DBMS Mini Project Report

Title: Hostel Allocation System

- Team Member – Rajdeep Rathod
- Enrollment Number – A70405224048
- Guide Name – Dr. Dipak Raskar
- Department of Computer Science and Engineering
- Amity University Mumbai
- Academic Year 2025 – 2026
- Semester III

Table of Contents

List of Tables/Figures

Figure No.	Title	Page No.
Figure 1	Level 0 and 1 DFD	04
Figure 2	Entity-Relationship (ER) Diagram of Hostel Allocation System	04
Figure 3	Sample Data	05
Figure 4	Dashboard Interface (Home Page)	05
Figure 5	Rooms Page	05

Chapter 1: Introduction

- Background and Motivation: The Hostel Allocation System aims to automate the process of room and bed assignments for students in college hostels. Traditionally, the allocation process is managed manually, which often leads to inefficiencies, data redundancy, and mismanagement. This system minimizes human intervention, ensures fair allocation, and maintains accurate records.
- Problem Statement: Manual hostel management often results in data inconsistency, difficulty in managing room availability, and delays during student admission. A robust database-backed application is needed to centralize this process.
- Project Objectives: -
  - Automate student allocation based on preferences.
  - Provide a real-time dashboard for warden and admin.
  - Maintain accurate occupancy and availability records.
  - Support manual and smart allocation modes.

Chapter 2: Literature Review

- Existing hostel management systems often lack dynamic allocation features. Traditional systems use static forms and require manual updates in spreadsheets. Recent developments in DBMS and Flask-based applications have enabled efficient CRUD operations with web interfaces.

- This project incorporates MySQL for relational database management, Flask for backend handling, and Bootstrap for responsive frontend design. It improves upon prior work by including a smart allocation feature and real-time visualization dashboard.

Chapter 3: Software Requirement Specification (SRS)

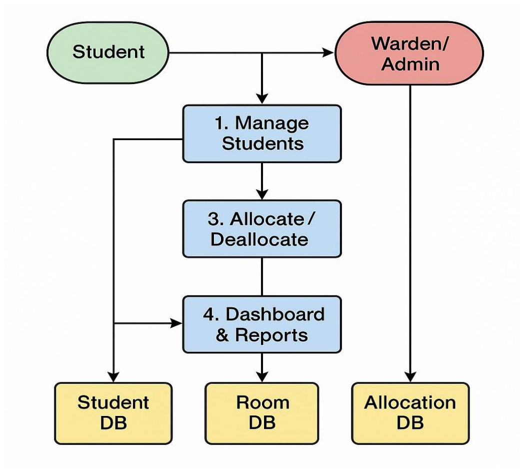
- Functional Requirements:
  - Student registration and management.
  - Room and dormitory data management.
  - Manual and smart allocation of rooms.
  - View allocations and deallocations.
  - Dashboard analytics.
- Non-Functional Requirements:
  - User-friendly interface.
  - Scalability and data integrity.
  - Secure CRUD operations.
- Hardware/Software Used:
  - Python 3.11
  - Flask Framework
  - MySQL Database
  - HTML, CSS, Bootstrap
  - VS Code
  - MySQL Workbench

Chapter 4: System Analysis and Design

- Assumptions: Each dorm consists of multiple rooms, each room has fixed bed capacity, and students can only occupy one bed.
- Data Flow Diagrams: represent how information moves within the Hostel Allocation System.

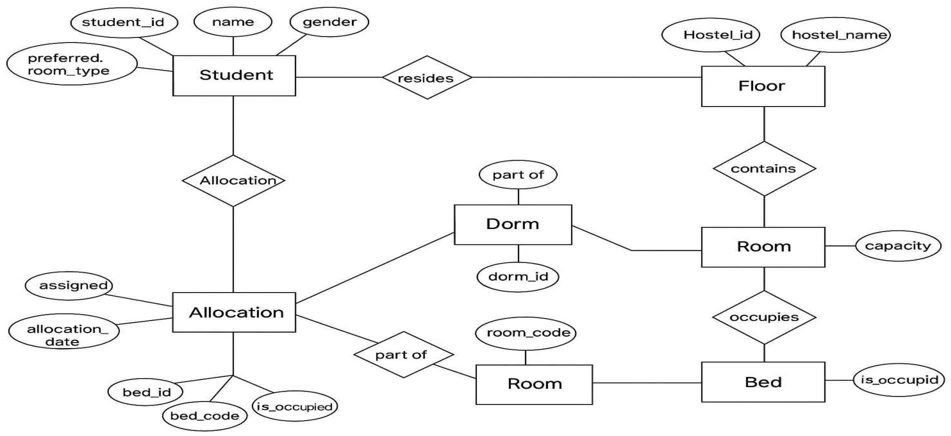
The Level 0 diagram shows the overall interaction between external entities (Students and Warden) and the system.

The Level 1 diagram decomposes the system into key functional modules such as Student Management, Room Management, and Allocation Management.



These diagrams help visualize the logical flow of data, the storage points, and the processes involved in allocation and record management.

- Entity-Relationship (ER) diagram

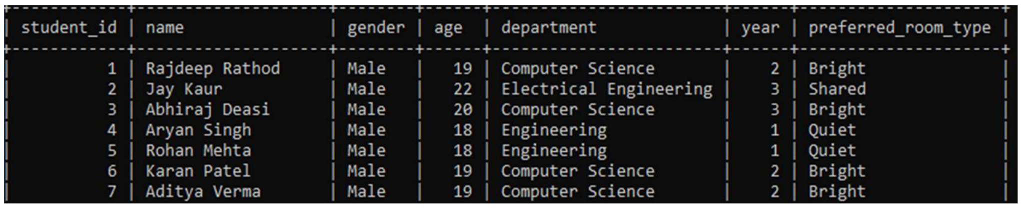


- Use Case Diagram:
  - Actors: Admin/Warden, Student.
  - Use Cases: Register Student, Add Room, Allocate, Deallocate, View Dashboard.

- Schema Design: Schema includes 7 main tables: hostel, floor, dorm, room, bed, student, and allocation.

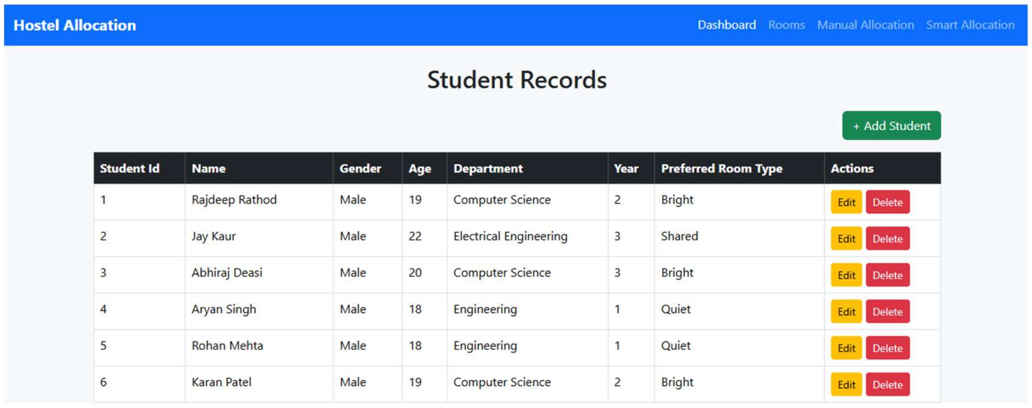
Chapter 5: Database Design

- Relational Schema: - hostel(hostel_id PK, hostel_name, gender) - floor(floor_id PK, floor_number, hostel_id FK) - dorm(dorm_id PK, dorm_number, floor_id FK, capacity) - room(room_id PK, room_code, dorm_id FK, capacity, room_type) - bed(bed_id PK, bed_code, room_id FK, is_occupied) - student(student_id PK, name, age, gender, department, year, preferred_room_type) - allocation(allocation_id PK, student_id FK, room_id FK, allocation_date)
- Normalization: All tables follow 3NF. No transitive dependencies or redundant data.
- Sample data: Include 28 sample rows for students and rooms.



```
+------------+-------------------------+--------+-----+-------------------------+------+---------------------+
| student_id | name                    | gender | age | department              | year | preferred_room_type |
+------------+-------------------------+--------+-----+-------------------------+------+---------------------+
|          1 | Rajdeep Rathod          | Male   |  19 | Computer Science        |    2 | Bright              |
|          2 | Jay Kaur                | Male   |  22 | Electrical Engineering  |    3 | Shared              |
|          3 | Abhiraj Deasi           | Male   |  20 | Computer Science        |    3 | Bright              |
|          4 | Aryan Singh             | Male   |  18 | Engineering             |    1 | Quiet               |
|          5 | Rohan Mehta             | Male   |  18 | Engineering             |    1 | Quiet               |
|          6 | Karan Patel             | Male   |  19 | Computer Science        |    2 | Bright              |
|          7 | Aditya Verma            | Male   |  19 | Computer Science        |    2 | Bright              |
```

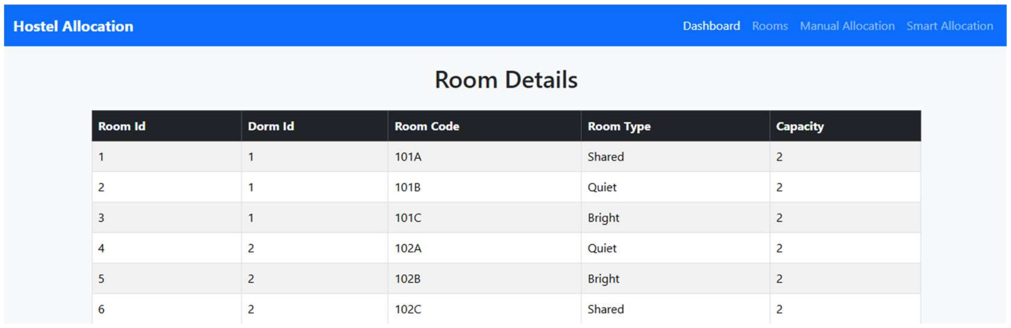
Chapter 6: Implementation



Student Records

Student Id	Name	Gender	Age	Department	Year	Preferred Room Type	Actions
1	Rajdeep Rathod	Male	19	Computer Science	2	Bright	Edit Delete
2	Jay Kaur	Male	22	Electrical Engineering	3	Shared	Edit Delete
3	Abhiraj Deasi	Male	20	Computer Science	3	Bright	Edit Delete
4	Aryan Singh	Male	18	Engineering	1	Quiet	Edit Delete
5	Rohan Mehta	Male	18	Engineering	1	Quiet	Edit Delete
6	Karan Patel	Male	19	Computer Science	2	Bright	Edit Delete

Dashboard

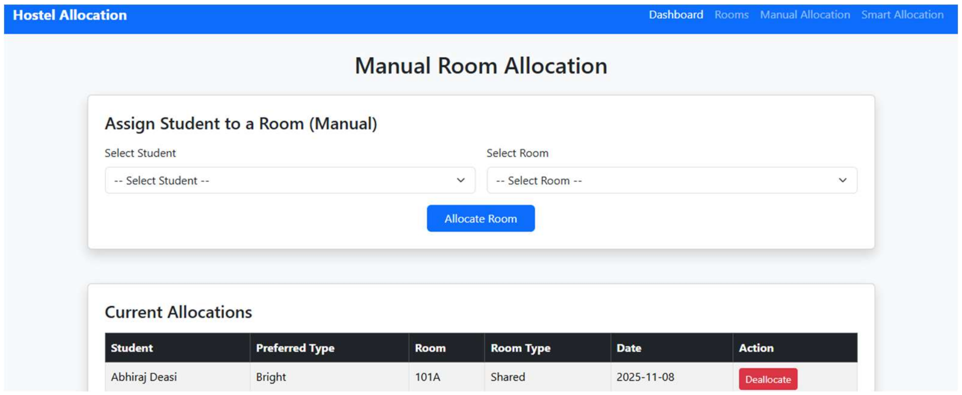

Hostel Allocation

Dashboard Rooms Manual Allocation Smart Allocation

Room Details

Room Id	Dorm Id	Room Code	Room Type	Capacity
1	1	101A	Shared	2
2	1	101B	Quiet	2
3	1	101C	Bright	2
4	2	102A	Quiet	2
5	2	102B	Bright	2
6	2	102C	Shared	2

Rooms


Hostel Allocation

Dashboard Rooms Manual Allocation Smart Allocation

Manual Room Allocation

Assign Student to a Room (Manual)

Select Student: -- Select Student --

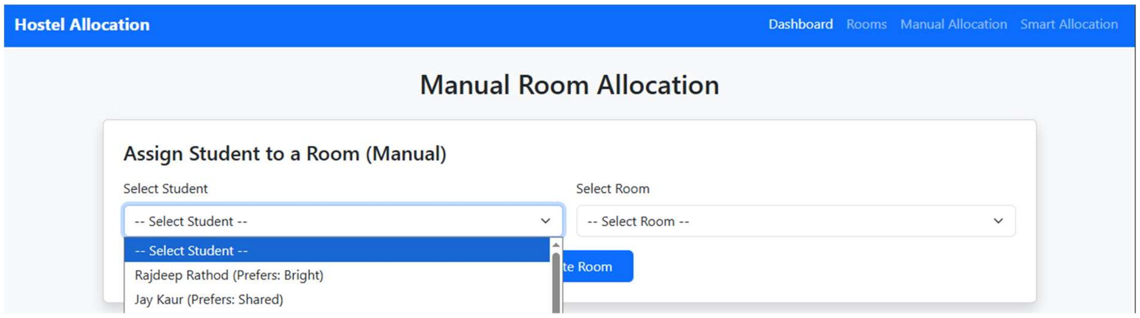
Select Room: -- Select Room --

Allocate Room

Current Allocations

Student	Preferred Type	Room	Room Type	Date	Action
Abhiraj Deasi	Bright	101A	Shared	2025-11-08	Deallocate

Manual Room Allocation


Hostel Allocation

Dashboard Rooms Manual Allocation Smart Allocation

Manual Room Allocation

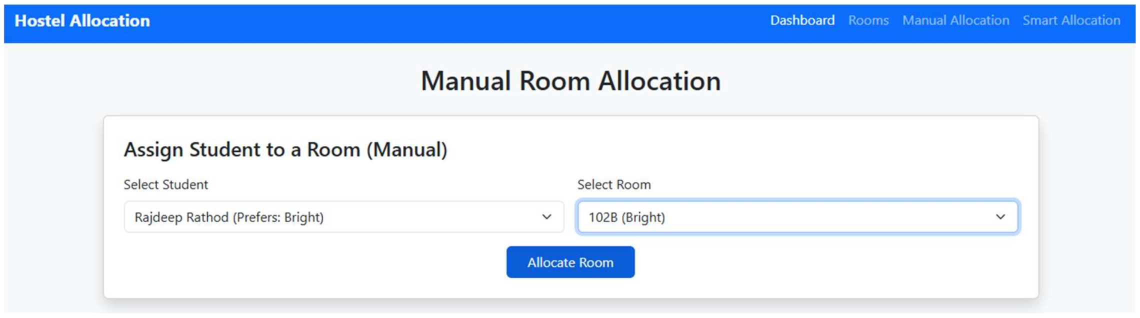
Assign Student to a Room (Manual)

Select Student: -- Select Student --

- -- Select Student --
- Rajdeep Rathod (Prefers: Bright)
- Jay Kaur (Prefers: Shared)

Select Room: -- Select Room --

te Room


Hostel Allocation

Dashboard Rooms Manual Allocation Smart Allocation

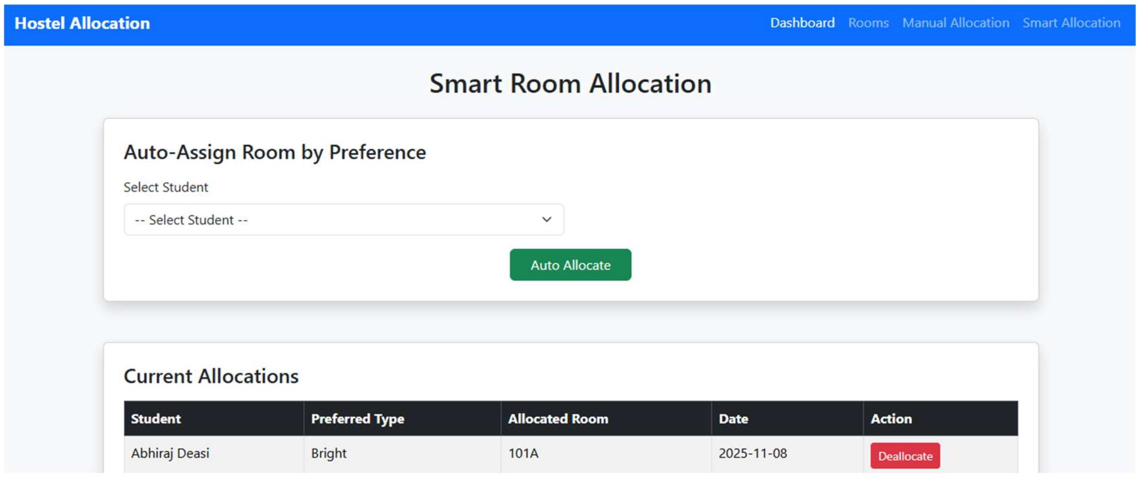
Manual Room Allocation

Assign Student to a Room (Manual)

Select Student: Rajdeep Rathod (Prefers: Bright)

Select Room: 102B (Bright)

Allocate Room



Smart Room Allocation

Auto-Assign Room by Preference

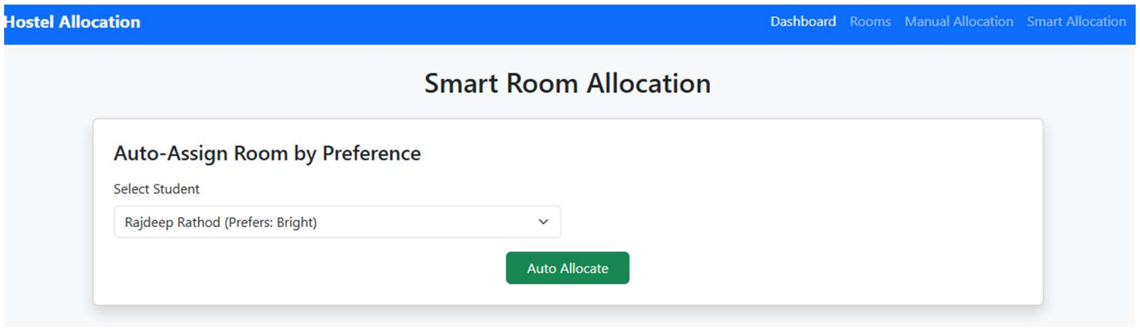
Select Student: -- Select Student --

Auto Allocate

Current Allocations

Student	Preferred Type	Allocated Room	Date	Action
Abhiraj Deasi	Bright	101A	2025-11-08	Deallocate

Smart Room Allocation



Smart Room Allocation

Auto-Assign Room by Preference

Select Student: Rajdeep Rathod (Prefers: Bright)

Auto Allocate

Sample SQL Queries:

```
SELECT s.name, r.room_code, a.allocation_date
FROM allocation a
JOIN student s ON a.student_id = s.student_id
JOIN room r ON a.room_id = r.room_id;
```

Trigger:

```
CREATE TRIGGER after_allocation_insert
AFTER INSERT ON allocation
FOR EACH ROW
UPDATE room SET capacity = capacity - 1 WHERE room_id = NEW.room_id;
```

Backend Snippet:

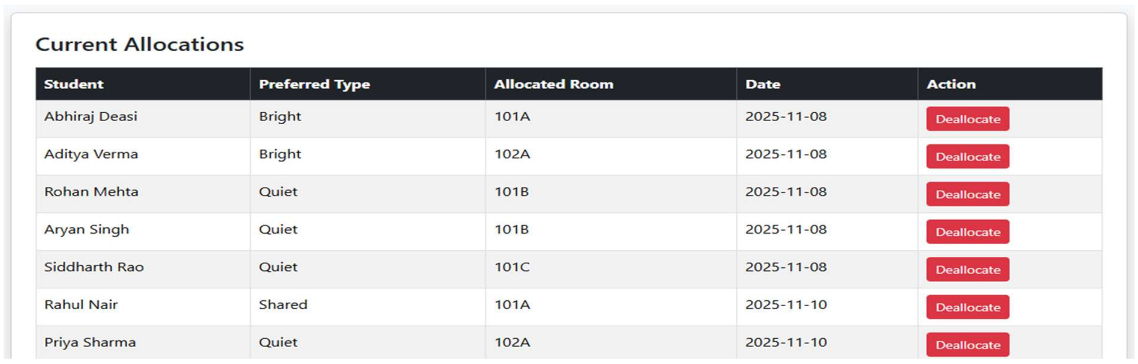
```
@app.route('/auto_allocate', methods=['GET', 'POST'])
def allocate_room_smart():
```

```
# auto-assign logic based on preferred type
...
```

Chapter 7: Testing and Results

Test ID	Test Case	Input	Expected Output	Actual Output	Result
TC01	Add Student	Valid Details	Student Added	Pass	Pass
TC02	Manual Allocation	Student + Room	Allocated	Pass	Pass
TC03	Smart Allocation	Student with Preference	Auto Assigned	Pass	Pass

- Error handling and limitations: Handles invalid entries, null fields, and full room scenarios.
- Results: System performs allocation successfully under all test cases.



Current Allocations

Student	Preferred Type	Allocated Room	Date	Action
Abhiraj Deasi	Bright	101A	2025-11-08	Deallocate
Aditya Verma	Bright	102A	2025-11-08	Deallocate
Rohan Mehta	Quiet	101B	2025-11-08	Deallocate
Aryan Singh	Quiet	101B	2025-11-08	Deallocate
Siddharth Rao	Quiet	101C	2025-11-08	Deallocate
Rahul Nair	Shared	101A	2025-11-10	Deallocate
Priya Sharma	Quiet	102A	2025-11-10	Deallocate

Chapter 8: Discussion

The system provides efficiency, accuracy, and transparency in hostel management. It replaces manual tracking with a reliable database system.

Challenges included managing relational integrity and implementing smart allocation logic.

Comparison with traditional systems shows significant improvement in automation and data reliability.

Chapter 9: Conclusion

This project successfully demonstrates the design and implementation of a hostel management system using MySQL and Flask. It automates room allocations and improves hostel administration. Future work can integrate IoT sensors for live bed tracking and deploy the system on the cloud.

Chapter 10: References

1. MySQL Documentation – https://dev.mysql.com/doc/
2. Flask Official Docs – https://flask.palletsprojects.com/
3. Bootstrap Documentation – https://getbootstrap.com/
4. W3Schools SQL Tutorial – https://www.w3schools.com/sql/