VISVASVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI, KARNATAKA -590018

PROJECT REPORT

ON

“HOSTEL MANAGEMENT SYSTEM”

Submitted in partial fulfillment of the requirements as a part of the DBMS Lab for the V Semester of degree of **Bachelor of Engineering in Computer Science and Engineering** *of Visvesvaraya Technological University, Belagavi*

Submitted by

Anushka Sinha **Chagi Manaswini**

1JS20CS033 **1JS20CS046**

UNDER THE GUIDANCE

Mrs Impana KP,

Assistant Professor, Dept of CSE,

JSSATE,Bengaluru

Department of Computer science and Engineering

JSS ACADEMY OF TECHNICAL EDUCATION

JSS MAHA VIDYAPEETHA , MYSURU

JSS ACADEMY OF TECHNICAL EDUCATION

JSS CAMPUS ROAD, UTTARAHALLI-KENGERI MAIN ROAD, BENGALURU-560060

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the project work entitled **“HOSTEL MANAGEMENT SYSTEM”** is a bonafide work carried out by **Anushka Sinha(1JS20CS033), Chagi Manaswini(1JS20CS046)** in partial fulfillment of the requirements for DBMS Laboratory with Mini Project (18CSL58) of the 5th Semester **Bachelor of engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belgaum** during the academic year 2021-2022.It is certified that all corrections and suggestions indicated for the Internal Assessment have been incorporated in the report deposited in the department Library. The project report has been approved as it satisfies the academic requirements in report of project work prescribed for the said degree.

Mrs.Impana KP **Dr.Mallikarjun,**

Assistant professor, Dept of CSE Professor and Head,Dept of CSE

JSSATE,Bengaluru

Name of the Examiners **Signature with Date:**

1)………………………… ………………………..

2)………………………… ……………………….

ABSTRACT

Hostel Management is a web application that is created for booking hostels for individuals. This will limit manual work and also make hostel allocation so much easier for students and hostel administrators. It manages data in the database and retrieves it whenever required

As the name specifies HOSTEL MANAGEMENT SYSTEM" is a software developed for managing various activities in the hostel. For the past few years the number of educational institutions are increasing rapidly. Thereby the number of hostels are also increasing for the accommodation of the students studying in this institution. And hence there is a lot of strain on the person who are running the hostel and software's are not usually used in this context. This particular project deals with the problems on managing a hostel and avoids the problems which occur when carried manually. Identification of the drawbacks of the existing system leads to the designing of computerized system that will be compatible to the existing system with the system Which is more user friendly and more GUI oriented

In our current era of automated systems with it being either software or hardware, it's not advisable to be using manual system. Hostels without a management system are usually done manually. Registration forms verification to other data saving processes are done manually and most at times, they are written on paper. Thus a lot of repetitions can be avoided with an automated system. The drawbacks of existing systems lead to the design of a computerized system that will help reduce a lot of manual inputs. With this system in place, we can improve the efficiency of the system, thus overcome the drawbacks of the existing manual system. This system is designed in favor of the hostel management which helps them to save the records of the students about their rooms and other things. It helps them from the manual work from which it is very difficult to find the record of the students and the mess bills of the students, and the information of about those ones who had left the hostel years before

ACKNOWLEDGEMENT

We express our humble pranams to His **Jagadaguru Sri Shivaratri Deshikendra Mahaswamji** for showering his blessings onus to receive good education and have a successful career.

The completion of any object involves the efforts of many people. We have been lucky enough to have received a lot of support from all the ends during the course of this project .So, we take the opportunity to express our gratitude to all whose guidance and encouragement helped us emerged successful.

We are thankful for the resourceful guidance, timely assistance and graceful of our guide Mrs.Imapana KP, Assistant Professor and Mrs Rashmi, Assistant Professor,

Department of Computer Science and Enginnering who has helped us in every aspect of our project work.

We express our sincere thanks to our beloved principal, for having supported us in all academic endeavours.

Last but not least, we would be immensely pleased to express out heartful thanks to all the teaching and non-teaching staff of the Department of CSE and our friends for their timely help, support and guidance.

Anushka Sinha

Chagi Manaswini

Table of Contents

Chapter Title **Page no**

List of Figures

Chapter 1

PREAMBLE

1.1 Introduction

A database is an organized collection of data. A relational database, more restrictively, is a collection of schemas, tables, queries, reports, views, and other elements. A databasemanagement system (DBMS) is a computer-software application that interacts with end- users, other applications, and the database itself to capture and analyze data. A general-purposeDBMS allows the definition, creation, querying, update, and administration of databases. Thereis a need for an application to make it easy for industries and trading companies to maintain their records and have a track of goods. **Hostel management system** is designed to manage all hostel activities like **hostel admissions, fees, room, mess allotment, hostel stores & generates related reports for smooth transactions**. It is also used to manage **monthly mess bill calculation, hostel staff payroll, student certificates**, etc.

Database Management System (DBMS)

Following the technology progress in the areas of processors, computer memory, computer storage, and computer networks, the sizes, capabilities, and performance of databases and their respective DBMSs have grown in orders of magnitude. The development of database technology can be divided into three eras based on data model or structure: navigational, SQL/relational, and post-relational. The two main early navigational data models were the hierarchical model, epitomized by IBM's IMS system, and the CODASYL model (network model), implemented in a number of products such as IDMS.

The relational model employs sets of ledger-style tables, each used for a different type of entity. Only in the mid-1980s did computing hardware become powerful enough to allow the wide deployment of relational systems (DBMSs plus applications). By the early 1990s, however, relational systems dominated in all large-scale data processing applications, and as of 2015 they remain dominant: IBM DB2, Oracle, MySQL, and Microsoft SQL Server are the top DBMS. The dominant database language, standardized SQL for the relational model, has influenced database languages for other data models.

1.1.1 Python

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of their features support functional programming and aspect-oriented programming (including metaprogramming and metaobjects)

- Python is used to short the program.
- Python is very flexible language.
- It provide more feature than any other programming language.
- Very easy to understand

1.1.2 Python Flask

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools

Applications that use the Flask framework include Pinterest and LinkedIn

Flask is a lightweight Python web framework that provides useful tools and features for **creating web applications in the Python Language**. It gives developers flexibility and is an accessible framework for new developers because you can build a web application quickly using only a single Python file

a backend this compact and controlled is capable of handling all the data processing required to support a full-featured frontend finance tracking app for fiscal fanatics

Advantages of Flask-based systems

- higher flexibility.
- higher compatibility with latest technologies.
- high scalability for simple web applications.
- technical experimentation.
- customization.
- slightly higher framework performance.
- easier to use for simple cases.
- smaller size of the code base

1.1.3 VS Code

Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

Visual Studio Code is **a streamlined code editor with support for development operations like debugging, task running, and version control**. It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDEs, such as Visual Studio IDE

Visual Studio Code is a free source code editor that **fully supports Python** and useful features such as real-time collaboration

VS Code is an excellent editor for students and other learner just getting started with HTML and CSS. This course focuses mainly on those students and learners who in the beginner to intermediate stages of learning to code with HTML, CSS, and Python.

1.1.4 Database-My SQL

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons. MySQL is released under an open-source license. So, we have nothing to pay to use it. MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages. MySQL uses a standard form of the well-known SQL data language.

MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc. It works very quickly and works well even with large data sets. MySQL is very friendly to PHP, the most appreciated language for web development. MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB) and is customizable.

1.1.5 Normalization

Normalization is a process of organizing the data in database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly. To overcome these anomalies, we need to normalize the data. There are 4 basic types of normalizations. They are:

- First normal form(1NF)
- Second normal form(2NF)
- Third normal form(3NF)
- Boyce & Codd normal form (BCNF)

First normal form (1NF) is defined as per rule as: an attribute (column) of a table cannot hold multiple values. It should hold only atomic values. This means that there shouldn’t be repetition of data in the tables.

A table is said to be in 2NF if the two conditions stated are satisfied. The table is in First normal form and all the non-prime attribute are dependent on the proper subset of any candidate key of table. The attribute that is not part of any candidate key are known as non-prime attribute.

A table design is said to be in 3NF if the table is in 2NF and Transitive functional dependency of non-prime attribute on any super key are removed.

Boyce Codd normal form (BCNF) is the advance version of 3NF that's why it is also referred as 3.5NF. BCNF is stricter than 3NF. A table complies with BCNF if it is in 3NF and for every functional dependency X->Y, X should be the super key of the table.

1.2 Objectives

Objectives of the **HOSTEL MANAGEMENT SYSTEM** are

1. Automating and streamlining the process of room allocation: The system can be used to track available rooms, assign rooms to guests, and manage waitlists.
2. Managing guest information: The system can be used to store and retrieve guest information, including contact details and booking history.
3. Managing payments and billing: The system can be used to process payments, generate invoices, and track financial transactions.
4. Managing inventory and supplies: The system can be used to track the inventory of supplies, such as linens and cleaning supplies, and generate purchase orders when stock is low.
5. Generating reports: The system can be used to generate various reports, such as occupancy reports and revenue reports, to help managers make informed decisions

1.3 Organization of Report

Chapter 1 provides the information about the basics of python and python-flask and MySQL. In **Chapter 2,** we discuss the software and hardware requirements to run the above applications. **Chapter 3** gives the idea of the project and its actual implementation. **Chapter 4** discusses about the results and discussions of the program. **Chapter 5** concludes by giving the direction for future enhancement.

1.4 Summary

The chapter discussed before is an overview about the Python, **Python flask** and **MySQL Database DBMS**. The scope of study and objectives of the project are mentioned clearly. The organization of the report is been pictured to increase thereadability. Further, coming up chapters depicts the use of various queries to implementvarious changes like insert, update, delete and also triggers to perform various functions.

Chapter 2

Requirement Specifications

2.1 SOFTWARE SPECIFICATION

- Operating System: Windows 11
- Front End: Python (Python-flask), VS Code
- Rear End: MySQL

2.2 HARDWARE SPECIFICATION

- Processor: x86 compatible processor with 1.7 GHz Clock Speed
- RAM: 512 MB or greater
- Hard Disk: 20 GB or grater
- Monitor: VGA/SVGA
- Keyboard: 104 keys standard
- Mouse: 2/3 button. Optical/Mechanical

2.3 USER SPECIFICATION

Every user:

- Should be comfortable with basic working of the computer
- Must have basic knowledge of English
- Must carry a login ID and password used for authentication

Chapter 3

System Design and Implementation

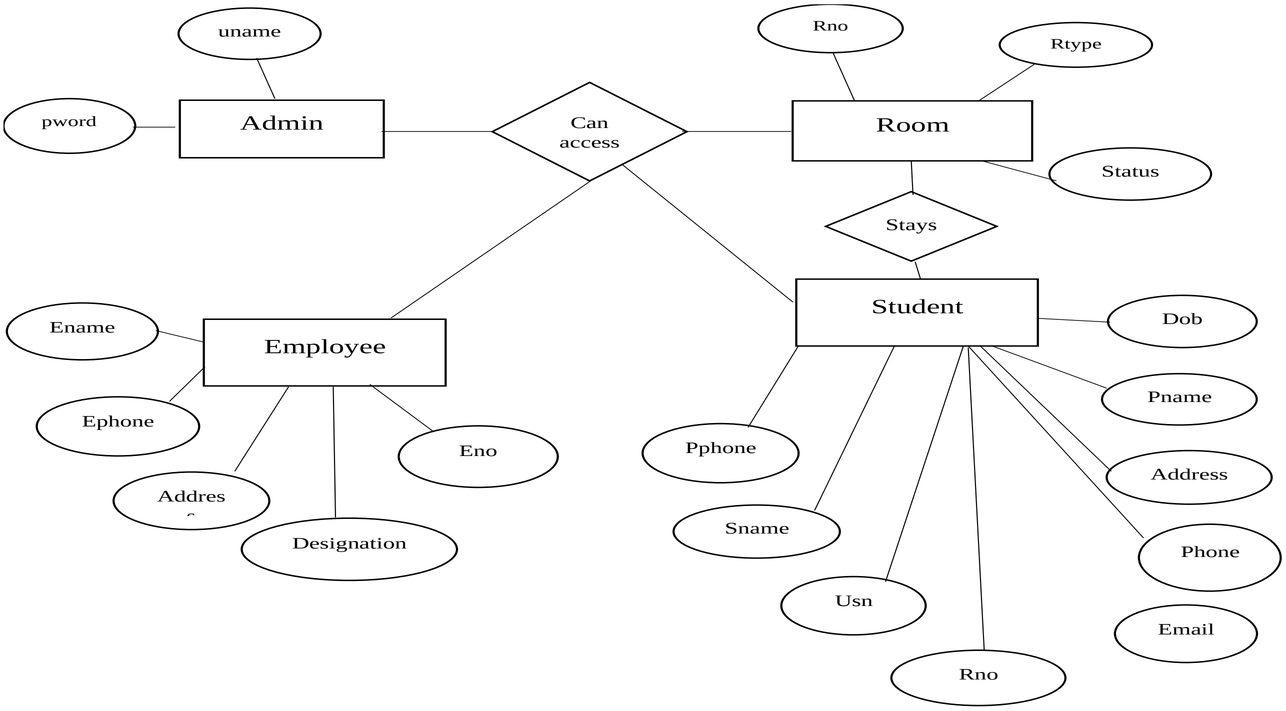
3.1 Introduction

Systems design is the process or art of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development.

This Project is implemented using VS Code, which is proven to be a very efficient tool in the field of Python programming. It is done under Windows11 platform. stdio.h library is used to create the objects and to translate them. Python programming language is used to implement the entire code. Interface to the program is provided with the help of MySQL Database**.**

3.2 ER Diagram

An entity–relationship model or the ER Diagram describes inter-related things of interest in a specific domain of knowledge. An ER model is composed of entity types and specifies relationships that can exist between instances of those entity types.

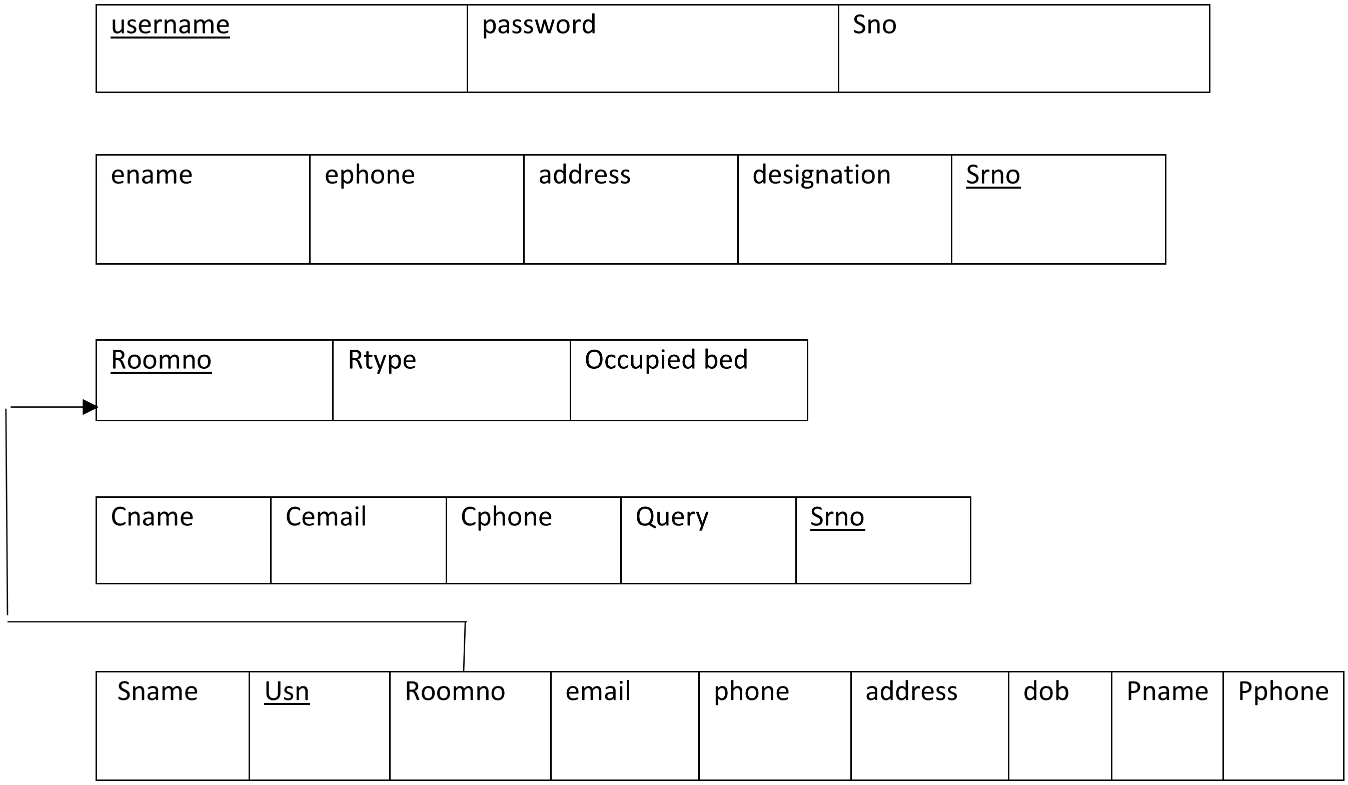


3.3 Schema Diagram

The schema diagram of a database system is its structure described in a formal languagesupported by the database management system (DBMS). The formal definition of a database schema is a set of formulas called integrity constraints imposed on a database.

The term "schema" refers to the organization of data as a blueprint of how the database is constructed. These integrity constraints ensure compatibility between parts of the schema. All constraints are expressible in the same language. A database can be considered a structure in realization of the database language. The states of a created conceptual schema are transformed into an explicit mapping, the database schema. This describes how real-world entities are modeled in the database.

Figure 3.2 Schema is defined for an Inventory Management System. All the various table used are described in the following schema. The necessary Primary key's and the corresponding Foreign keys are also represented.



username	password	Sno

ename	ephone	address	designation	Srno

Roomno	Rtype	Occupied bed

Cname	Cemail	Cphone	Query	Srno

Sname	Usn	Roomno	email	phone	address	dob	Pname	Pphone

3.4 Queries

The below mentioned are all the queries used to perform various tasks in MySQL such as insert, delete, update. A short description of the query is also provided.

3.4.1 Creating tables

ADMIN TABLE

```
Create table admin (
Username varchar (20),
Password varchar (20));
```

DESCRIBING ADMIN TABLE:

Field	Type	Null	Key	Default	Extra
Username	Varchar (20)	NOT NULL		NO	
Password	Varchar (20)	NOT NULL		NO	

EMPLOYEE TABLE

```
Create table employee (
Ename varchar (20),
Ephone varchar (20),
Address varchar (20),
Designation varchar (20),
srno integer);
```

DESCRIBING EMPLOYEE TABLE:

Field	Type	Null	Key	Default	Extra
Ename	Varchar (20)	NOT NULL		NO	
Ephone	Varchar (20)	NOT NULL		NO	
Address	Varchar (20)	NOT NULL		NO	
Designation	Varchar (20)	NOT NULL		NO	
srno	Varchar (20)	NOT NULL		NO	

ROOM TABLE

```
Create table room (
Rno integer,
Rtype varchar (20),
Status varchar (20));
```

DESCRIBING ROOM TABLE:

Field	Type	Null	Key	Default	Extra
Rno	Integer	NOT NULL		NO	
Rtype	Varchar (20)	NOT NULL		NO	
Status	Varchar (20)	NOT NULL		NO	

STUDENT TABLE

```
Create table student (
Sname varchar (20),
Usn varchar (20),
Rno integer,
Email varchar (20),
Phone integer,
Address varchar (20),
Dob varchar (20),
Pname varchar (20),
Pphone integer);
```

DESCRIBING STUDENT TABLE:

Field	Type	Null	Key	Default	Extra
Sname	Varchar(20)	NOT NULL		NO	
Usn	Varchar(20)	NOT NULL		NO	
rno	Varchar(20)	NOT NULL		NO	
Email	Varchar(20)	NOT NULL		NO	
Phone	Integer	NOT NULL		NO	
Address	Varchar(20)	NOT NULL		NO	
Dob	Varchar(20)	NOT NULL		NO	
Pname	Varchar(20)	NOT NULL		NO	
Pphone	Integer	NOT NULL		NO	

CONTACT TABLE

```
Create table contact (
Cname varchar (20),
Cemail varchar (20),
Cphone integer,
Query varchar (20),
Srno integer);
```

DESCRIBING CONTACT TABLE:

Field	Type	Null	Key	Default	Extra
Cname	Varchar(20)	NOT NULL		NO	
Cemail	Varchar(20)	NOT NULL		NO	
Cphone	integer	NOT NULL		NO	
Query	Varchar(20)	YES		NULL	
srno	Varchar(20)	NOT NULL		NO	

3.4.2 Inserting values into Tables

ADMIN TABLE

```
INSERT    INTO     ADMIN    VALUES (‘TOMMY’,123456);

INSERT    INTO     ADMIN    VALUES (‘GEETHA’,124456);

INSERT    INTO     ADMIN    VALUES (‘ANUSHKA’,323456);
```

INSERT INTO ADMIN VALUES (‘VARSHA’,123356);

INSERT INTO ADMIN VALUES (‘MYTHRI’,123156);

INSERT INTO ADMIN VALUES (‘RAKSHA’,123456);

INSERT INTO ADMIN VALUES (‘PRANATHI’,923456);

INSERT INTO ADMIN VALUES (‘DEEPIKA’,123466);

INSERT INTO ADMIN VALUES (‘VARSHITHA’,703456);

INSERT INTO ADMIN VALUES (‘SHRUTHI’,123906);

EMPLOYEE TABLE

INSERT INTO EMPLOYEE VALUES
(‘SHRUTHI’,1239065678,’DELHI’,’SECURITY’,10);

INSERT INTO EMPLOYEE VALUES
(‘SONU’,8790997491,’MYSORE’,’WARDEN’,20);

INSERT INTO EMPLOYEE VALUES
(‘KAPOOR’, 8790997492 ,’PATNA’,’SECURITY’,90);

INSERT INTO EMPLOYEE VALUES
(‘BINDHU’,9004455678,’DELHI’,’SECURITY’,10);

```
INSERT INTO EMPLOYEE VALUES
(‘SIRI’,9239065678,’GUJARATH’,’COOK’,50);

INSERT INTO EMPLOYEE VALUES
(‘RAMYA’,9089065678,’MUMBAI’,’COOK’,60);

INSERT INTO EMPLOYEE VALUES
(‘KALPANA’,8889065678,’KOLKATA’,’CLEANING’,30);

INSERT INTO EMPLOYEE VALUES
(‘HARITHA’,1239065698,’PUNJAB’,’CLEANING’,20);

INSERT INTO EMPLOYEE VALUES
(‘SHAMITHA’,8239065678,’UDUPI’,’VICEWARDEN’,80);

INSERT INTO EMPLOYEE VALUES
(‘SRAVANI’,990065678,’BANGALORE’,’CLEANING’,60);
```

STUDENT TABLE

```
INSERT INTO STUDENT VALUES
(‘SHRUTHI’,’1JS20CS001’,100,SHRU@GMAIL.COM,1234567890,’PAKIAN’
,’01-JAN-2000’,’MAHESH’,9087654321);

INSERT INTO STUDENT VALUES
(‘RANI’,’1JS20CS002’,101,RANI@GMAIL.COM,9234567890,’GOA’,’02-
JAN-2000’,’SURESH’,8087654321);

INSERT INTO STUDENT VALUES
```

```
(‘ANUSHKA’,’1JS20CS031’,103,ANU@GMAIL.COM,9934567890,’BIHAR’,’
26-JAN-2000’,’RAJESH’,9088654321);

INSERT INTO STUDENT VALUES
(‘KHUSHI’,’1JS201S001’,100,KHUSHI@GMAIL.COM,9434567890,’KOLKA
TA’,’01-FEB-2003’,’PAHESH’,7087654321);

INSERT INTO STUDENT VALUES
(‘AMRUTHA’,’1JS20CS033’,103,AMMU@GMAIL.COM,1094567890,’GOA’,
’03-AUG-2003’,’RAMESH’,3087654321);

INSERT INTO STUDENT VALUES
(‘THANUSHA’,’1JS20CS170’,104,TANU@GMAIL.COM,6734567890,’KOLA
R’,’04-MAR-2000’,’DATTA’,9017654321);

INSERT INTO STUDENT VALUES
(‘YASHAWINI’,’1JS20ME001’,200,YASHU@GMAIL.COM,1234767890,’SHI
VMOGA’,’03-OCT-2000’,’PRAJWAL’,9088654321);

INSERT INTO STUDENT VALUES
(‘MEGHANA’,’1JS20EC023’,120,MEGH@GMAIL.COM,7034567890,’ANDH
RZ’,’02-MAY-2004’,’RAHUL’,9087654320);

INSERT INTO STUDENT VALUES
(‘SUREKHA’,’1JS20CS020’,105,SUSUU@GMAIL.COM,9100167890,’ANDA
AMAN’,’21-JUL-2002’,’SAKETH’,2087654321);

INSERT INTO STUDENT VALUES
(‘PINKY’,’1JS20CIV34’,202,PINKS@GMAIL.COM,1234097890,’DAMAN’,’0
3-NOV-2015’,’KITTU’,6087654321);
```

ROOM TABLE

```
INSERT INTO ROOM VALUES (101,’SINGLE’,1);
INSERT INTO ROOM VALUES (102,’SINGLE’,1);
INSERT INTO ROOM VALUES (103,’SINGLE’,1);
INSERT INTO ROOM VALUES (104,’SINGLE’,0);
INSERT INTO ROOM VALUES (105,’SINGLE’,0);
INSERT INTO ROOM VALUES (106,’TRIPLE’,1);
INSERT INTO ROOM VALUES (107, ‘TRIPLE’,3);
INSERT INTO ROOM VALUES (108,’ TRIPLE’,3);
INSERT INTO ROOM VALUES (109,’ TRIPLE’,2);
INSERT INTO ROOM VALUES (110,’ TRIPLE’,1);
INSERT INTO ROOM VALUES (111,’ TRIPLE’,2);
INSERT INTO ROOM VALUES (112,’ TRIPLE’,0);
INSERT INTO ROOM VALUES (113,’ TRIPLE’,3);
```

INSERT INTO ROOM VALUES (114,’ TRIPLE’,2);

INSERT INTO ROOM VALUES (115,’ DOUBLE’,1);

INSERT INTO ROOM VALUES (116,’ DOUBLE’,1);

INSERT INTO ROOM VALUES (117,’ DOUBLE’,2);

INSERT INTO ROOM VALUES (118,’ DOUBLE’,2);

INSERT INTO ROOM VALUES (119,’ DOUBLE’,2);

INSERT INTO ROOM VALUES (120,’ DOUBLE’,1);

CONTACT TABLE

INSERT INTO CONTACT VALUES (‘KITTU’,’KITTU@GMAIL.COM’,’GOOD’,20);

INSERT INTO CONTACT VALUES (‘ASHA’,’ASHA@GMAIL.COM’,’GOOD’,10);

INSERT INTO CONTACT VALUES (‘PANDU’,’PANDU@GMAIL.COM’,’BAD’,40);

INSERT INTO CONTACT VALUES (‘SIRI’,’SIRI@GMAIL.COM’,’AVERAGE’,60);

INSERT INTO CONTACT VALUES

```
(‘PONKI’,’PONKS@GMAIL.COM’,’GOOD’,30);

INSERT INTO CONTACT VALUES
(‘LAKSHMI’,’LAKSHU@GMAIL.COM’,’VERY GOOD’,90);

INSERT INTO CONTACT VALUES
(‘AKSHAY’,’AKSHAY@GMAIL.COM’,’GOOD’,40);

INSERT INTO CONTACT VALUES
(‘ANANYA’,’ANU@GMAIL.COM’,’VERYGOOD’,80);

INSERT INTO CONTACT VALUES
(‘ANUSHA’,’ANUSHA@GMAIL.COM’,’GOOD’,30);

INSERT INTO CONTACT VALUES
(‘KASHI’,’KASH@GMAIL.COM’,’BAD’,70);
```

3.4.3 Query to demonstrate Deleting the value in the table:

```
@app.route("/delete/<string:srno>", methods=['GET', 'POST'])
def delete(srno):
    student=Student.query.filter_by(srno=srno).first()
    db.session.delete(student)
    db.session.commit()
    return redirect('/sdetails')
```

Description: This query is used to delete a row in vendor table when the name enteredin text field matches in the vendor table of database.

3.4.4 Stored Procedures

A stored procedure is nothing more than prepared SQL code that the developer saves so the application can reuse the code over and over again. So, if the developer thinks about a query that you write over and over again, instead of having to write that query each time you would save it as a stored procedure and then just call the stored procedure to execute the SQL code that you saved as part of the stored procedure.

Query

```
select roomno, type from room where roomno NOT in( SELECT roomno from student)
```

Procedure:

```
DELIMITER /

CREATE PROCEDURE REQ () BEGIN
SELECT * FROM SNAME; END
/
```

Description: Here whenever the **Stored Procedure REQ ()** is called the SNAME table will be displayed.

3.4.5 Triggers

Triggers are stored programs, which are automatically executed or fired when some events occur. Triggers are stored into database and invoked repeatedly, when specific condition match.

Trigger:

```
INSERT INTO trigr VALUES(null,NEW.roomno,NEW.usn,'Registered',NOW())
```

Description: This trigger is triggered when the quantity becomes 0 in the vendor table andthe row gets deleted.

3.4.6 Views

In SQL, a view is a virtual table based on the result-set of an SQL statement. A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.

You can add SQL statements and functions to a view and present the data as if the data were coming from one single table.

Query:

```
select `r`.`roomno` AS `roomno`,`r`.`type` AS `type`,count(`hms`.`student`.`usn`) AS
`occ_bed`
from (`hms`.`room` `r` left join `hms`.`student` on(`r`.`roomno` =
`hms`.`student`.`roomno`)) group by `r`.`roomno`,`r`.`type`
```

3.5 Pseudo Code

Pseudocode is an informal high-level description of the operating principle of a computer program or other algorithm. It uses the structural conventions of a normal programming language, but is intended for human reading rather than machine reading

3.5.1 Algorithm for login

```
Step 1: BEGIN

Step 2: Enter username and password

Step 3: Verify the credentials entered with that in the login table

Step 4: If Credentials match, then proceed to the Admin page Else show login
```

failed

Step 5: End if

Step 6: END

3.5.2 Algorithm for Table Display

Step1: BEGIN

Step 2: Establish connection with the database using the username and password ofthe database.

Step 3: Define the select query to retrieve all the values from the DBMS

Step 4: Define DefaultTableModel for the table and fetch the details from thedatabase.

Step 5: END

3.5.3 Algorithm for Insert

Step 1: BEGIN

Step 2: Get all the necessary values required for insertion into variable defined in themethod.

Step 3: Define the query for insertion as stated above.

Step 4: Execute the Query using the **(Select * from)** the required table.

Step 5: END

3.5.4 Algorithm for update

Step 1: BEGIN

Step 2: Get all the necessary values required for updating the values into the variable defined in the method

Step 3: UPDATE table name
SET column1 = value1, column2 = value2, ...
WHERE condition;

Step 4: Define the Query for Updating as stated above.

Step 5: Execute the Query using the **executeUpdate()** method defined.

Step 6: END

3.5.5 Algorithm for Delete

Step 1: BEGIN

Step 2: Get the model number of the instrument which is to be deleted into a variabledefined in the method.

Step 3: Delete from table_name where condition;

Step 4: Define the Query for deleting as stated above.

Step 5: Execute the Query using the **executeUpdate()** method defined .

Step 6: END

Chapter 4

Results and Discussions

The project is compiled and executed using VS Code and MySQL. We have put in few screen shots here to show the working of our application.

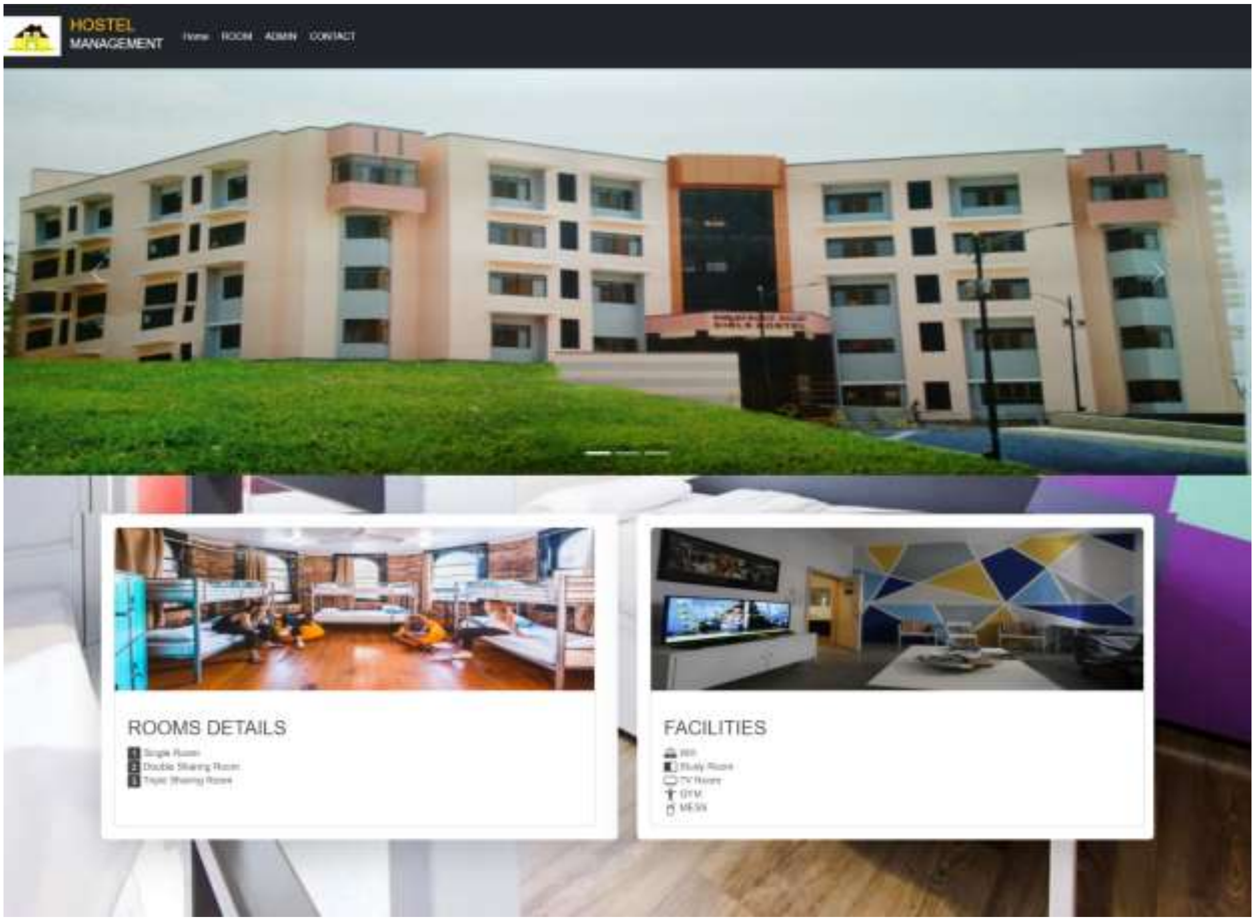
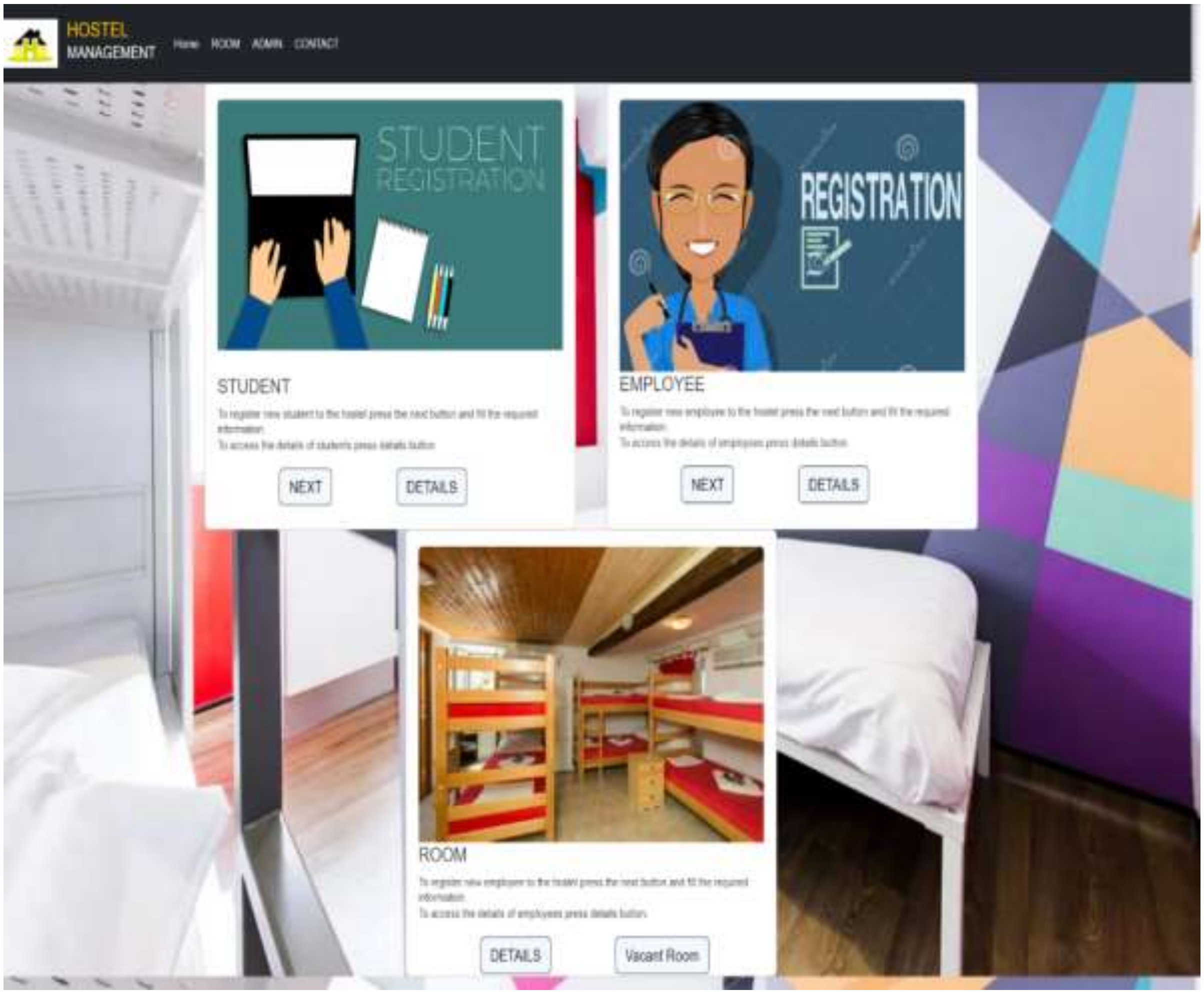


Figure 4.1 Home Page



STUDENT

To register new student to the hostel press the next button and fill the required information.
To access the details of student's press details button.

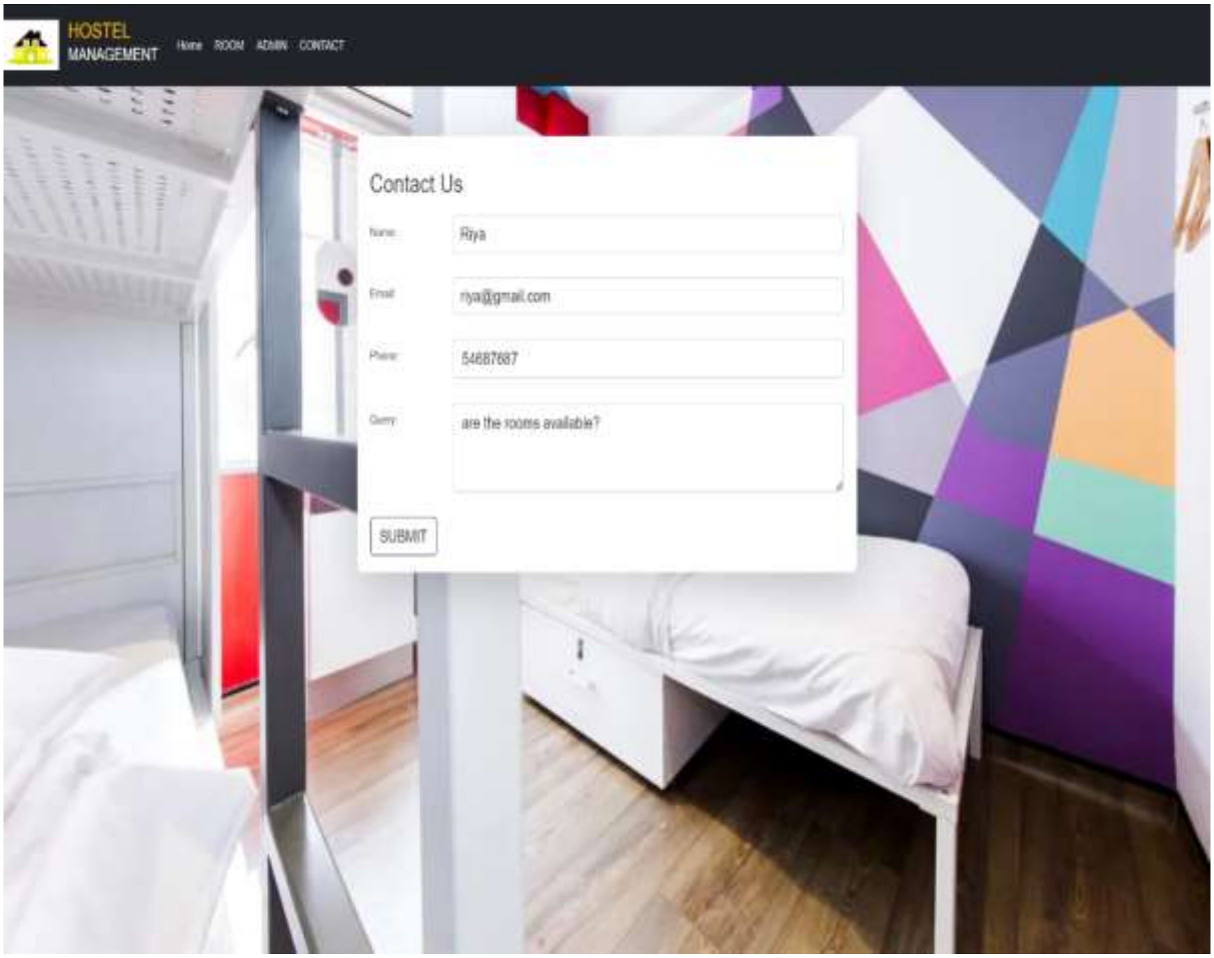
EMPLOYEE

To register new employee to the hostel press the next button and fill the required information.
To access the details of employees press details button.

ROOM

To register new employee to the hostel press the next button and fill the required information.
To access the details of employees press details button.

Figure 4.2 Admin Redirect Page



Contact Us

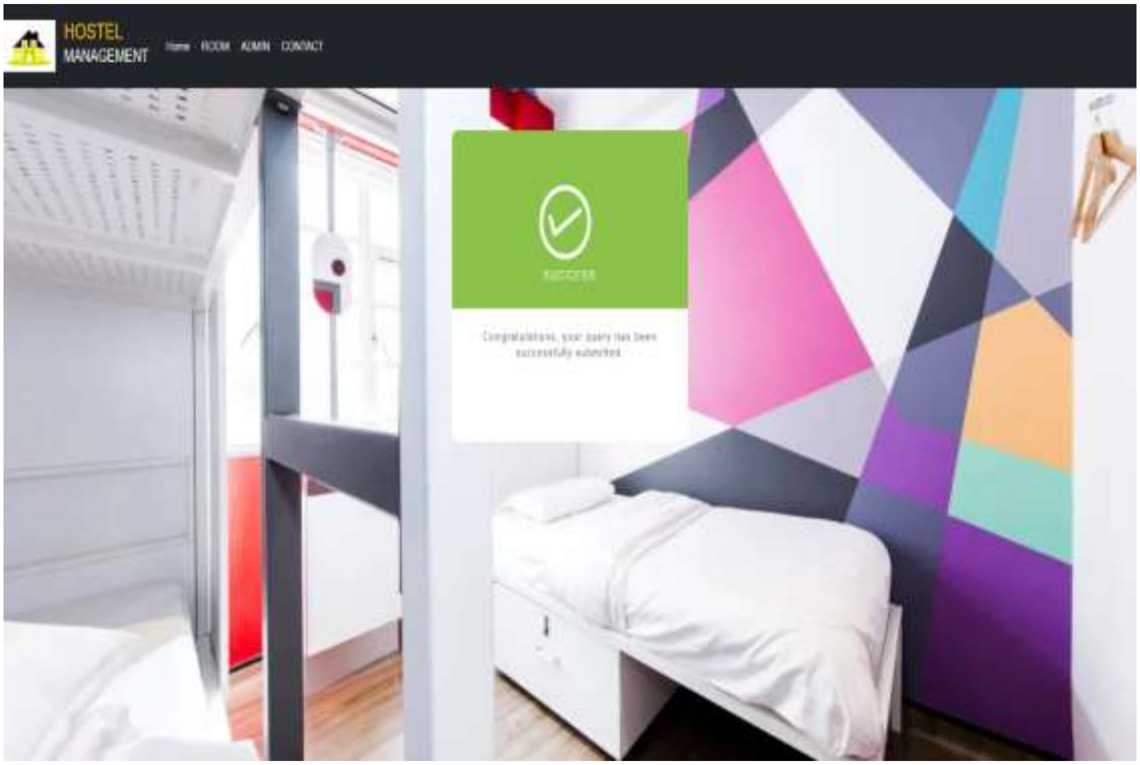
Name: Riya

Email: riya@gmail.com

Phone: 54687687

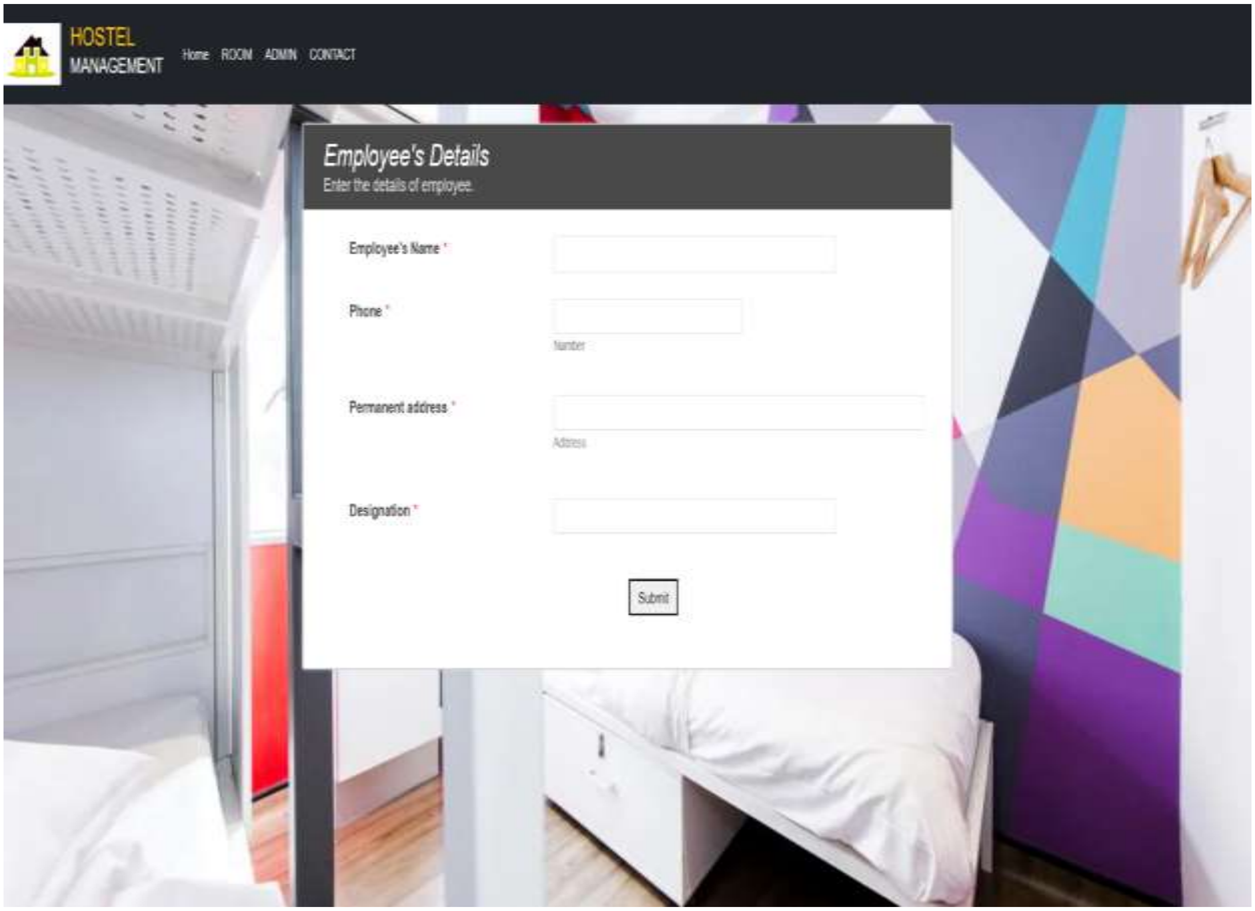
Query: are the rooms available?

Figure 4.3 Contact Page



Congratulations, your query has been successfully submitted

Figure 4.4 Contact confirmation page



Employee's Details

Enter the details of employee.

Employee's Name *

Phone *

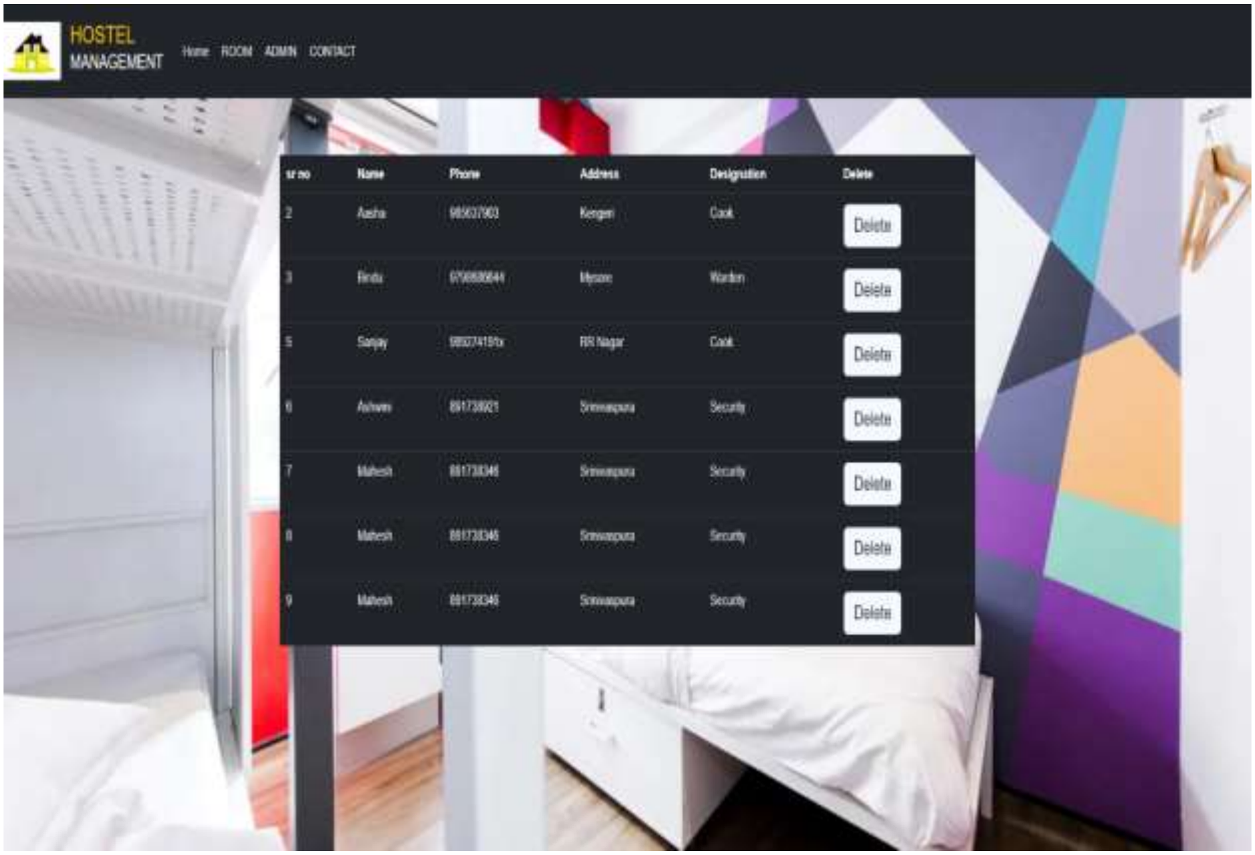
Number

Permanent address *

Address

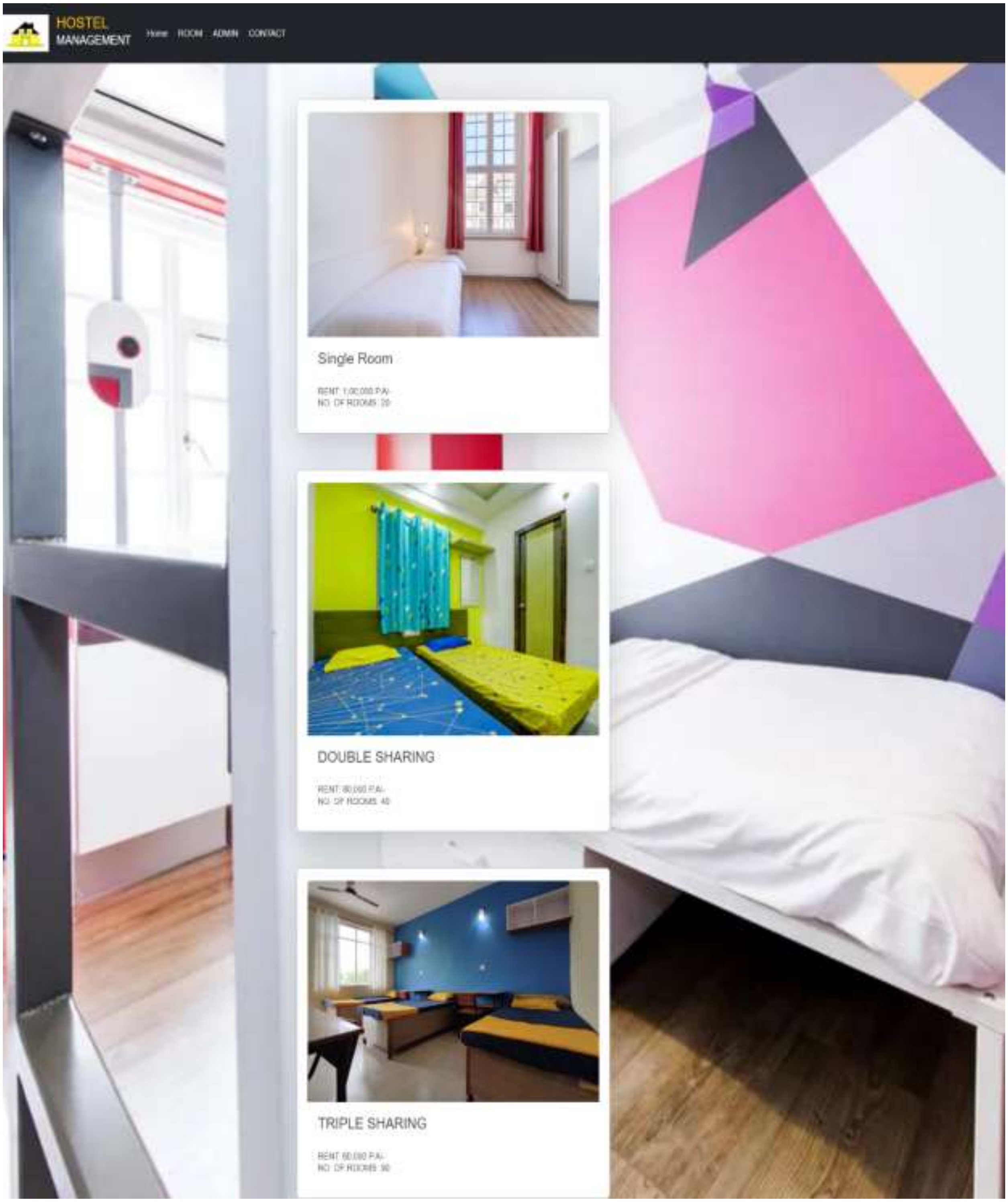
Designation *

Figure 4.5 Employee Page



sr no	Name	Phone	Address	Designation	Delete
2	Aasha	985637903	Kengeri	Cook	Delete
3	Bindu	9798808644	Mysore	Warden	Delete
5	Sanjay	989274191x	RR Nagar	Cook	Delete
6	Ashwini	891738921	Srinivaspura	Security	Delete
7	Mahesh	891738346	Srinivaspura	Security	Delete
8	Mahesh	891738346	Srinivaspura	Security	Delete
9	Mahesh	891738346	Srinivaspura	Security	Delete

Figure 4.6 Employee details Page



Single Room

RENT: 1,00,000 P.A/-
NO. OF ROOMS: 20

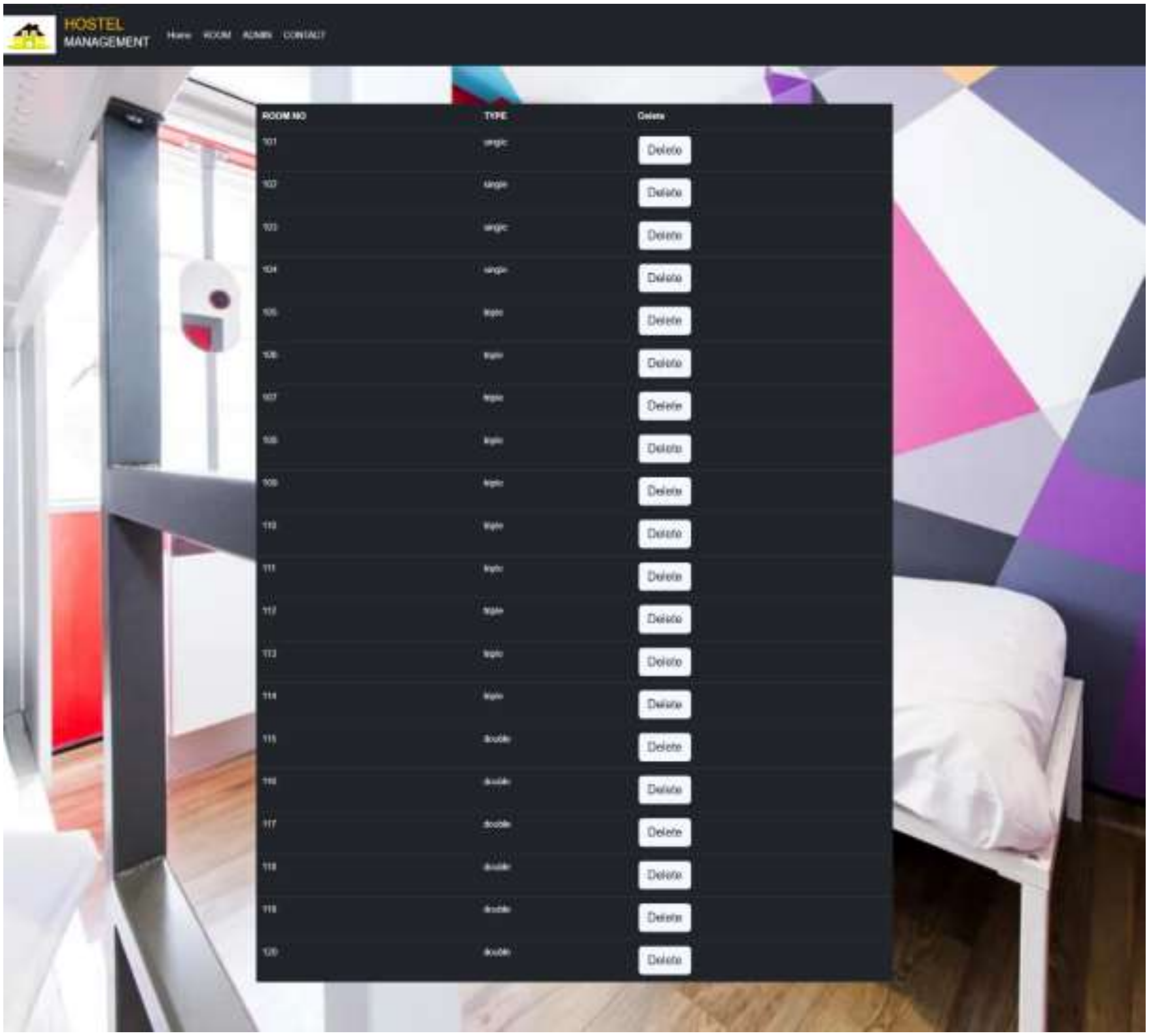
DOUBLE SHARING

RENT: 80,000 P.A/-
NO. OF ROOMS: 40

TRIPLE SHARING

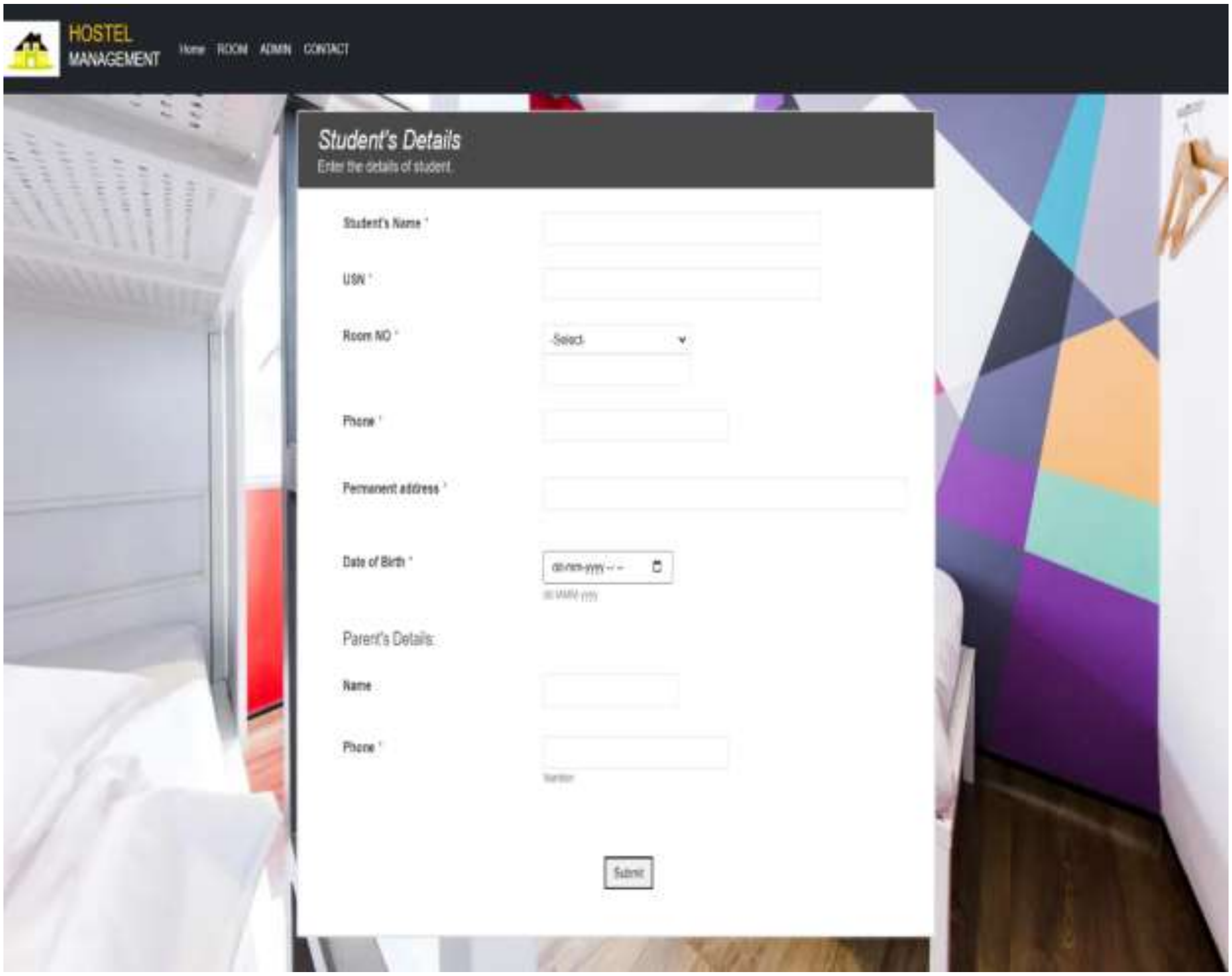
RENT: 60,000 P.A/-
NO. OF ROOMS: 90

Figure 4.7 Room Page



ROOM NO	TYPE	Delete
101	single	Delete
102	single	Delete
103	single	Delete
104	single	Delete
105	triple	Delete
106	triple	Delete
107	triple	Delete
108	triple	Delete
109	triple	Delete
110	triple	Delete
111	triple	Delete
112	triple	Delete
113	triple	Delete
114	triple	Delete
115	double	Delete
116	double	Delete
117	double	Delete
118	double	Delete
119	double	Delete
120	double	Delete

Figure 4.8 Room details Page



Student's Details

Enter the details of student.

Student's Name *

USN *

Room NO *

Phone *

Permanent address *

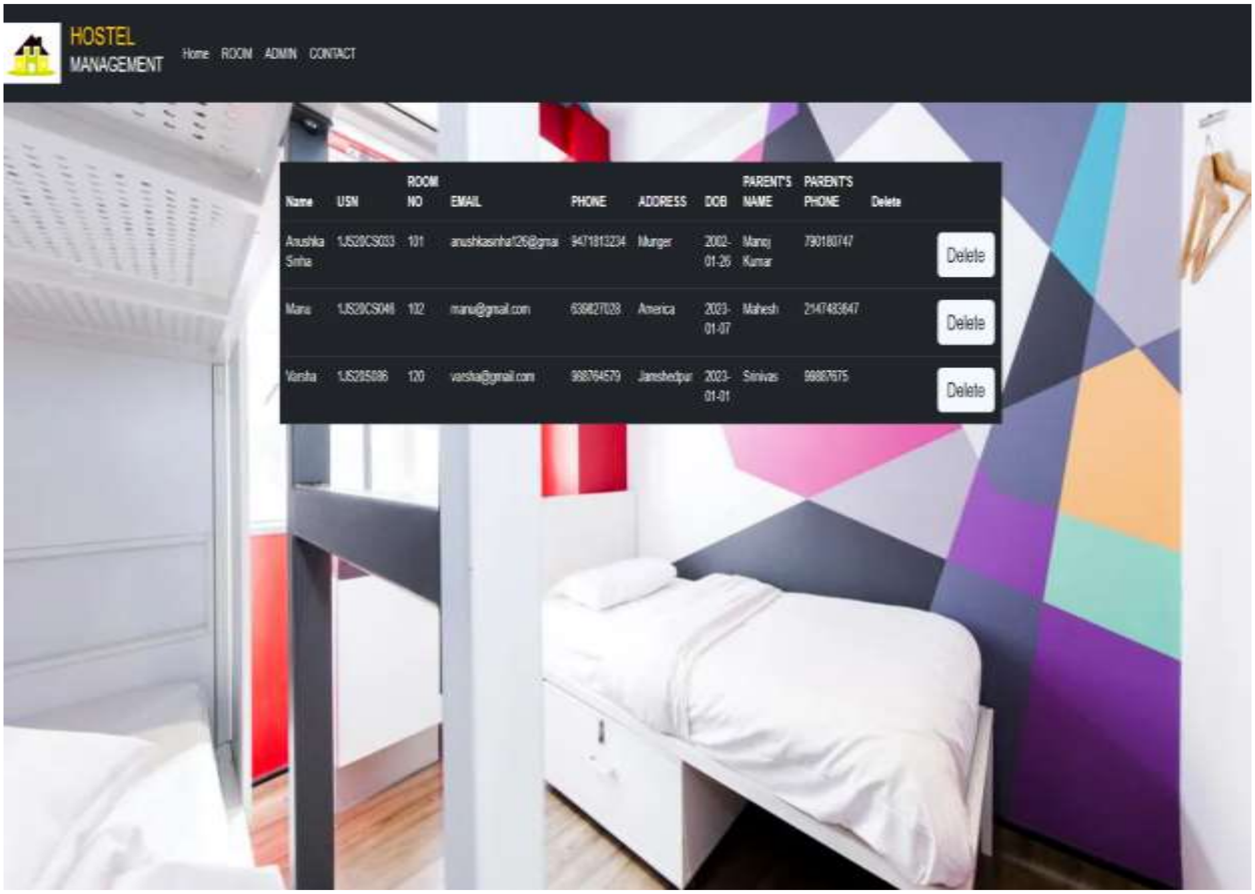
Date of Birth *

Parent's Details:

Name

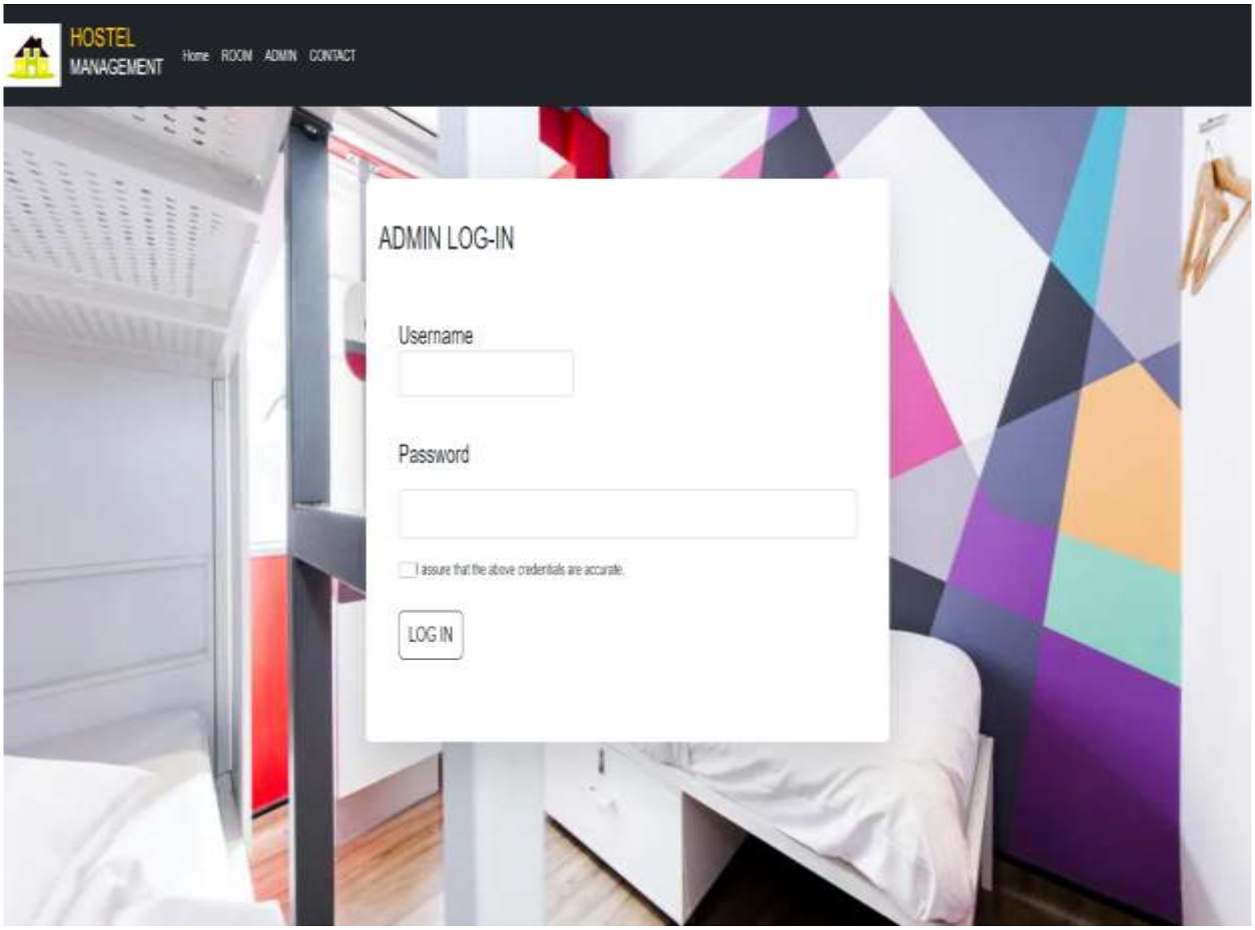
Phone *

Figure 4.9 Student Page



Name	USN	ROOM NO	EMAIL	PHONE	ADDRESS	DOB	PARENT'S NAME	PARENT'S PHONE	Delete
Anushka Sinha	1JS20CS033	101	anushkasinha126@gmai	9471813234	Munger	2002-01-26	Manoj Kumar	790180747	Delete
Manu	1JS20CS046	102	manu@gmail.com	639827028	America	2023-01-07	Mahesh	2147483647	Delete
Varsha	1JS20IS086	120	varsha@gmail.com	988764579	Jamshedpur	2023-01-01	Srinivas	99887675	Delete

Figure 4.10 Student details Page



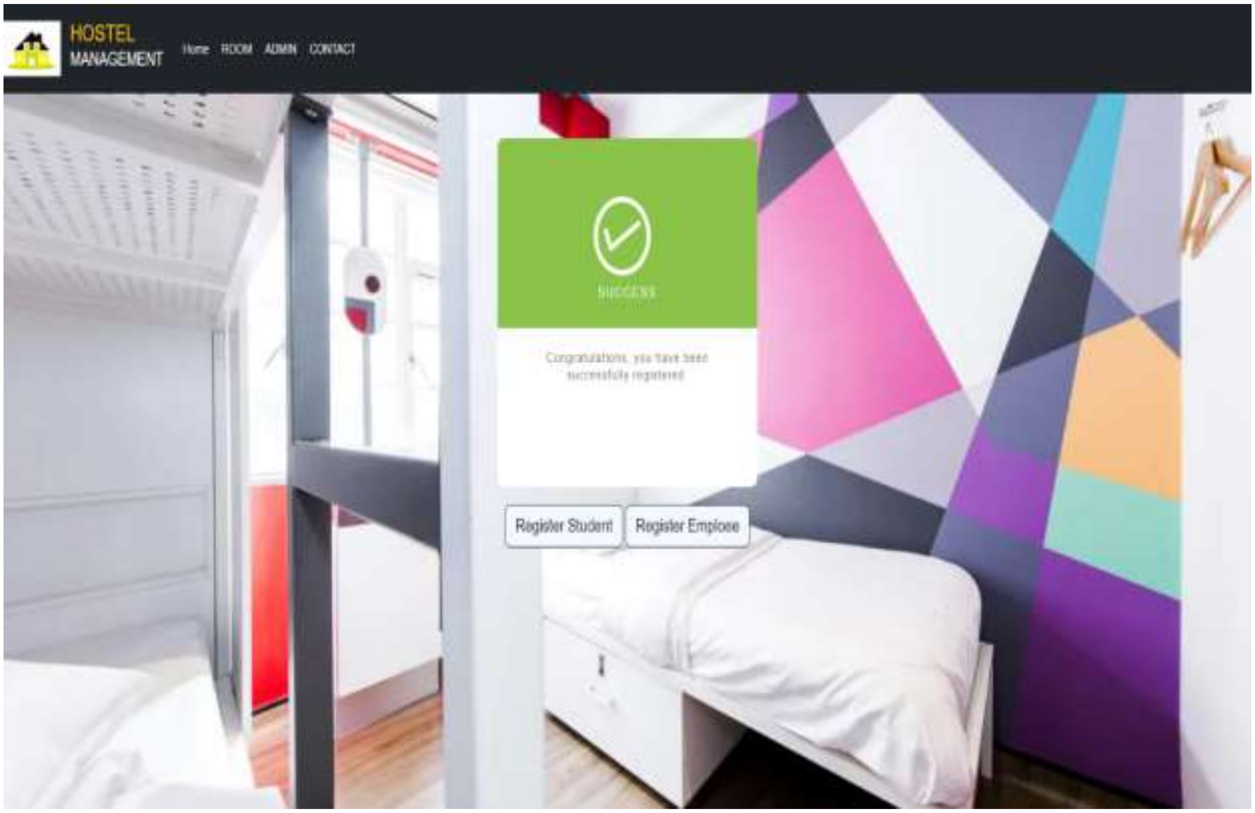
ADMIN LOG-IN

Username

Password

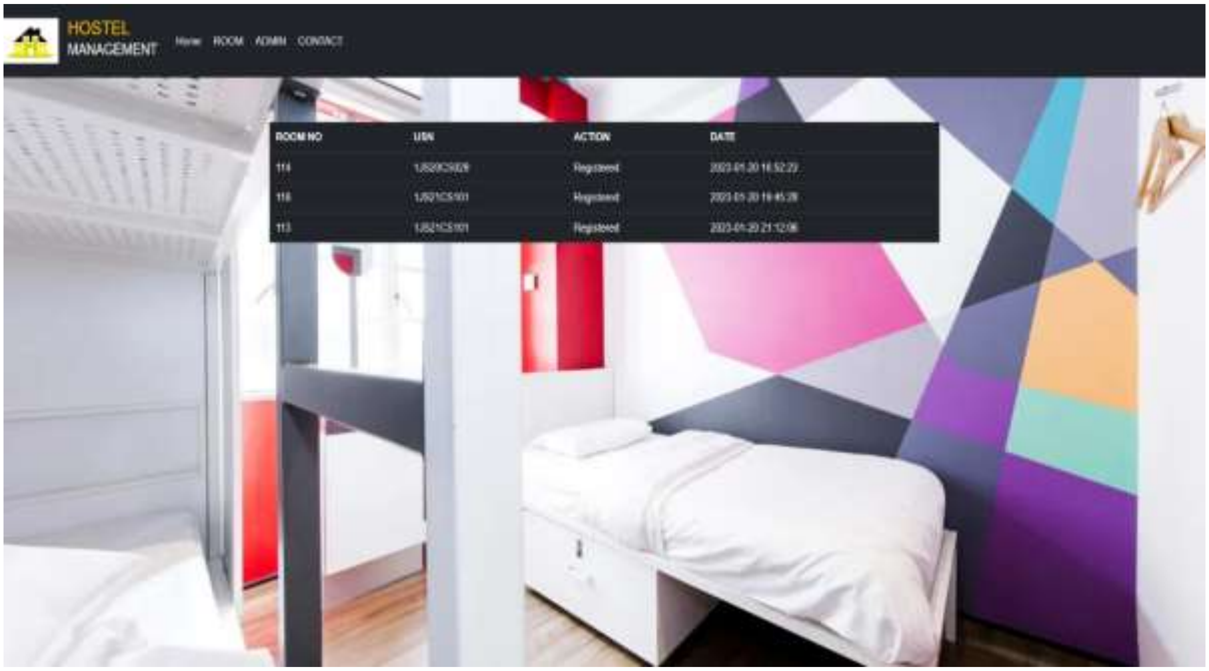
□ I assure that the above credentials are accurate.

Figure 4.11 Admin login Page



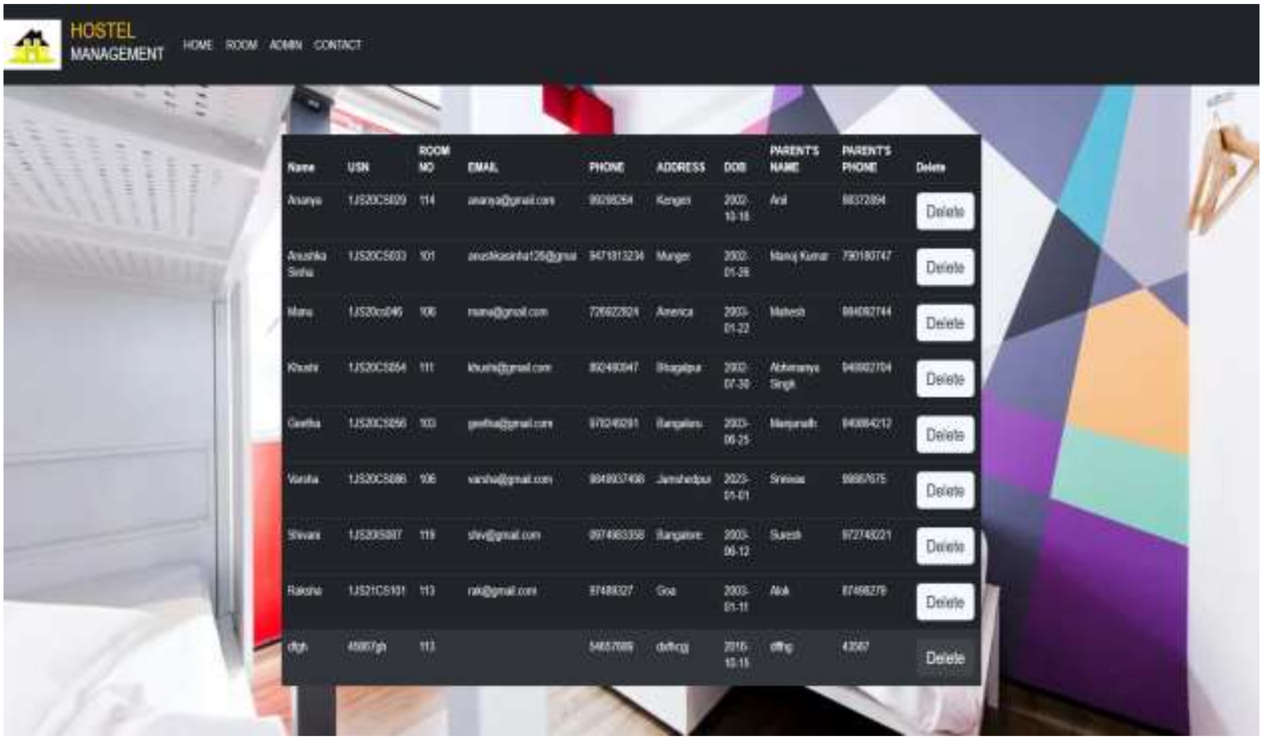
Congratulations, you have been successfully registered

Figure 4.12 Registration successful Page



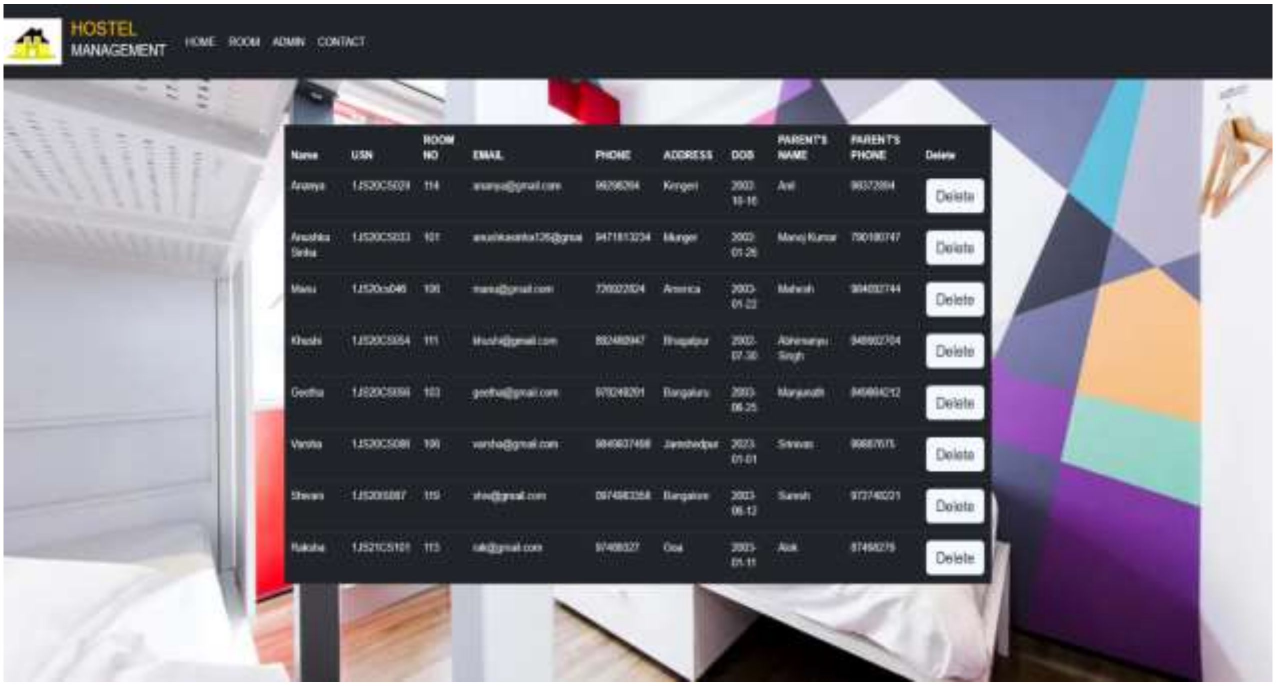
ROOM NO	USN	ACTION	DATE
114	1JS20CS029	Registered	2023-01-20 16:52:22
116	1JS21CS101	Registered	2023-01-20 19:45:28
113	1JS21CS101	Registered	2023-01-20 21:12:06

Figure 4.13 Trigger Page



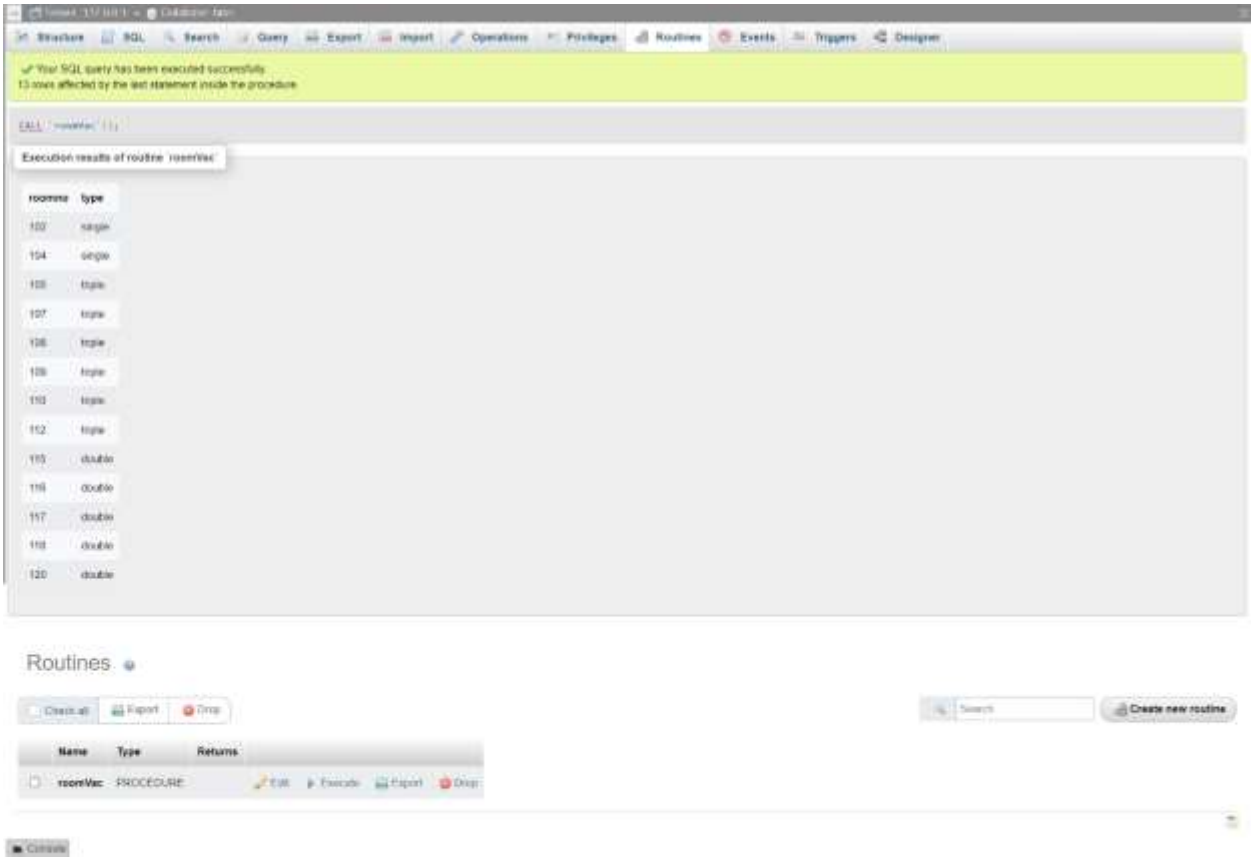
Name	USN	ROOM NO	EMAIL	PHONE	ADDRESS	DOB	PARENT'S NAME	PARENT'S PHONE	Delete
Ananya	1JS20CS029	114	ananya@gmail.com	99298264	Kengeri	2002-10-18	Anil	98372894	Delete
Anushka Sinha	1JS20CS033	101	anushkasinha126@gmai	9471813234	Munger	2002-01-28	Manoj Kumar	790180747	Delete
Manu	1JS20cs046	106	manu@gmail.com	726922824	America	2003-01-22	Mahesh	984092744	Delete
Khushi	1JS20CS054	111	khushi@gmail.com	892480947	Bhagalpur	2002-07-30	Abhimanyu Singh	948902704	Delete
Geetha	1JS20CS056	103	geetha@gmail.com	970249291	Bangaluru	2003-06-25	Manjunath	940884212	Delete
Varsha	1JS20CS086	106	varsha@gmail.com	9849037498	Jamshedpur	2023-01-01	Srinivas	99867675	Delete
Shivani	1JS20IS087	119	shiv@gmail.com	0974983358	Bangalore	2003-06-12	Suresh	972748221	Delete
Raksha	1JS21CS101	113	rak@gmail.com	87489327	Goa	2003-01-11	Alok	87498279	Delete
dfgh	45667gh	113		54657686	dxfhcgj	2016-10-15	dfhg	43567	Delete

Figure 4.14 Before Deletion Page



Name	USN	ROOM NO	EMAIL	PHONE	ADDRESS	DOB	PARENT'S NAME	PARENT'S PHONE	Delete
Ananya	1JS20CS029	114	ananya@gmail.com	99298264	Kengeri	2002-10-18	Anil	98372894	Delete
Anushka Sinha	1JS20CS033	101	anushkasinha126@gmai	9471813234	Munger	2002-01-28	Manoj Kumar	790180747	Delete
Manu	1JS20cs046	106	manu@gmail.com	726922824	America	2003-01-22	Mahesh	984092744	Delete
Khushi	1JS20CS054	111	khushi@gmail.com	892480947	Bhagalpur	2002-07-30	Abhimanyu Singh	948902704	Delete
Geetha	1JS20CS056	103	geetha@gmail.com	970249201	Bangaluru	2003-06-25	Manjunath	940884212	Delete
Varsha	1JS20CS086	106	varsha@gmail.com	9849037498	Jamshedpur	2023-01-01	Srinivas	99867675	Delete
Shivani	1JS20IS087	119	shiv@gmail.com	0974983358	Bangalore	2003-06-12	Suresh	972748221	Delete
Raksha	1JS21CS101	113	rak@gmail.com	87489327	Goa	2003-01-11	Alok	87498279	Delete

Figure 4.15 After Deletion Page



✓ Your SQL query has been executed successfully.
13 rows affected by the last statement inside the procedure

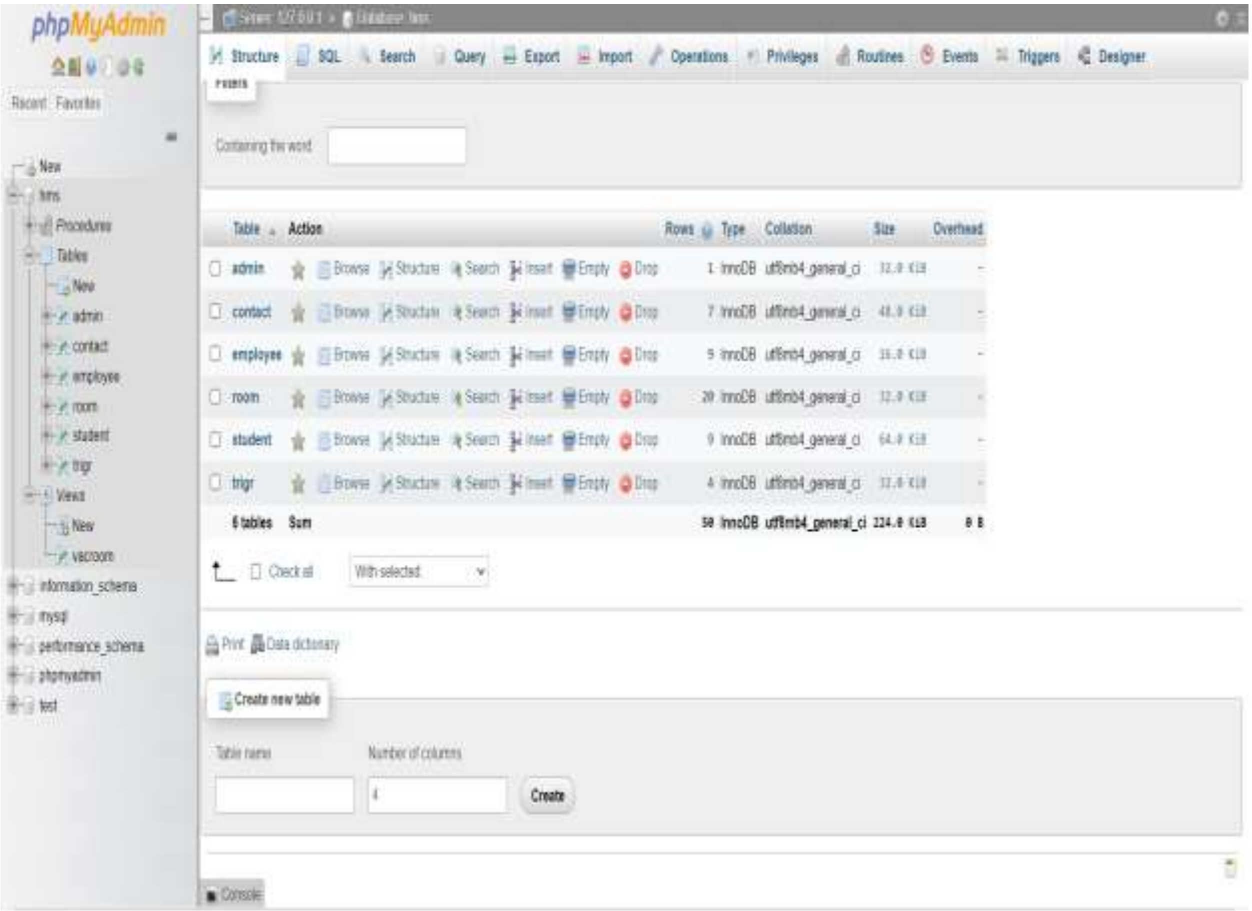
```
CALL `roomVac`();
```

Execution results of routine `roomVac`

roomno	type
102	single
104	single
105	triple
107	triple
108	triple
109	triple
110	triple
112	triple
115	double
116	double
117	double
118	double
120	double

	Name	Type	Returns
□	roomVac	PROCEDURE	

Figure 4.16 View Page



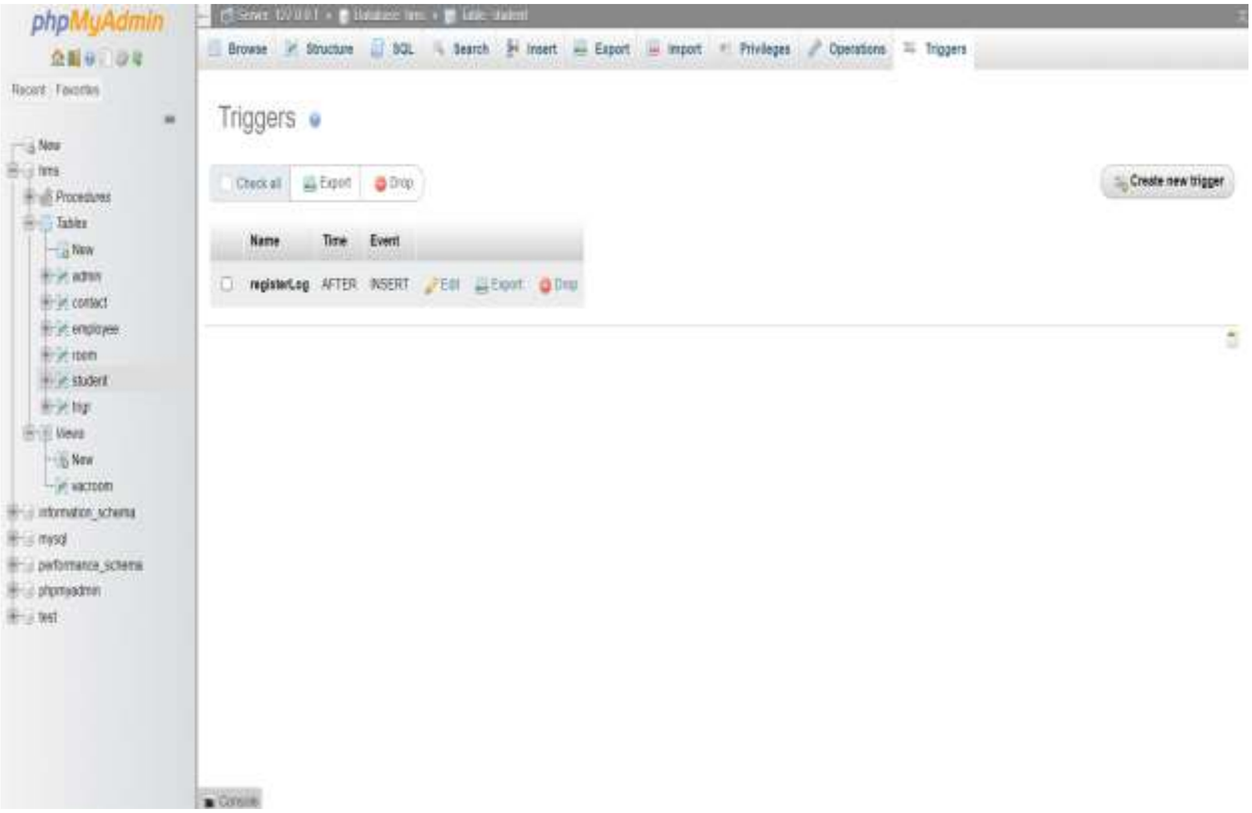
Containing the word:

Table	Action	Rows	Type	Collation	Size	Overhead
admin	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	32.0 KiB	-
contact	Browse Structure Search Insert Empty Drop	7	InnoDB	utf8mb4_general_ci	48.0 KiB	-
employee	Browse Structure Search Insert Empty Drop	9	InnoDB	utf8mb4_general_ci	16.0 KiB	-
room	Browse Structure Search Insert Empty Drop	20	InnoDB	utf8mb4_general_ci	32.0 KiB	-
student	Browse Structure Search Insert Empty Drop	9	InnoDB	utf8mb4_general_ci	64.0 KiB	-
trigr	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	32.0 KiB	-
6 tables	Sum	50	InnoDB	utf8mb4_general_ci	224.0 KiB	0 B

Table name:

Number of columns: 4

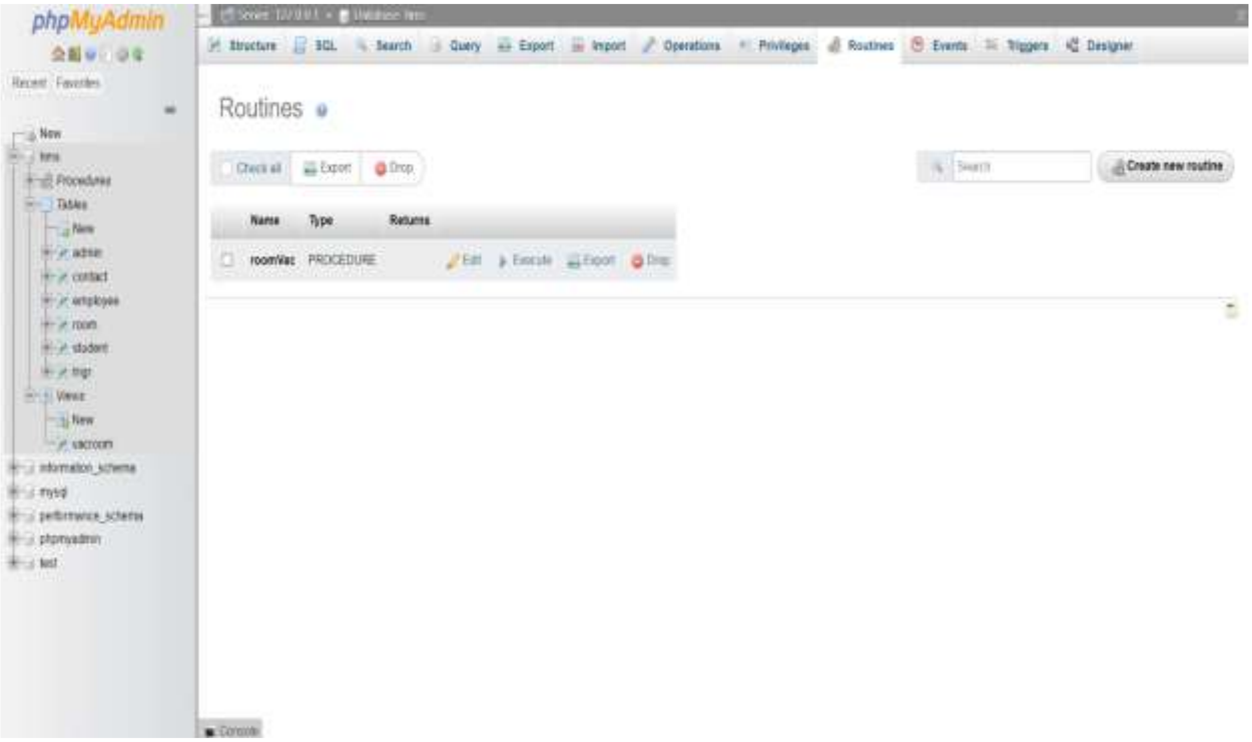
Figure 4.17 Database Page



Triggers

Name	Time	Event
registerLog	AFTER	INSERT

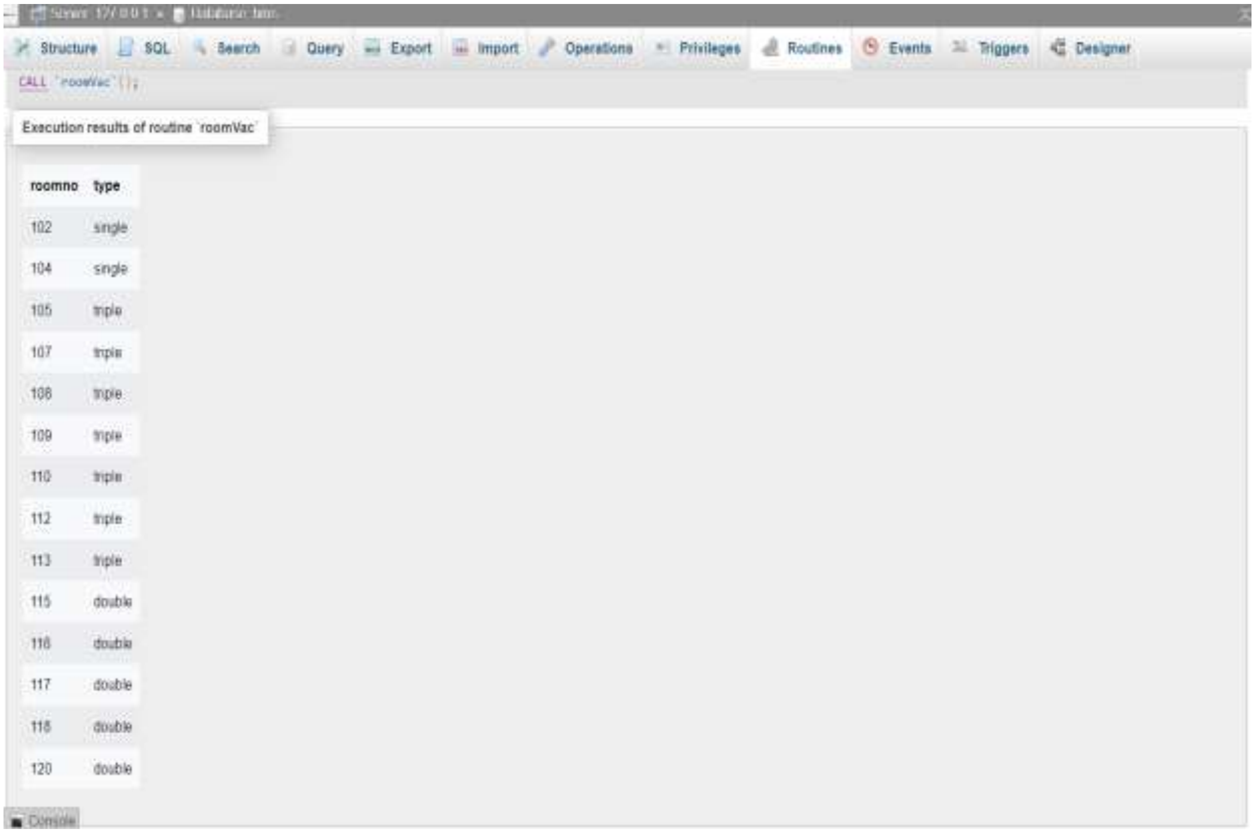
Figure 4.18 Trigger



Routines

	Name	Type	Returns				
□	roomVac	PROCEDURE		Edit	Execute	Export	Drop

Figure 4.19 Stored Procedure



```
CALL `roomVac`();
```

Execution results of routine `roomVac`

roomno	type
102	single
104	single
105	triple
107	triple
108	triple
109	triple
110	triple
112	triple
113	triple
115	double
116	double
117	double
118	double
120	double

Figure 4.20 Result of Stored Procedure

Conclusion and Future Enhancements

Conclusion

The Hostel Management System is a sort of interaction sequence diagram that shows how a group of items interacts and in what order. Software engineers and business experts use these diagrams to understand the requirements of the system or to describe an existing process.

The hostel management system is webbased software to provide college students accommodation to the university hostel more efficiently. This project also keeps details of the hostellers and applied students. It is headed by Warden.

HMS is a webbased system which automates all the activities in the hostel which is currently manual. It enhances the overall performance of the system. Removes the necessity of maintaining all the registers. Removes all kinds of manual errors

This project made by the handwork of me and is a small effort in the development of larger program, and it involves limited aspect which are needed in daytoday hostel operation.

The software development although is a very difficult task but it can be carried out successfully with the effort of my work After the completion of this project we learnt different things about software and its development. I also learnt to work in a group and realize its importance.

Future Enhancements

Hostel management system (HMS).This hostel management software is designed for people who want to manage various activities in the hostel. This project is designed to fulfill the need of the future generation. This project is small package which includes different categories as well as having all possible features. we have expected that it will be helpful to the, customers as well as Administrative member. we will improve this project in future with online room reservation system and also improve in security by providing a magnetic coated card to permanent students for automatic check in and check out. we are waiting for your best suggestion and encouragement which could make us improve the future programming much better than the one we have carried out.