**HOSTEL MANAGEMENT**

A

RDBMS Lab Project Report

Submitted in the partial fulfilment of the requirement for the award of

Bachelor of Technology in Computer and Communication Engineering

By:

Vansh Jaggi

229309101

CCE – B



April, 2024

Department of Computer and Communication Engineering

School of Computing and Intelligent Systems

Manipal University Jaipur

VPO. Dehmi Kalan, Jaipur, Rajasthan, India – 303007

Department of Computer and Communication Engineering

School of Computing and Intelligent Systems, Manipal University Jaipur,

Dehmi Kalan, Jaipur, Rajasthan, India- 303007

# **STUDENT DECLARATION**

I hereby declare that this project “Hostel Management” is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the University or other Institute, except where due acknowledgements has been made in the text.

|  |  |
| --- | --- |
| Place: Manipal University Jaipur | Vansh Jaggi |
| Date: 22/04/2024 | 229309101 |
|  | B.Tech. (CCE) 4th Semester |
|  | Section: B |

# **ABSTRACT**

This project presents a Hostel Management System developed using Python, Tkinter, and MySQL. The system focuses on core functionalities such as student registration, automatic room allocation, and record viewing. Administrators can easily add new students, capturing essential details, and allocate rooms automatically based on availability. Additionally, the system allows viewing of comprehensive records including student information, departments, hostels, available rooms, and mess facilities. This simplified approach aims to offer a user-friendly solution for basic hostel management needs, ensuring ease of use and efficient student accommodation.

**TABLE OF CONTENTS**

Contents Page Number

[Student Declaration i](#_Toc5969)

[Abstract ii](#_Toc5970)

1. Introduction 1
2. Software and Hardware Requirements 2
3. Entity-Relationship Model 3
4. Relational Model 4
5. Methodology 5
6. Key Operations in Database 6-12
   1. Creation of tables
   2. Inserting values
   3. Data selection (Joins)
7. Key features in Database 13
   1. Transactions
8. Front-End (CODE) 14-17
9. Front-End 18-19
10. Conclusion 20
11. Bibliography 21
12. **INTRODUCTION**

In the dynamic landscape of educational institutions, the efficient management of hostel facilities stands as a cornerstone for providing students with a conducive living environment. As student populations grow and administrative tasks multiply, the importance of implementing robust Hostel Management Systems becomes increasingly evident. Thus, this project endeavors to introduce a comprehensive yet user-friendly solution, developed through the integration of Python, Tkinter, and MySQL technologies.

Hostel Management Systems serve as the backbone of operations, facilitating seamless processes for student accommodation, room allocation, and administrative record-keeping. Recognizing the significance of these systems, this project focuses on simplifying the complexities inherent in hostel management. By prioritizing core functionalities and intuitive design, it aims to provide administrators with a tool that streamlines operations while remaining accessible to users of varying technical expertise.

The essence of this project lies in its ability to automate key processes, particularly student registration and room allocation. Through a user-friendly interface built using Tkinter, administrators can effortlessly input student details and preferences, allowing the system to intelligently assign rooms based on availability and predefined criteria. Furthermore, the system enables comprehensive record viewing, encompassing essential aspects such as student information, departmental affiliations, hostel facilities, available rooms, and mess arrangements.

By striking a balance between functionality and simplicity, this project seeks to address the fundamental requirements of hostel management. It aims to empower administrators with the tools necessary to efficiently manage accommodation processes, optimize resource utilization, and enhance the overall living experience for students. Through its intuitive design and robust capabilities, the Simplified Hostel Management System aims to set a standard for effective hostel management solutions in educational institutions.

1. **HARDWARE AND SOFTWARE REQUIREMENTS**

**2.1 Hardware Requirements**:

1. Computer or server capable of running Python and MySQL.

2. Adequate storage space to store the Python scripts, Tkinter interface files, and MySQL database.

3. Sufficient RAM to handle the operations of the application effectively, depending on the expected workload and concurrent users.

**2.2 Software Requirements**:

1. Operating System: Compatible with Windows, Linux, or macOS.

2. Python: Version 3.8 installed on the system.

3. Tkinter: Python's standard GUI (Graphical User Interface) toolkit, usually included with Python installations.

4. MySQL: Database management system installed and configured on the system.

5. MySQL Connector/Python: Python driver for connecting to MySQL databases, can be installed via pip.

**3. ENTITY RELATIONSHIP DIAGRAM**

Here's the Entity Relationship (ER) diagram outlining the structure and connections within the project's database.

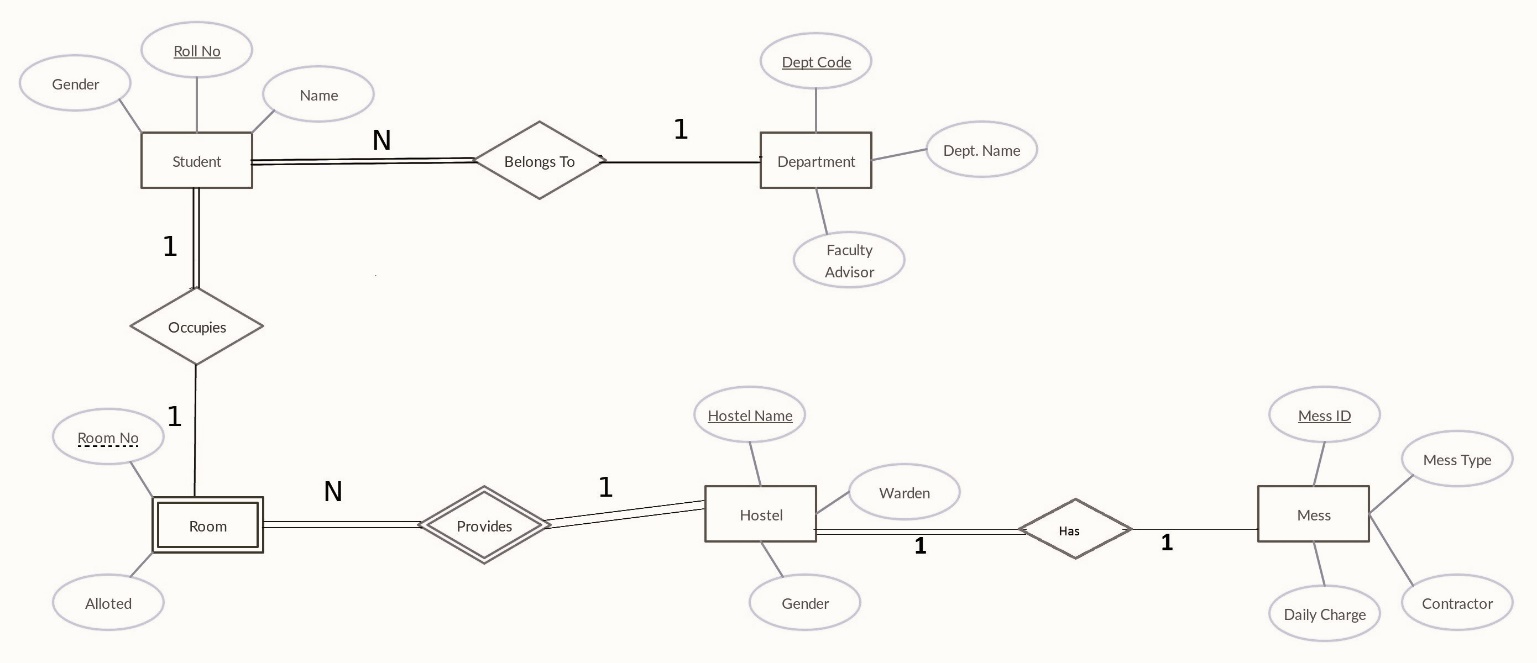
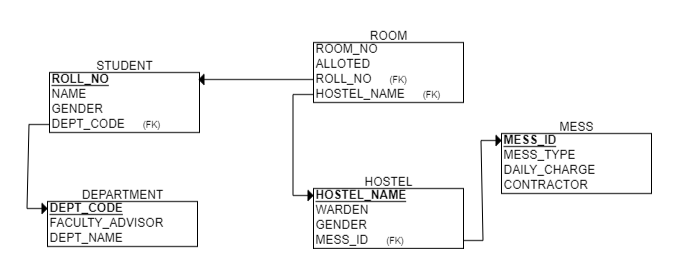


Figure 1. E-R diagram for Hostel management system.

**4. RELATIONAL MODEL**

Here's the Relational Model, depicting the organization of data tables and their relationships within the project's database.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No.** | **Name of Table** | **Primary Key** | **Foreign Key** | **Multi-valued Attribute** | **Single-Valued Attribute** |
| 1 | Student | Roll\_no | Department.Dept\_code | Name | Roll\_no, Name, Gender |
| 2 | Department | Dept\_code | - | - | Dept\_code, Dept\_name, Faculty\_Advisor |
| 3 | Room | Room\_no | Student.Roll\_no  Hostel.Hostel\_name | - | Room\_no, Alloted |
| 4 | Hostel | Hostel\_name | Mess.Mess\_id | - | Hostel\_name, Warden, Gender |
| 5 | Mess | Mess\_id | - | - | Mess\_id, Mess\_type, Daily\_charge, Contractor |

**5. METHODOLOGY**

1. Entity and Relations Identification: Entities such as Students, Departments, Hostels, Rooms, and Mess, along with their relationships, have been identified for the hostel management system.

2. Database Schema Design: A comprehensive database schema has been developed, defining tables for each entity and establishing relationships between them. The schema has been normalized to eliminate redundancy and dependency issues.

3. Normalization: Normalization techniques including First Normal Form (1NF), Second Normal Form (2NF), and Third Normal Form (3NF) have been applied to ensure efficient storage and retrieval of data while maintaining data integrity.

4. Implementation of Database Features: MySQL has been utilized to implement the database schema and features required for the hostel management system. This includes setting up tables, defining relationships using foreign keys, and optimizing queries for performance.

5. Frontend Development: The frontend of the application has been developed using Tkinter, incorporating GUI elements such as windows, buttons, and text fields to create an intuitive user interface. It seamlessly integrates with the backend database functionality for smooth interaction and data manipulation.

6. Transaction Management: Transaction management has been implemented to ensure the atomicity, consistency, isolation, and durability (ACID properties) of database operations. Transactions are handled effectively to maintain data integrity.

With these steps completed, the hostel management system has been systematically designed, developed, and implemented to effectively utilize entity-relationship modeling, database schema design, normalization techniques, and transaction management.

**6. KEY OPERATIONS IN DATABASE**

**6.1 CREATION OF TABLE**

--CREATING DATABASE

CREATE DATABASE HOSTEL\_MANAGEMENT;

USE HOSTEL\_MANAGEMENT

--CREATING TABLES

CREATE TABLE DEPARTMENT(

DEPT\_CODE INT PRIMARY KEY,

DEPT\_NAME VARCHAR(50) NOT NULL,

FACULTY\_ADVISOR VARCHAR(50));

CREATE TABLE MESS(

MESS\_ID INT PRIMARY KEY,

MESS\_TYPE VARCHAR(50),

DAILY\_CHARGES INT NOT NULL,

CONTRACTOR VARCHAR(50) NOT NULL);

CREATE TABLE HOSTEL(

HOSTEL\_NAME VARCHAR(50) PRIMARY KEY,

WARDEN VARCHAR(50) NOT NULL,

GENDER VARCHAR(10) NOT NULL,

MESS\_ID INT,

FOREIGN KEY (MESS\_ID) REFERENCES MESS(MESS\_ID));

CREATE TABLE STUDENT(

ROLL\_NO INT PRIMARY KEY,

NAME VARCHAR(50) NOT NULL,

GENDER VARCHAR(10) NOT NULL,

DEPT\_CODE INT,

FOREIGN KEY (DEPT\_CODE) REFERENCES DEPARTMENT(DEPT\_CODE));

CREATE TABLE ROOM(

ROOM\_NO INT PRIMARY KEY,

ALLOTED VARCHAR(1) NOT NULL,

ROLL\_NO INT,

FOREIGN KEY (ROLL\_NO) REFERENCES STUDENT(ROLL\_NO),

HOSTEL\_NAME VARCHAR(50),

FOREIGN KEY (HOSTEL\_NAME) REFERENCES HOSTEL(HOSTEL\_NAME));

**6.2 INSERTING VALUES**

--INSERTING VALUES

INSERT INTO DEPARTMENT (DEPT\_CODE, DEPT\_NAME, FACULTY\_ADVISOR) VALUES

(101, 'Computer Science', 'Dr. John Smith'),

(102, 'Electrical Engineering', 'Dr. Emily Johnson'),

(103, 'Mechanical Engineering', 'Dr. Michael Williams'),

(104, 'Civil Engineering', 'Dr. Sarah Brown'),

(105, 'Mathematics', 'Dr. David Wilson');

INSERT INTO MESS (MESS\_ID, MESS\_TYPE, DAILY\_CHARGES, CONTRACTOR) VALUES

(1, 'Vegetarian', 150, 'Green Foods Pvt. Ltd.'),

(2, 'Non-Vegetarian', 200, 'Delight Foods Pvt. Ltd.'),

(3, 'Vegan', 180, 'Health Foods Pvt. Ltd.');

INSERT INTO HOSTEL (HOSTEL\_NAME, WARDEN, GENDER, MESS\_ID) VALUES

('Alpha Hostel', 'Dr. Samantha White', 'Male', 1),

('Beta Hostel', 'Dr. Peter Green', 'Female', 2),

('Gamma Hostel', 'Dr. Rachel Brown', 'Male', 3);

INSERT INTO STUDENT (ROLL\_NO, NAME, GENDER, DEPT\_CODE) VALUES

(1001, 'John Doe', 'Male', 101),

(1002, 'Jane Smith', 'Female', 101),

(1003, 'Michael Johnson', 'Male', 102),

(1004, 'Emily Williams', 'Female', 102),

(1005, 'Christopher Brown', 'Male', 103),

(1006, 'Sarah Martinez', 'Female', 103),

(1007, 'Daniel Taylor', 'Male', 104),

(1008, 'Olivia Garcia', 'Female', 104),

(1009, 'Matthew Lee', 'Male', 105),

(1010, 'Emma Rodriguez', 'Female', 105),

(1011, 'David Hernandez', 'Male', 101),

(1012, 'Sophia Lopez', 'Female', 101),

(1013, 'James Gonzalez', 'Male', 102),

(1014, 'Ava Wilson', 'Female', 102),

(1015, 'Logan Perez', 'Male', 103),

(1016, 'Isabella Moore', 'Female', 103),

(1017, 'Benjamin Taylor', 'Male', 104),

(1018, 'Mia King', 'Female', 104),

(1019, 'Ethan Rivera', 'Male', 105),

(1020, 'Charlotte Scott', 'Female', 105);

-- INSERT statements for ROOM table

insert into ROOM(ROOM\_NO,ALLOTED,ROLL\_NO,HOSTEL\_NAME) VALUES

(101,'Y',1001,'Alpha Hostel'),

(104,'Y',1002,'Alpha Hostel'),

(105,'Y',1003,'Alpha Hostel'),

(109,'Y',1004,'Alpha Hostel'),

(110,'Y',1005,'Alpha Hostel'),

(115,'Y',1006,'Alpha Hostel'),

(116,'Y',1007,'Alpha Hostel'),

(117,'Y',1008,'Alpha Hostel'),

(118,'Y',1009,'Alpha Hostel'),

(119,'Y',1010,'Alpha Hostel');

insert into ROOM(ROOM\_NO,ALLOTED,HOSTEL\_NAME) VALUES

(102,'N','Alpha Hostel'),

(103,'N','Alpha Hostel'),

(106,'N','Alpha Hostel'),

(107,'N','Alpha Hostel'),

(108,'N','Alpha Hostel'),

(111,'N','Alpha Hostel'),

(112,'N','Alpha Hostel'),

(113,'N','Alpha Hostel'),

(114,'N','Alpha Hostel'),

(120,'N','Alpha Hostel');

insert into ROOM(ROOM\_NO,ALLOTED,ROLL\_NO,HOSTEL\_NAME) VALUES

(301,'Y',1016,'Gamma Hostel'),

(303,'Y',1017,'Gamma Hostel'),

(305,'Y',1018,'Gamma Hostel'),

(307,'Y',1019,'Gamma Hostel'),

(310,'Y',1020,'Gamma Hostel');

insert into ROOM(ROOM\_NO,ALLOTED,HOSTEL\_NAME) VALUES

(302,'N','Gamma Hostel'),

(304,'N','Gamma Hostel'),

(306,'N','Gamma Hostel'),

(308,'N','Gamma Hostel'),

(309,'N','Gamma Hostel');

insert into ROOM(ROOM\_NO,ALLOTED,ROLL\_NO,HOSTEL\_NAME) VALUES

(201,'Y',1011,'Beta Hostel'),

(203,'Y',1012,'Beta Hostel'),

(205,'Y',1013,'Beta Hostel'),

(207,'Y',1014,'Beta Hostel'),

(210,'Y',1015,'Beta Hostel');

insert into ROOM(ROOM\_NO,ALLOTED,HOSTEL\_NAME) VALUES

(202,'N','Beta Hostel'),

(204,'N','Beta Hostel'),

(206,'N','Beta Hostel'),

(208,'N','Beta Hostel'),

(209,'N','Beta Hostel');

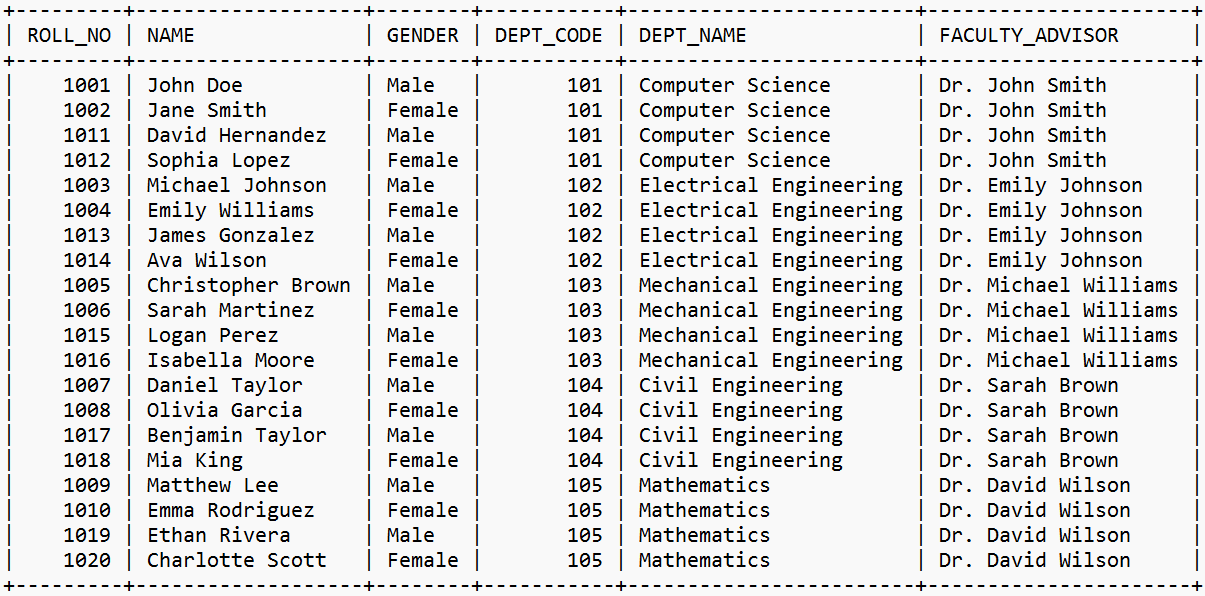
**6.3 DATA SELECTION (JOINS)**

-- INNER JOIN

SELECT ROLL\_NO, NAME, GENDER, department.DEPT\_CODE, department.DEPT\_NAME, department.FACULTY\_ADVISOR

FROM student

INNER JOIN department ON student.DEPT\_CODE = department.DEPT\_CODE;



-- LEFT OUTER JOIN

SELECT ROOM.ROOM\_NO,ROOM.HOSTEL\_NAME,STUDENT.ROLL\_NO,NAME,GENDER

from STUDENT

LEFT JOIN ROOM ON ROOM.ROLL\_NO = STUDENT.ROLL\_NO;

A table of names and numbers

Description automatically generated

-- SELF CARTESIAN PRODUCT

select A.MESS\_ID,A.MESS\_TYPE,B.DAILY\_CHARGES,B.CONTRACTOR

FROM MESS as A, MESS as B;

A table with numbers and letters

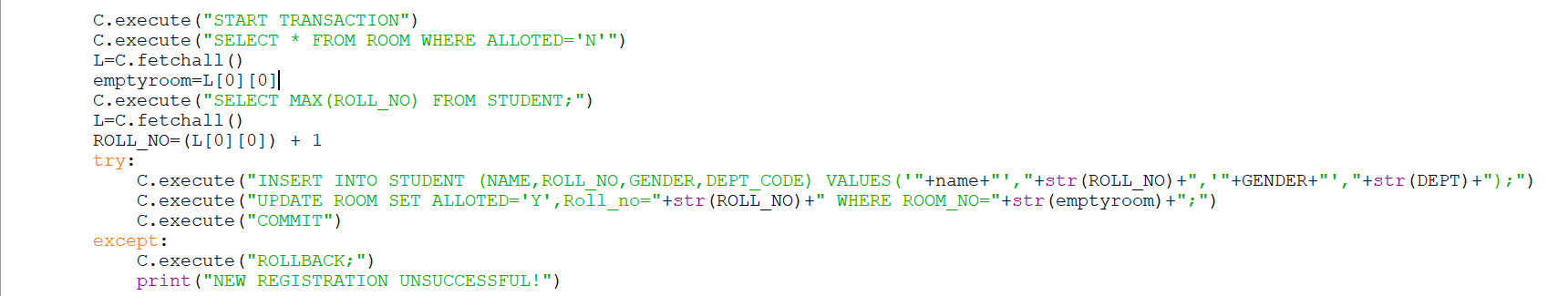
Description automatically generated

**7. KEY FEATURES IN DATABASE**

**7.1 TRANSACTIONS**

Transactions are used in the project in the front end functions to ensure ACID properties of the database and ensures uniformity in the database.

Below is the sample of transactions as used in the project.



**8. FRONT END (CODE)**

The frontend of the project encompasses the graphical user interface (GUI) developed using Tkinter, a Python library for creating interactive applications. It provides users with a visually intuitive platform to interact with the hostel management system, allowing for tasks such as student registration, room allocation, and record viewing. Tkinter's capabilities enable the creation of windows, buttons, text fields, and other UI elements, facilitating seamless navigation and operation of the system. Through its simplicity and versatility, the frontend enhances user experience and ensures efficient interaction with the underlying functionalities of the application.

Below are the snippets of the code for front-end.

A computer screen shot of a computer code

Description automatically generated with medium confidence

The Beginning of the code imports modules and establishes connection with MySQL for further interactions.

Codes for different screens of the application:

A computer code on a white background

Description automatically generated

A computer code with many colorful text

Description automatically generated with medium confidence

A computer screen with text

Description automatically generated with medium confidenceA computer screen shot

Description automatically generated A computer screen shot

Description automatically generated A computer screen shot of a computer code

Description automatically generatedA white background with colorful text

Description automatically generated

Code for Adding new students to the Hostel:A screenshot of a computer code

Description automatically generated

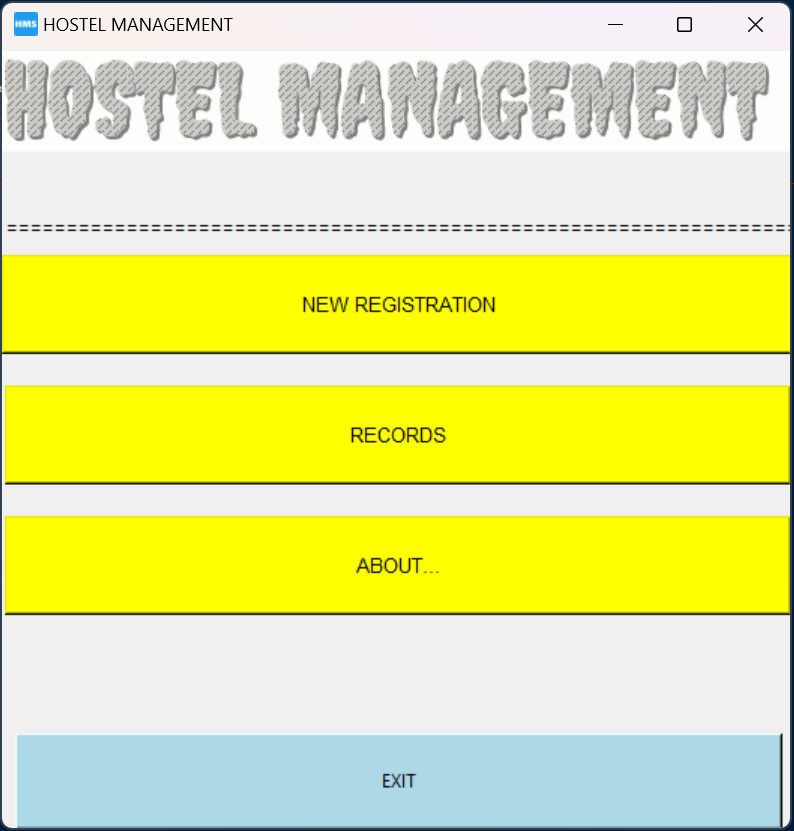
MAIN PAGE CODE:

A computer code on a white background

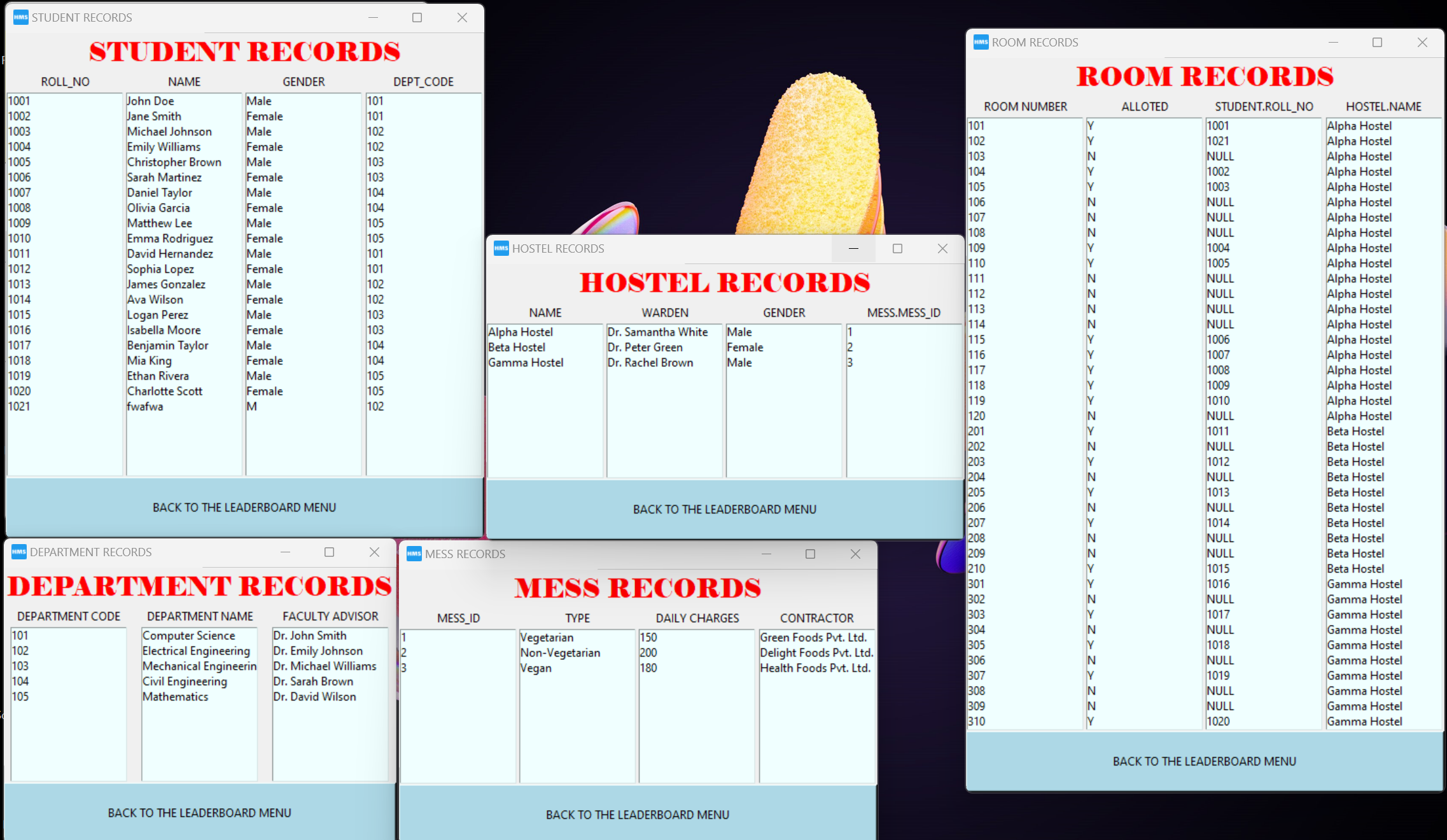
Description automatically generated

**9. FRONT-END**

**--Home Page and Records Menu**



**--Records Display Screens**



**--New Registration Screen**A screenshot of a computer

Description automatically generated

**--About Screen**



**10. CONCLUSION**

In conclusion, the development of the hostel management system using Python, Tkinter, and MySQL has provided a robust solution for efficiently managing student accommodation and related tasks. By incorporating entity-relationship modeling, database schema design, normalization, and transaction management, the system offers a user-friendly interface for administrators while ensuring data integrity and reliability. Moving forward, continuous feedback and iterative improvements will further enhance the system's functionality and usability, ultimately contributing to an enhanced hostel management experience for both administrators and students.

**11. BIBLIOGRAPHY**

1. Youtube(Tkinter)
2. Geekforgeeks.org (Modules and functions)
3. Stackoverflow.com (Modules and functions)
4. Textanim.com (HOSTEL\_MANAGEMENT.gif)
5. Favicon.io (ICON.ico)