HOSTEL MESS BOOKING SYSTEM

TEAM 3

MEMBERS:

1.PUNNYA NC

2.ADHIKESH RS

3.DEEPU PRASAD H

4.SREERAG

INSTITUTION: NSTI CALICUT

DATE:28/02/2025

ACKNOWLEDGMENT

We would like to express our sincere gratitude to our mentor and everyone who supported us throughout the development of the **Hostel Mess Booking System**. Special thanks to **Adwaith RS** for their invaluable guidance and collaboration.

ABSTRACT

The **Hostel Mess Booking System** is a simple yet effective solution for managing hostel meal bookings. The system allows students to register themselves, book meals for breakfast, lunch, and dinner, and view their meal bookings. This project utilizes a Python script and a MySQL database to handle the registration and booking process. The expected outcome is to streamline meal management, minimize errors, and provide students with a hassle-free dining experience.

TABLE OF CONTENTS:

INTRODUCTION:

Overview of the Project:

The **Hostel Mess Booking System** is designed to automate and simplify the process of booking meals in hostels. The system allows students to register themselves, select and book meals for the day (breakfast, lunch, and dinner), and check their booked meals. The database is used to store student information and meal bookings for efficient management.

Purpose and Importance:

The purpose of this system is to eliminate the manual intervention required for meal booking, thereby reducing human errors and improving efficiency. This system is essential for hostels to manage meal availability and ensure that students have a smooth and organized dining experience.

PROBLEM STATEMENT:

Hostels often face difficulties in managing meal bookings, leading to overbooking, underbooking, and wasted food. The manual process for meal bookings is error-prone and time-consuming, creating inefficiencies. This project aims to address these issues by automating the booking process.

Objectives:

- To allow students to register their details in the system.

- To provide a feature for students to book their meals for a given day (breakfast, lunch, dinner).
- To allow students to view their booked meals.
- To streamline hostel meal management and reduce errors.

Literature Review / Background Study

Existing Solutions or Related Work:

- Existing meal booking systems in educational institutions and hostels include basic functionalities for meal ordering, but many of them lack advanced features such as:
- Real-time meal availability.
- Easy tracking of past bookings.
- A user-friendly interface for students and hostel management.

- ***Gaps in Existing Systems:***

- Most current solutions do not integrate meal booking with student registration. They also lack features for students to view their meal history or preferences easily.

- ***How This Project Addresses Those Gaps:***

- This project solves the problem by integrating meal booking with student registration. Students can easily register, book their meals, and check their bookings for any given day. It provides a seamless experience for both students and hostel management.

SYSTEM REQUIREMENTS:

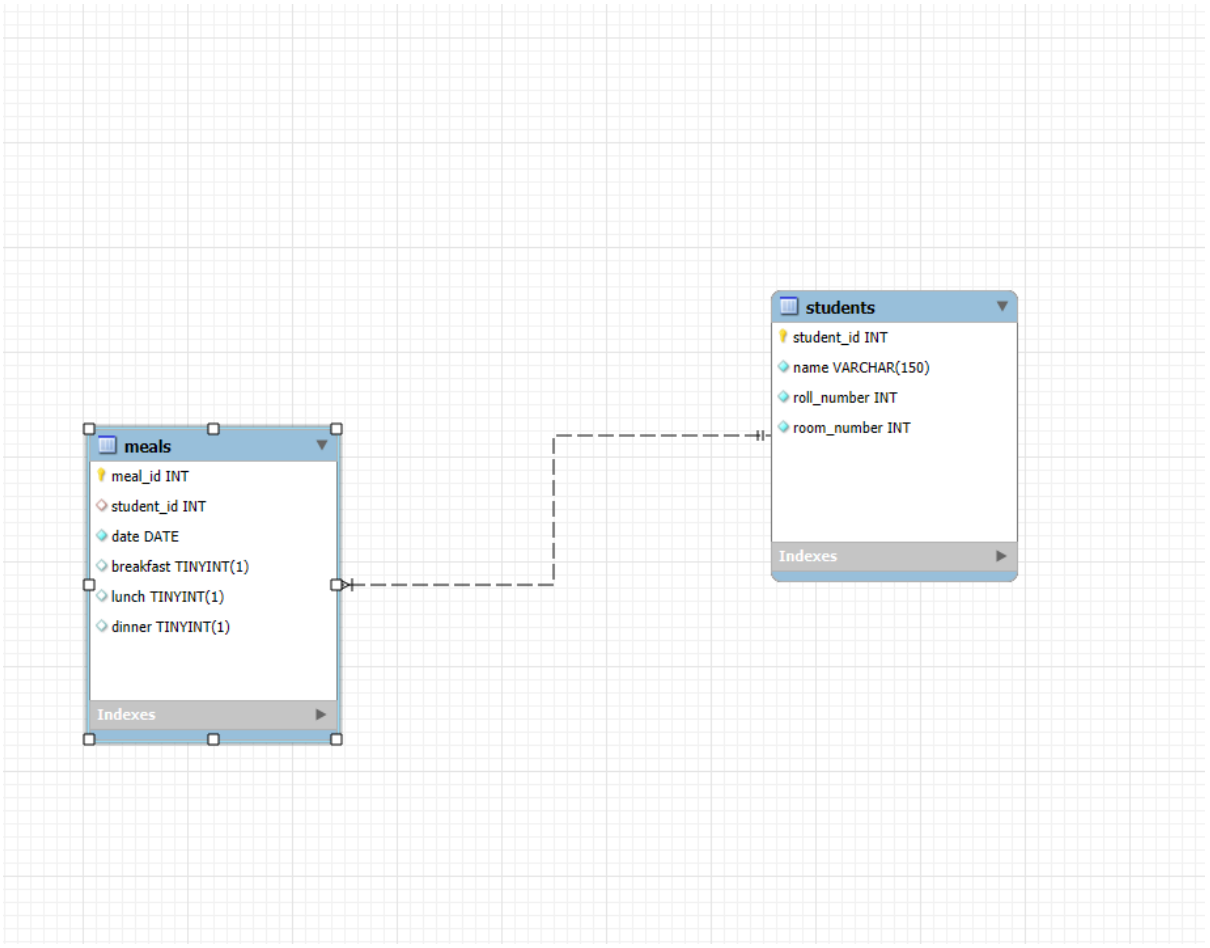
Hardware Requirements:

- Computers or servers to run the Python script and database.
- A network connection to connect to the MySQL database.

Software Requirements:

- **Programming Language**: Python 3
- **Database**: MySQL
- **Library/Framework**: MySQL Connector for Python
- **Operating System**: Linux/Windows (with MySQL installed)

ER DIAGRAM:



IMPLEMENTATION DETAILS

- **Python**: For scripting the entire system.
- **MySQL**: For database storage of student and meal data.
- **MySQL Connector**: Python library to interact with MySQL.

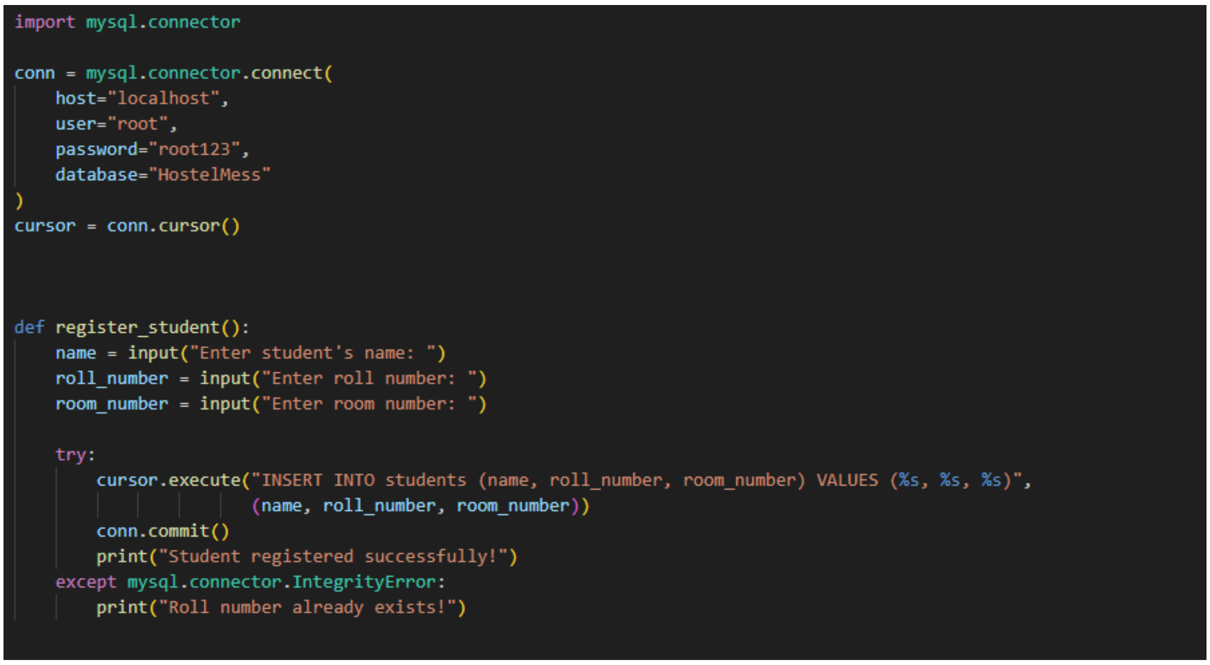
Modules and Their Functionalities:

1. **register_student()**: Registers a new student by capturing their name, roll number, and room number. It checks if the roll number is unique and stores the data in the database.

2. **book_meal():** Allows students to book meals (breakfast, lunch, and dinner) for a specific day. It checks if the student exists in the database and adds meal preferences.
3. **view_booked_meals():** Displays the meal bookings made by a student for a given roll number.

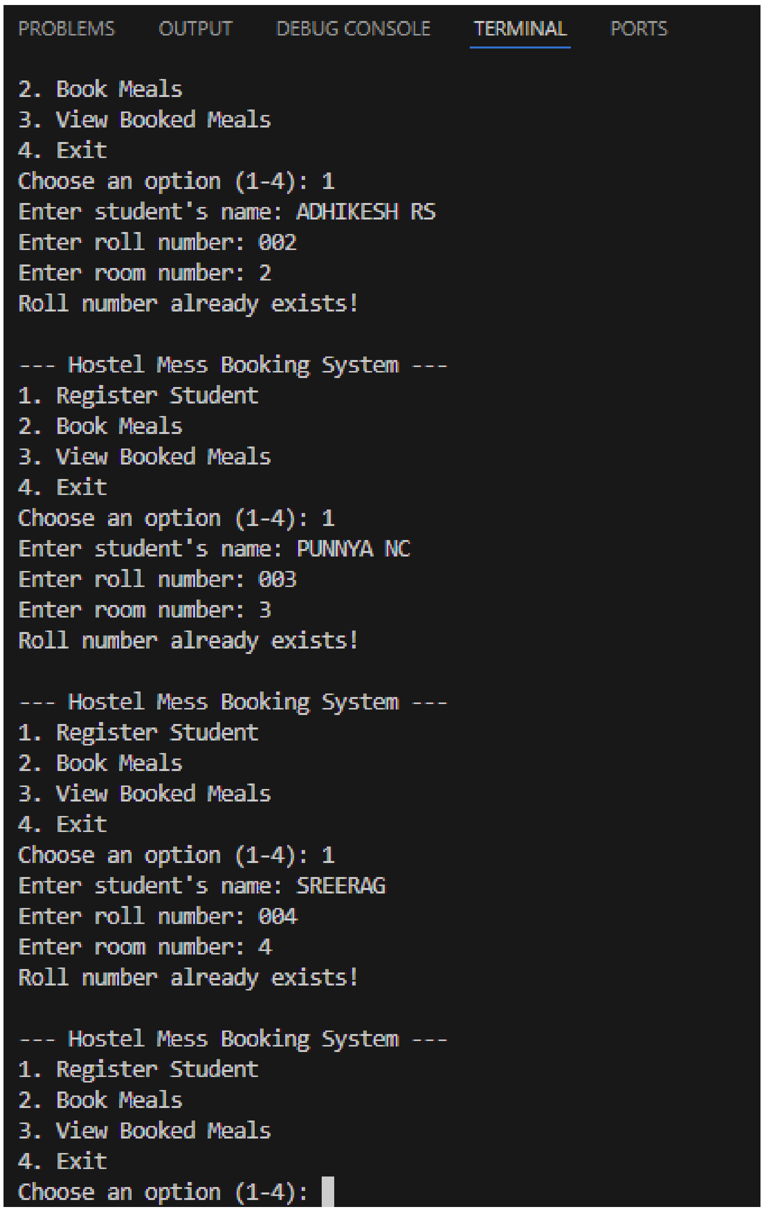
TESTING:

- Test Case 1: Register a student with a unique roll number



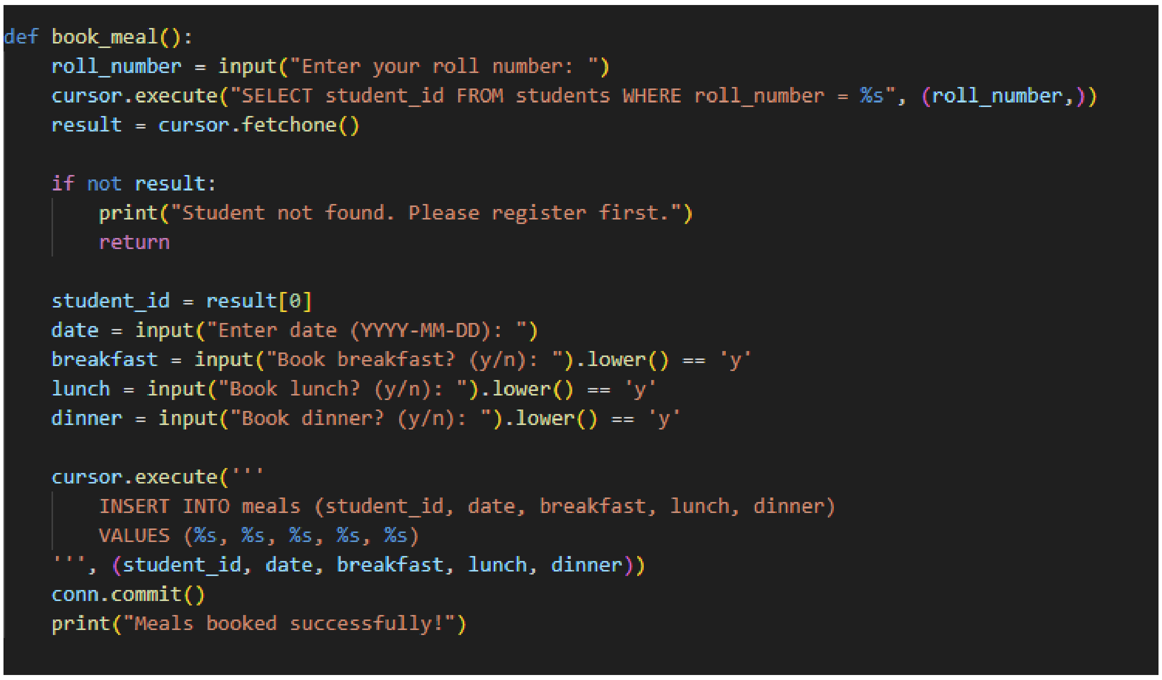
```python
import mysql.connector

conn = mysql.connector.connect(
    host="localhost",
    user="root",
    password="root123",
    database="HostelMess"
)
cursor = conn.cursor()



def register_student():
    name = input("Enter student's name: ")
    roll_number = input("Enter roll number: ")
    room_number = input("Enter room number: ")

    try:
        cursor.execute("INSERT INTO students (name, roll_number, room_number) VALUES (%s, %s, %s)",
                       (name, roll_number, room_number))
        conn.commit()
        print("Student registered successfully!")
    except mysql.connector.IntegrityError:
        print("Roll number already exists!")
```

OUT PUT:



```
2. Book Meals
3. View Booked Meals
4. Exit
Choose an option (1-4): 1
Enter student's name: ADHIKESH RS
Enter roll number: 002
Enter room number: 2
Roll number already exists!

--- Hostel Mess Booking System ---
1. Register Student
2. Book Meals
3. View Booked Meals
4. Exit
Choose an option (1-4): 1
Enter student's name: PUNNYA NC
Enter roll number: 003
Enter room number: 3
Roll number already exists!

--- Hostel Mess Booking System ---
1. Register Student
2. Book Meals
3. View Booked Meals
4. Exit
Choose an option (1-4): 1
Enter student's name: SREERAG
Enter roll number: 004
Enter room number: 4
Roll number already exists!

--- Hostel Mess Booking System ---
1. Register Student
2. Book Meals
3. View Booked Meals
4. Exit
Choose an option (1-4):
```

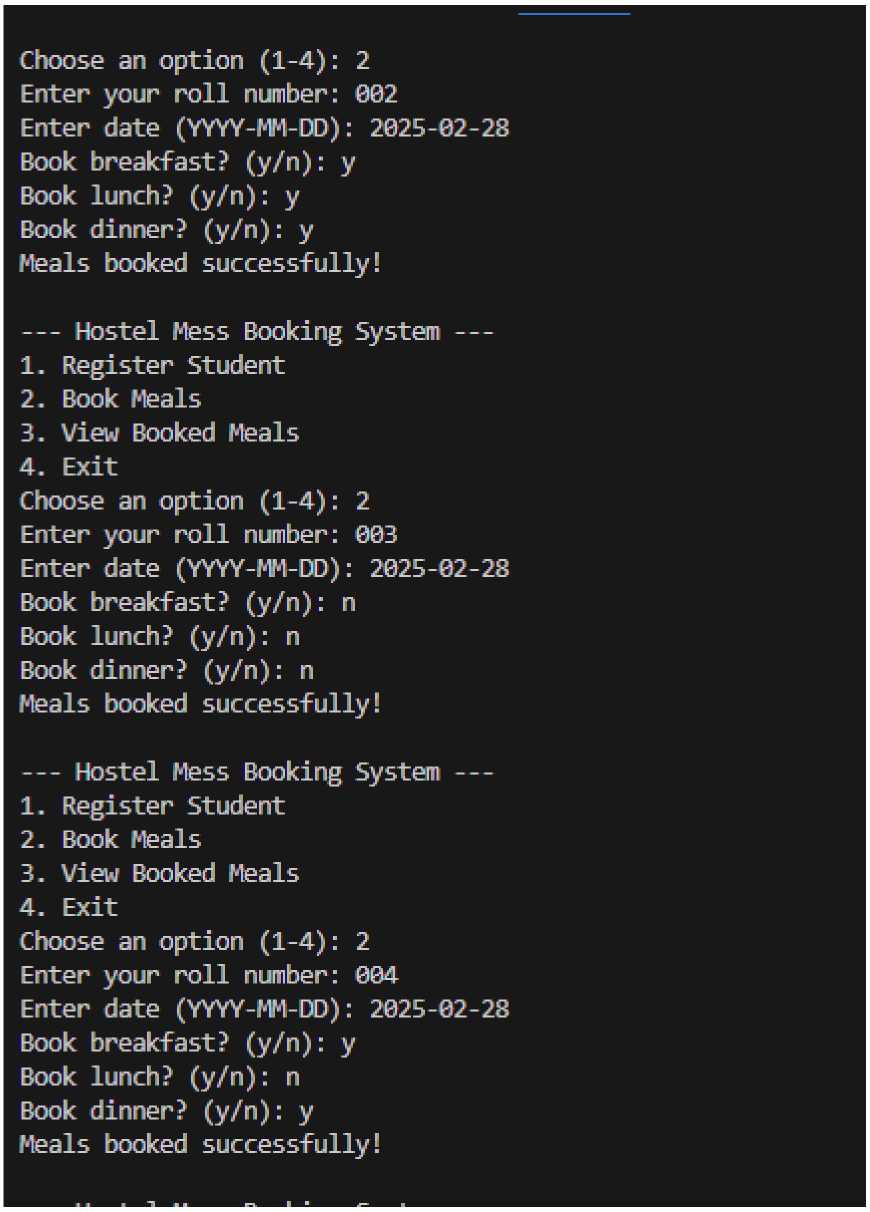
- Test Case 2: Book meals for a student



```
def book_meal():
    roll_number = input("Enter your roll number: ")
    cursor.execute("SELECT student_id FROM students WHERE roll_number = %s", (roll_number,))
    result = cursor.fetchone()

    if not result:
        print("Student not found. Please register first.")
        return

    student_id = result[0]
    date = input("Enter date (YYYY-MM-DD): ")
    breakfast = input("Book breakfast? (y/n): ").lower() == 'y'
    lunch = input("Book lunch? (y/n): ").lower() == 'y'
    dinner = input("Book dinner? (y/n): ").lower() == 'y'

    cursor.execute('''
        INSERT INTO meals (student_id, date, breakfast, lunch, dinner)
        VALUES (%s, %s, %s, %s, %s)
    ''', (student_id, date, breakfast, lunch, dinner))
    conn.commit()
    print("Meals booked successfully!")
```

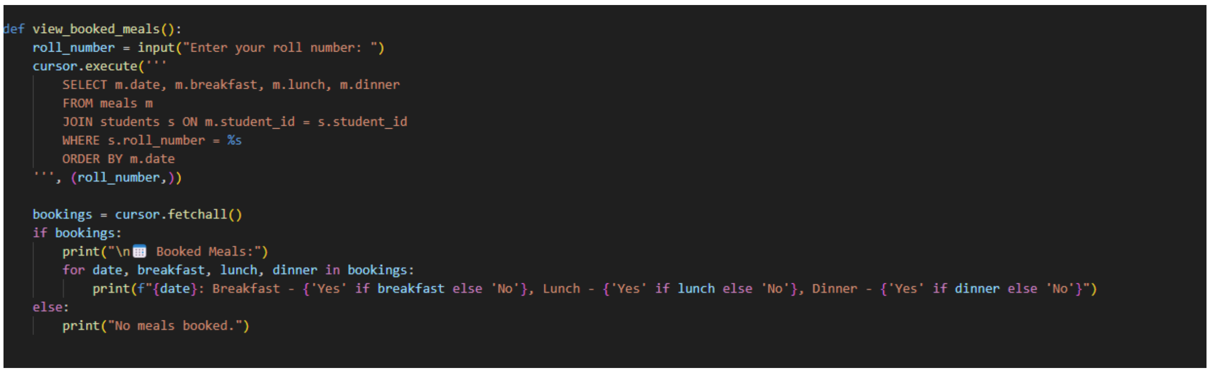
-

OUT PUT:



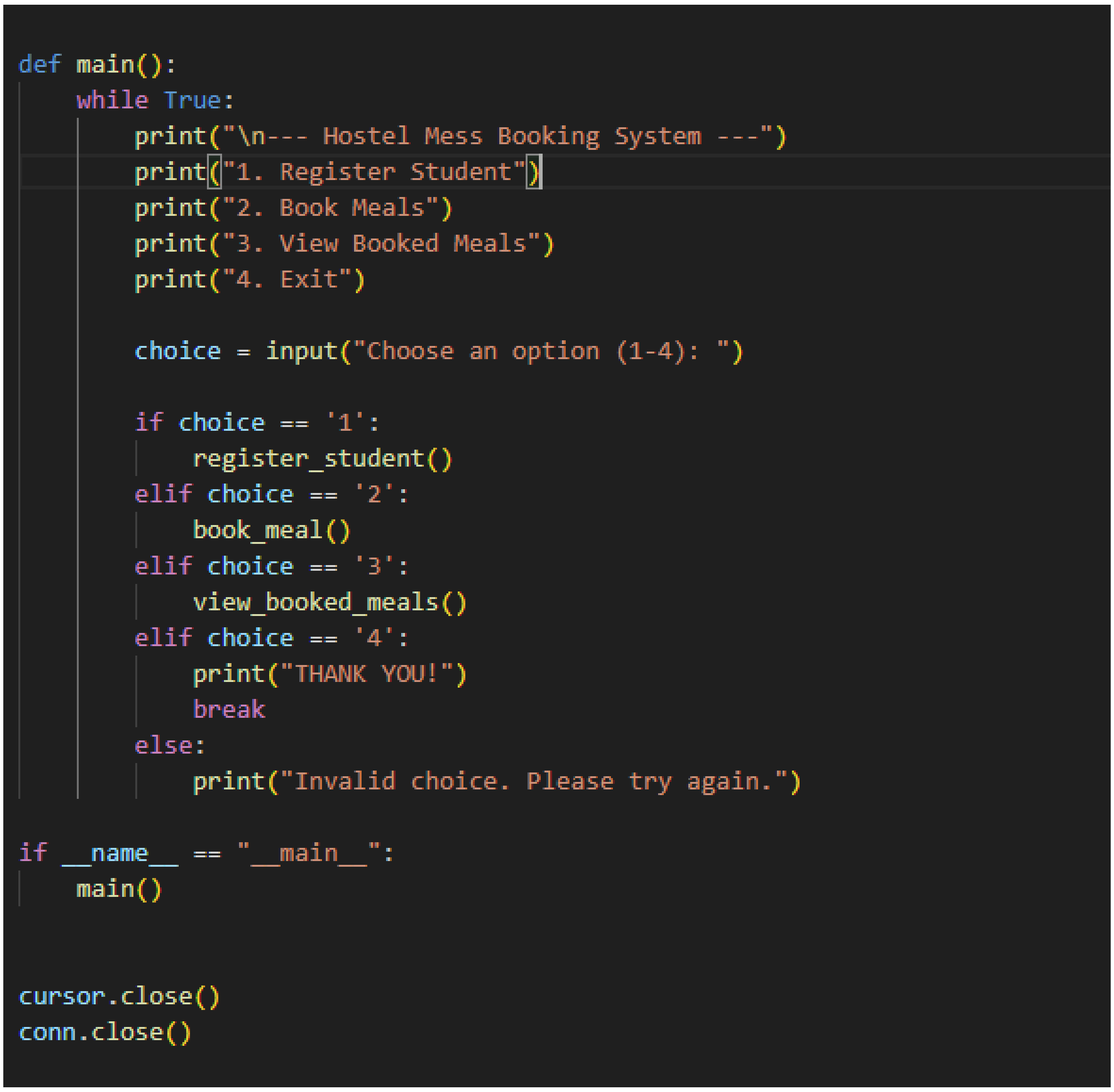
```
Choose an option (1-4): 2
Enter your roll number: 002
Enter date (YYYY-MM-DD): 2025-02-28
Book breakfast? (y/n): y
Book lunch? (y/n): y
Book dinner? (y/n): y
Meals booked successfully!

--- Hostel Mess Booking System ---
1. Register Student
2. Book Meals
3. View Booked Meals
4. Exit
Choose an option (1-4): 2
Enter your roll number: 003
Enter date (YYYY-MM-DD): 2025-02-28
Book breakfast? (y/n): n
Book lunch? (y/n): n
Book dinner? (y/n): n
Meals booked successfully!

--- Hostel Mess Booking System ---
1. Register Student
2. Book Meals
3. View Booked Meals
4. Exit
Choose an option (1-4): 2
Enter your roll number: 004
Enter date (YYYY-MM-DD): 2025-02-28
Book breakfast? (y/n): y
Book lunch? (y/n): n
Book dinner? (y/n): y
Meals booked successfully!
```

- **Test Case 3:** View booked meals for a student



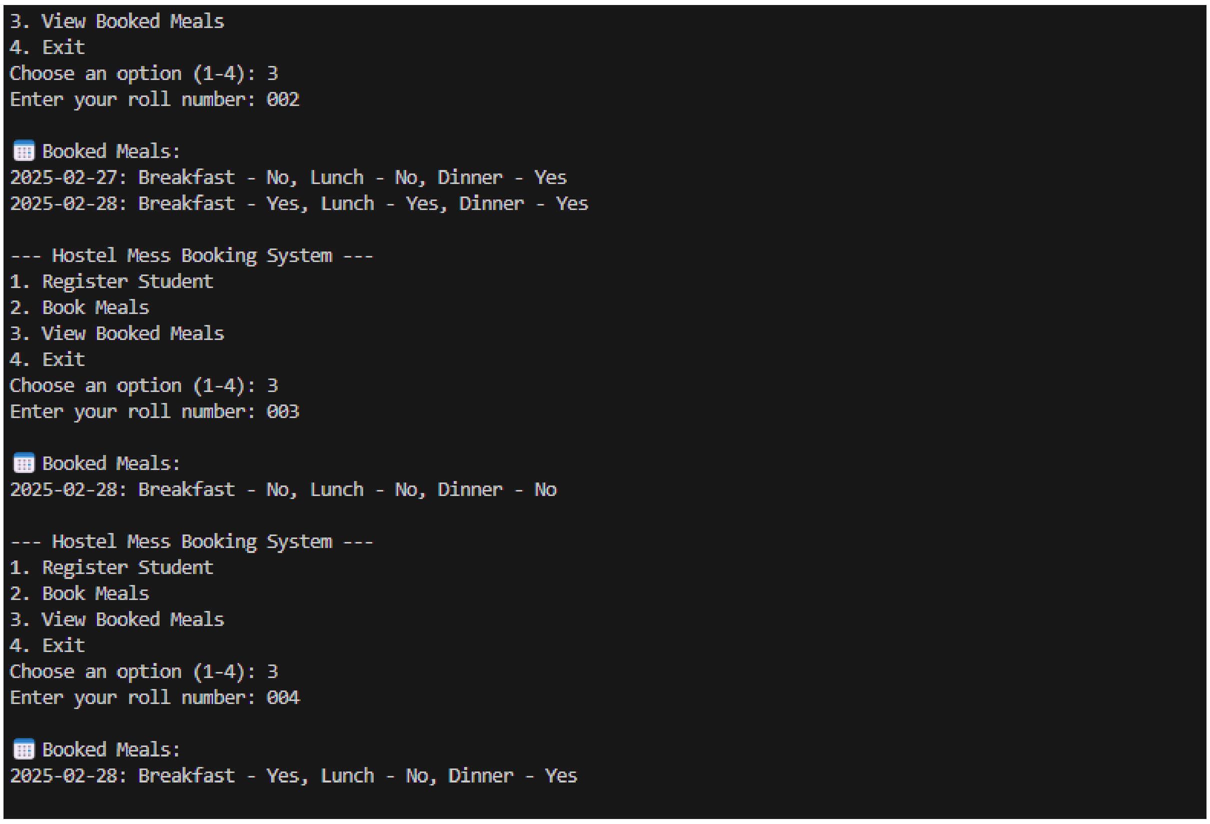
```python
def view_booked_meals():
    roll_number = input("Enter your roll number: ")
    cursor.execute('''
        SELECT m.date, m.breakfast, m.lunch, m.dinner
        FROM meals m
        JOIN students s ON m.student_id = s.student_id
        WHERE s.roll_number = %s
        ORDER BY m.date
    ''', (roll_number,))

    bookings = cursor.fetchall()
    if bookings:
        print("\n📅 Booked Meals:")
        for date, breakfast, lunch, dinner in bookings:
            print(f"{date}: Breakfast - {'Yes' if breakfast else 'No'}, Lunch - {'Yes' if lunch else 'No'}, Dinner - {'Yes' if dinner else 'No'}")
    else:
        print("No meals booked.")
```

CALL THE MAIN FUNCTION:



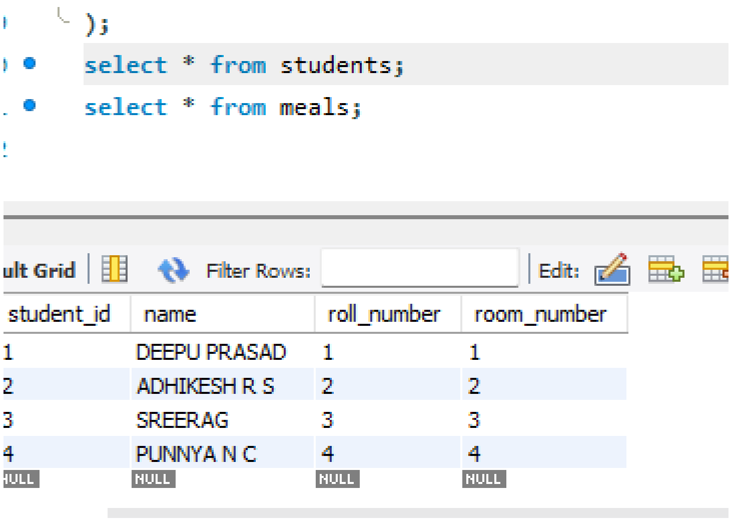
```python
def main():
    while True:
        print("\n--- Hostel Mess Booking System ---")
        print("1. Register Student")
        print("2. Book Meals")
        print("3. View Booked Meals")
        print("4. Exit")

        choice = input("Choose an option (1-4): ")

        if choice == '1':
            register_student()
        elif choice == '2':
            book_meal()
        elif choice == '3':
            view_booked_meals()
        elif choice == '4':
            print("THANK YOU!")
            break
        else:
            print("Invalid choice. Please try again.")

if __name__ == "__main__":
    main()


cursor.close()
conn.close()
```

OUT PUT:



```
3. View Booked Meals
4. Exit
Choose an option (1-4): 3
Enter your roll number: 002

📅 Booked Meals:
2025-02-27: Breakfast - No, Lunch - No, Dinner - Yes
2025-02-28: Breakfast - Yes, Lunch - Yes, Dinner - Yes

--- Hostel Mess Booking System ---
1. Register Student
2. Book Meals
3. View Booked Meals
4. Exit
Choose an option (1-4): 3
Enter your roll number: 003

📅 Booked Meals:
2025-02-28: Breakfast - No, Lunch - No, Dinner - No

--- Hostel Mess Booking System ---
1. Register Student
2. Book Meals
3. View Booked Meals
4. Exit
Choose an option (1-4): 3
Enter your roll number: 004

📅 Booked Meals:
2025-02-28: Breakfast - Yes, Lunch - No, Dinner - Yes
```

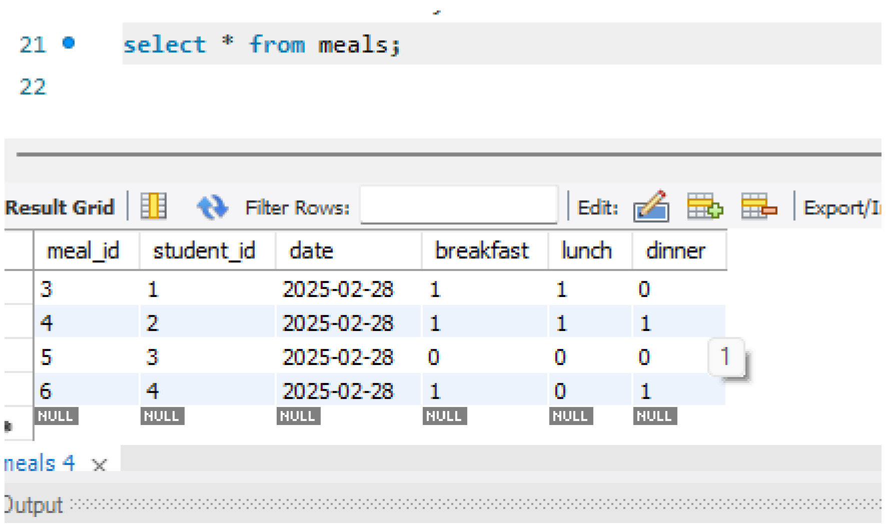
Student Table



```
);
select * from students;
select * from meals;
```

student_id	name	roll_number	room_number
1	DEEPU PRASAD	1	1
2	ADHIKESH R S	2	2
3	SREERAG	3	3
4	PUNNYA N C	4	4
NULL	NULL	NULL	NULL

Meals Table:



```
select * from meals;
```

meal_id	student_id	date	breakfast	lunch	dinner
3	1	2025-02-28	1	1	0
4	2	2025-02-28	1	1	1
5	3	2025-02-28	0	0	0
6	4	2025-02-28	1	0	1
NULL	NULL	NULL	NULL	NULL	NULL

Results and Discussion

Key Findings:

- The system functions as expected, allowing students to register and book meals efficiently.
- The database properly stores meal bookings, and students can view their meal bookings for any given date.

Performance Analysis:

- The system performs well with quick data retrieval from the MySQL database.
- No significant delays were observed during meal bookings or student registration.

CONCLUSION:

The **Hostel Mess Booking System** was successfully developed and tested. It automates the meal booking process, making it easier for students to register and book their meals while reducing administrative errors.

References:

- Python Documentation: https://docs.python.org/
- MySQL Documentation:https://dev.mysql.com/doc/