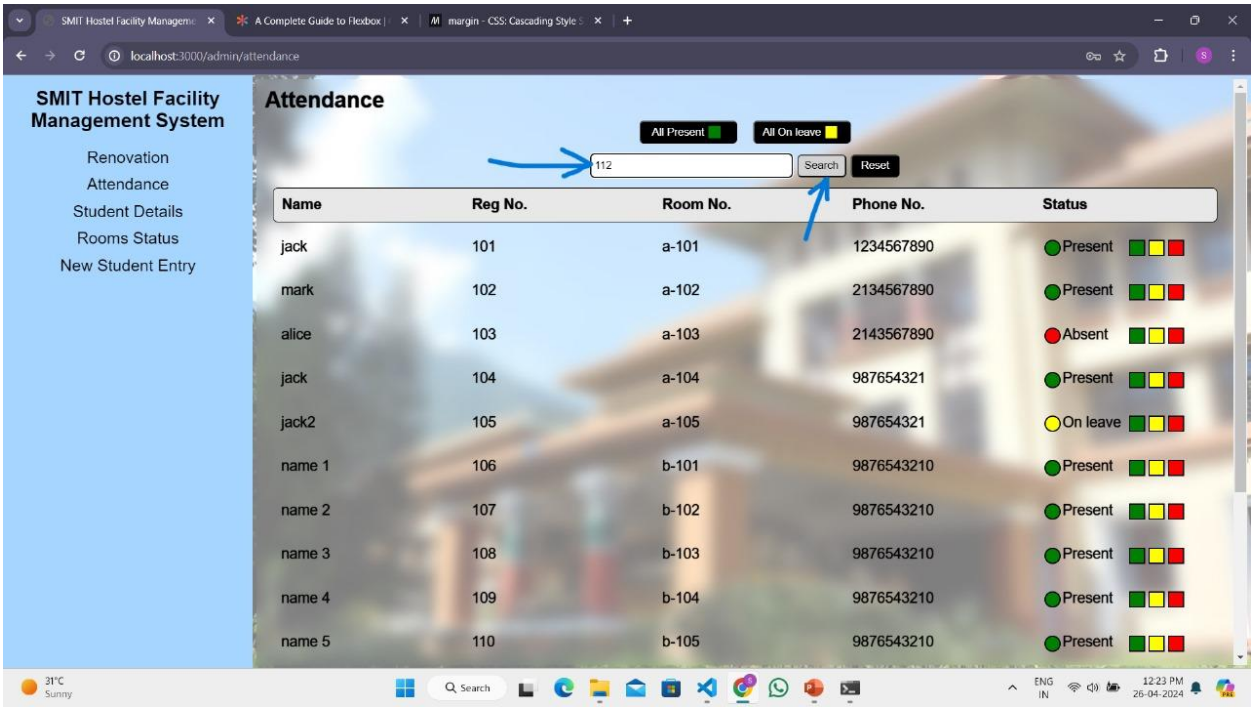

Figure 15: Search Student

Description- The staff can search any student with their name, registration number or room number.

5.1.10 Search Results

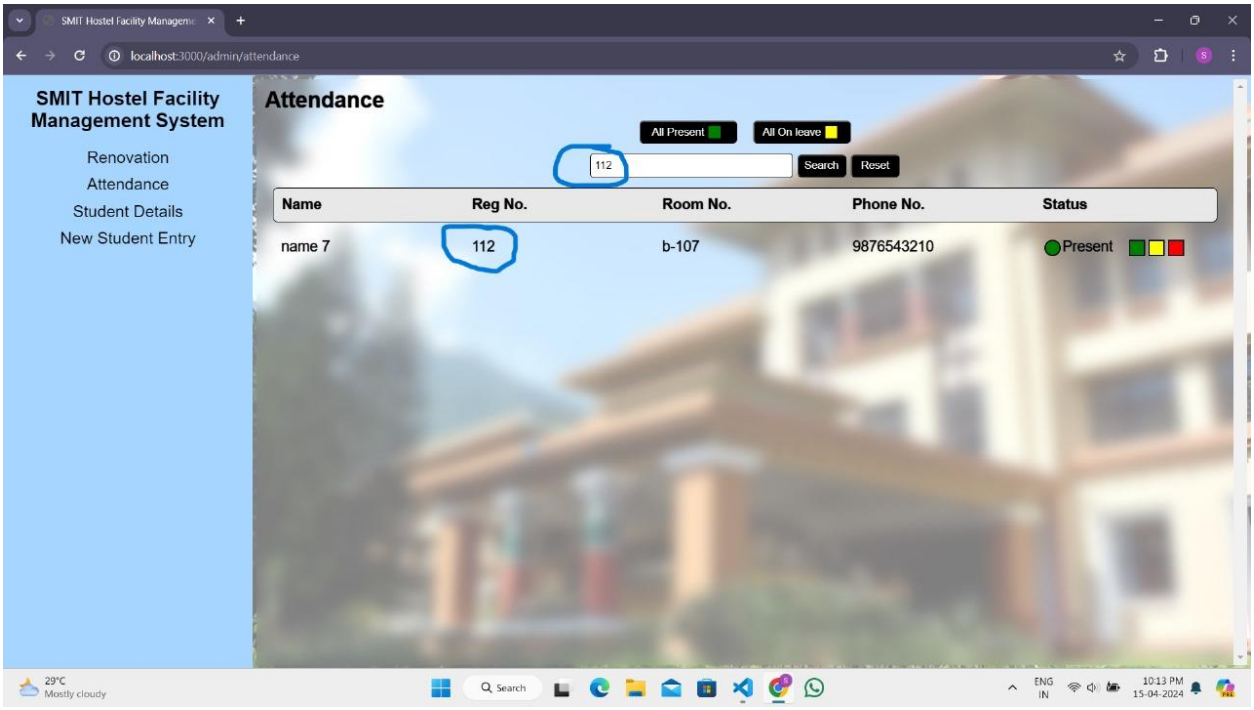

Figure 16: Search Results

Description- Here's an example of search results and this was searched using the student's registration number.

6. SUMMARY AND CONCLUSION

6.1 Summary of Achievements

The new hostel management system we've introduced will make some significant improvements. Firstly, it makes tasks like assigning rooms to new students much quicker and simpler. It's also helped with registering new students and keeping track of who's staying in the hostel. It makes life easier for both students and staff. It's much easier for students to communicate with the staff if they have any issues or need help. There is an option just for that, so students can get assistance faster. Additionally, going digital means we're using a lot less paper now, which is better for the environment. And if we need to find something, it's much quicker to look it up on the computer. Lastly, feedback will be conducted, and changes will be made based on that. Overall, the new system will make living in the hostel a lot easier and more modern.

6.2 Problems Encountered During Project

There were difficulties in accurately knowing and understanding the requirements of the hostel management system. This led to some confusion and delays in the planning phase as we worked to clarify and refine the project scope.

6.3 Future Scopes of the Project

- Individual student accounts can be maintained with their records of behaviour in the hostel, any disciplinary action taken against them, etc.
- Developing a mobile application for the HFMS would provide students and staff with greater flexibility and accessibility.
- As technologies become more common, there's an opportunity to integrate the hostel management system with IoT devices to automate certain tasks.

6.4 Limitations of the Project

- Student entry/exit for leaves cannot be recorded in this system and have to be managed manually in traditional methods.
- Because hostel operations will rely on technology for essential functions, any technical problems or outages could cause disruptions.
- Some users, particularly older staff members or students who are less familiar with technology, may struggle to adapt to the new system.

6.5 Conclusion

In summary, the hostel management system has made things better by making tasks easier, improving communication, and reducing paperwork. Despite some challenges, we've worked hard to overcome them and create a system that's more user-friendly for both staff and students. Moving forward, we'll keep working to make it even better and ensure it meets the needs of everyone involved in the hostel. Overall, it's a big step forward in making hostel life smoother and more enjoyable for everyone.

REFERENCES

[1] Prof. Deepali Narkhede, Rutuja Bamgude, Mayuri Sonawane, Mandar Shevade , https://www.ijraset.com/research-paper/hostel-management-system-hms

[2] JUNE, EMMANUEL ADU BAFFOE. HOSTEL MANAGEMENT SYSTEM. Diss. 2016.https://www.academia.edu/37040228/HOSTEL_MANAGEMENT_SYSTEM_full_project_1_

[3] Brown, A. (2019). "HostelAdmin: Simplifying Hostel Management." Journal of Applied Computing, 8(2), 45-52.

[4] Agrawal, Sneha, Sugandh Rastogi, and Shivani Trivedi. "Cloud Based Hostel Facility Automation System." 2023 International Conference on Computational Intelligence, Communication Technology and Networking (CICTN). IEEE, 2023.

[5] Lee, S. (2020). "HostelGenius: Streamlining Hostel Operations." International Journal of Computer Science and Applications, 11(4), 78-84.

[6] Chen, M. (2021). "HostelTracker: Enhancing Hostel Efficiency." Journal of Information Systems, 15(3), 112-118.

[7] Garcia, D. (2022). "HostelPro+: Advancements in Hostel Management Technology." International Journal of Computer Science and Information Security, 20(1), 112-120.

[8] Johnson, E. (2021). "HostelEase: Enhancing Hostel Management Experience." Journal of Information Technology Research, 14(2), 67-75.