# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

**"Jnana Sangama",** Belgaum-590018.

DBMS Mini Project On

## "Hotel In-House Booking Management System"

**Submitted in partial fulfillment for the requirements of the V Semester degree of**

**BACHELOR OF ENGINEERING**
**IN**
**INFORMATION SCIENCE AND ENGINEERING**

**For The Academic Year**
**2021-2022 By**


**KAVYASHREE S**
**(1DB19IS036)**


**Under the Guidance of**


**Prof.Basavaraj Neelagund**
**Prof.Rohini B.R.**
**Asst.Professor,**
**Dept.Of ISE**



**Department of Information Science and Engineering**

**DON BOSCO INSTITUTE OF TECHNOLOGY**

**Kumbalagodu, Mysore Road, Bengaluru - 560 074.**

# DON BOSCO INSTITUTE OF TECHNOLOGY

**Kumbalagodu, Bengaluru – 560 074.**

**DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING**

# CERTIFICATE

This is to certify that the mini project report entitled **"HOTEL IN-HOUSE BOOKING MANAGEMENT SYSTEM"** is a bonafide work carried out by **KAVYASHREE S (1DB19IS036)** in partial fulfillment of award of Degree of **Bachelor of Engineering in Information Science and Engineering** of Visvesvaraya Technological University, Belagavi, during the academic year 2021-2022. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated. The mini project has been approved as it satisfies the academic requirements associated with the degree mentioned.

**Signature of guide** **Signature of HOD**

**1. ……**

**2. ……** **…….**

**Mr.Basavaraj N & Mrs.Rohini B.R.**
Asst. Professor
Dept. of ISE,
DBIT, Bengaluru.

**Prof. Gowramma G.S.**
Head of Dept.,
Dept. of ISE,
DBIT, Bengaluru

**External Viva**

| **Name of the Examiners** | **Signature with Date** |
|---|---|
| **1. ……..** | **……** |
| **2. ……..** | **……** |

# ACKNOWLEDGEMENT

Here by I am submitting the DBMS mini project report on **"HOTEL IN-HOUSE BOOKING MANAGEMENT SYSTEM"**, as per the scheme of Visvesvaraya Technological University, Belgaum.

In this connection, I would like to express my deep sense of gratitude to my beloved institution Don Bosco Institute of Engineering and also I like to express my sincere gratitude and indebtedness to **Dr.Hemadri Naidu T, Principal, DBIT, Bangalore.**

I would like to express my sincere gratitude to **Prof.Gowramma G.S.** Professor and Head of Dept. of Information Science and Engineering, for providing a congenial environment to work in and carryout my mini project.

I would like to express the deepest sense of gratitude to thank my Project Guide **"Prof.Basavaraj N & Prof.Rohini B.R"**, Asst. Professor, Department of Information Science and Engineering, DBIT, Bangalore for their constant help and support extended towards me during the course of the project.

Finally, I am very much thankful to all the teaching and non-teaching members of the Department of Information Science and Engineering, my seniors, friends and my parents for their constant encouragement, support and help throughout completion of report.

**KAVYASHREE S**

**(1DB19IS036)**

# ABSTRACT

The purpose of Hotel In-House Booking Management system is to automate the existing manual system by the help of computerized equipments and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. The required software and hardware are easily available and easy to work with.

Hotel In-House Booking system, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus, it will help organization in better utilization of resources. The organization can maintain computerized records without redundant entries, that means that one need not be distracted by information that is not relevant, while being able to reach the information.

# TABLE OF CONTENTS

# CHAPTER 1

# INTRODUCTION

Any business that does not have a software application is missing out on one of the most powerful time saving tools available to them. The main reason that it is important for business to have a software application is maintain all the details. These days most business organizations use a desktop or web based or a mobile app for management of records. For this reason of dependency on software applications. We develop a desktop application for hotels in which administrator and customer get a convenient environment for various operations. Usually, the client uses MS Excel or paper, and maintains records, however it is not possible them to share the data from multiple system in multi user environment, there is lot of duplicate work, and chance of mistake. When the records are changed they need to update each and every excel file.

The smart Hotel management system eliminates most of the limitations of the existing software. Increasing efficiency and effectiveness, automation, accuracy, user-friendly interface, information availability, communication capacity, maintenance, cost reduction makes our system smarter than the existing system. We integrate some new and prominent features along with all the necessary features.

## 1.1 PROJECT SUMMARY:

The project, Hotel In-House Booking Management System is a desktop-based application that allows the hotel manager to handle all the hotel booking activities. The hotel manager is very busy person and does not have time to sit and manage the entire activities manually on paper. This application gives him the power and flexibility to manage the entire system from a single system. This project provides room booking for the customers and allows the manager to post available rooms and also the manager can go through the check-in and check-out status and payment status of a customer.

## 1.2 PURPOSE:

- The main purpose of the project is to design and develop a user-friendly efficient computerized Hotel In-House Booking management system.
- An accurate system without any data redundancy.

- Storing the details of the customers.
- Provides easy way of booking rooms.
- Maintaining and updating customer account: generating bill, keeping track of transactions.
- Secured data storage for authority end.
- Provides secure, reliable and fast management system. Thus, it will help organization in better utilization of resources.

## 1.3 OBJECTIVE:

The main objective of this project is to manage the details of Hotel, Rooms, Services, Payments, Bookings. It manages all the information about Hotel, Customers, Bookings. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the Hotel, Rooms, Customers. It tracks all the details about the Customers, Payments, and Bookings.

## 1.4 TOOLS AND TECHNOLOGIES USED:

**Visual Studio Code:** Visual Studio Code is a code editor with support for development operations like debugging, task running, and version control.

Visual Studio Code

**XAMPP:** It helps a local host or server to test its website and clients via computers and laptops before releasing it to the main server. It is a platform that furnishes a suitable environment to test and verify the working of projects based on apache, Perl, MySQL databse, and PHP through the system of the host itself.

# CHAPTER 2

## SYSTEM REQUIREMENT STUDY

The requirements can be broken down into 2 major categories namely hardware and software requirements. The former specifies the minimal hardware facilities expected in a system in which the project has to be run. The latter specifies the essential software needed to build and to run the project.

### 2.1 HARDWARE REQUIREMENTS:

The Hardware requirements are very minimal and the program can be run on most of the machines.

- Processor - AMD Ryzen 5 or better
- Processor Speed - 2.10 GHz or above
- Hard Disk - 8GB or above
- RAM - 4MB or above
- Storage Space - Approx. 2.17MB

### 2.2 SOFTWARE REQUIREMENTS:

- Operating System - WINDOWS 10
- Front end - HTML, CSS, PHP, JavaScript, Bootstrap
- Back end - MySql
- Server - Xampp
- Editor - Visual Studio Code

# CHAPTER 3

# SYSTEM ANALYSIS

## 3.1 EXISTING SYSTEM:

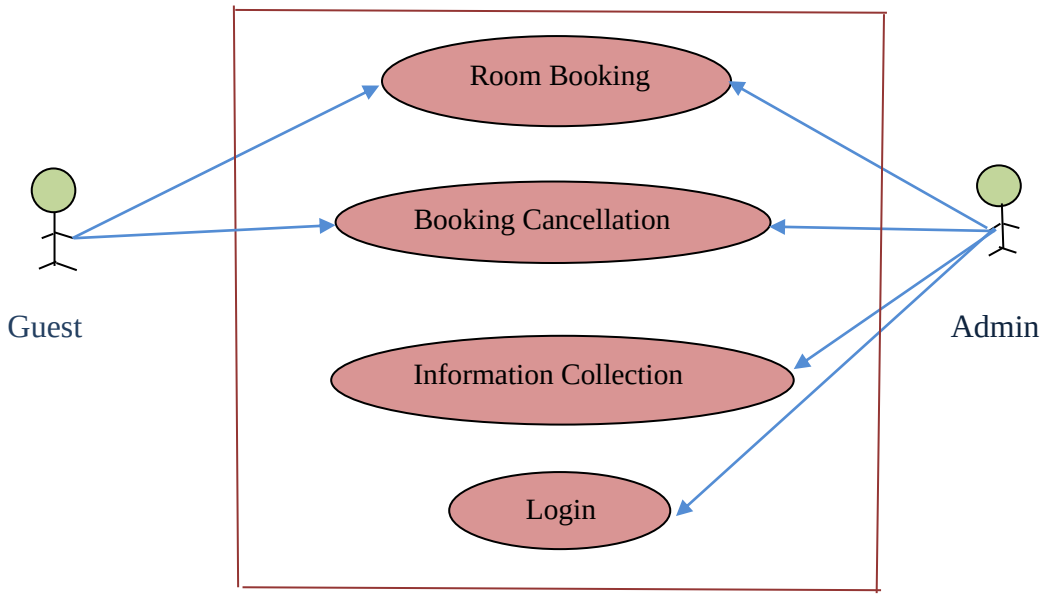
According to current scenario, day by day new technologies comes in picture. Technology has become so vital that now everything can be done at the tip of your fingers. According to the current study many hotel organizations are using software application, Management will become easy as by simple GUI.

Creating a full-fledged desktop application will make it easy, which will display all the details on the software and keep the admin updated about latest room status. An application which will provide utilities like customers details, room detail, bill details etc. The main objective of this project is to provide easiness and automation to the process of managing customer and room information.

## 3.2 PROPOSED SYSTEM:

The proposed system is a web-based application which allows the owner or manager of the hotel to book the room for customer by providing the required details.

## 3.3 USE CASE DIAGRAM:



## 3.4 FLOWCHART:

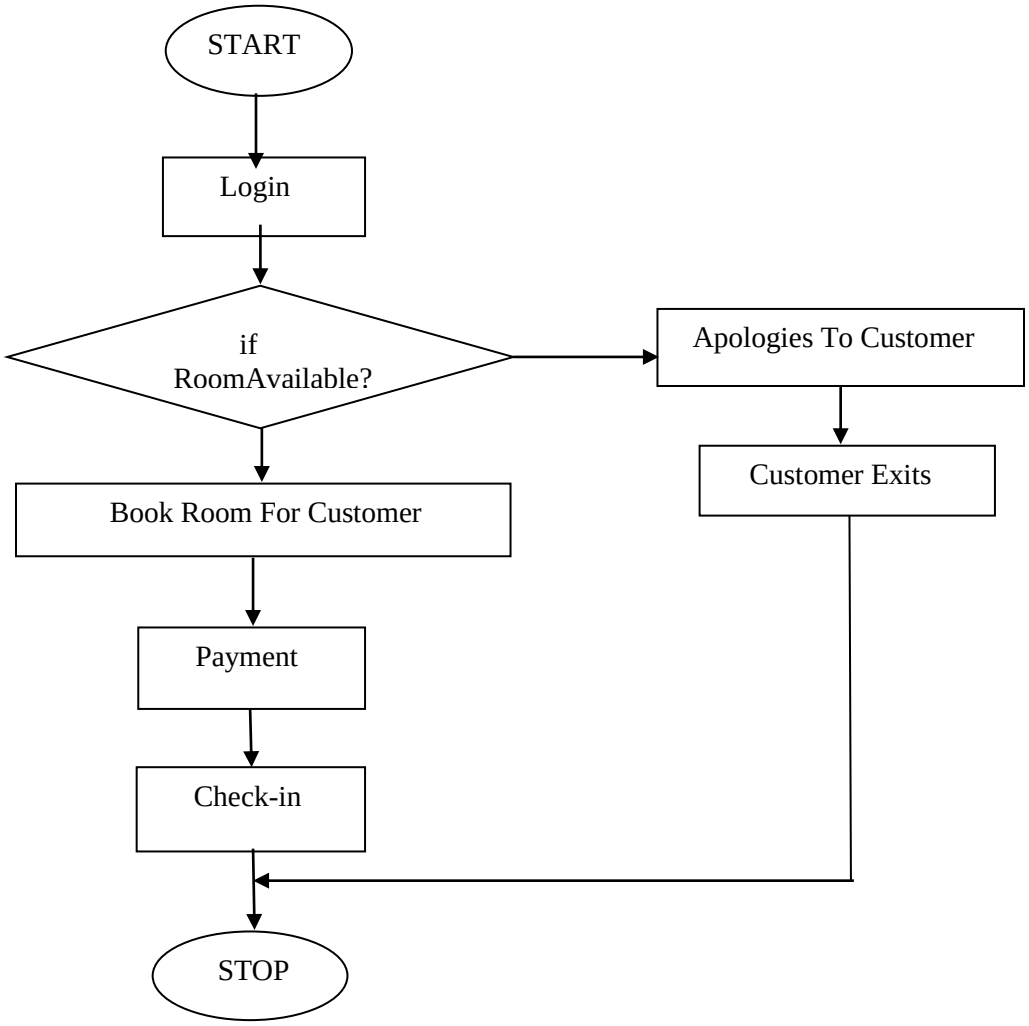



Fig 3.4 Flowchart of HIBMS

# CHAPTER 4

# SYSTEM DESIGN

## 4.1 MAIN PROJECT MODULE:

**Admin:** The owner or the manager of the hotel or anyone who is responsible for running the project i.e entering values in database, updating the database etc. He also maintains the various accounts of Customers. Also, can process a reservation.

## 4.2 SYSTEM OVERVIEW:

The Hotel In-House Booking Management System provides with the following functionalities:

- **Make a reservation:** Make a reservation for a customer by taking all the information about customers and by checking the availability of rooms in the hotel.
- **Manage Rooms:** Admin can add Rooms or he can also edit the room number and room type.
- **Room Status:** Admin can check the status of the room in the hotel i.e the number of rooms which are vacant and which are non-vacant.
- **Check-in:** After making reservation, customer can check-in in to the hotel room.
- **Check-out:** Now the total bill of a customer is generated and shown it to the customer at the time of customer check-out.

## 4.3 TABLE DESCRIPTION:

The Description of all the tables which are present in this project are:

BOOKING:

BOOKING table has the attributes BOOKING_ID, CUSTOMER_ID, ROOM_ID BOOKING_DATE, CHECK_IN, CHECK_OUT, TOTAL_PRICE, REMAINING_PRICE, PAYMENT STATUS and BOOKING_ID is used as the primary key as shown in the table 4.3.1.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|
| 1 | booking_id | int(10) | | | No | None | | AUTO_INCREMENT | Change Drop More |
| 2 | customer_id | int(10) | | | No | None | | | Change Drop More |
| 3 | room_id | int(10) | | | No | None | | | Change Drop More |
| 4 | booking_date | timestamp | | | No | current_timestamp() | | | Change Drop More |
| 5 | check_in | varchar(100) | latin1_swedish_ci | | Yes | NULL | | | Change Drop More |
| 6 | check_out | varchar(100) | latin1_swedish_ci | | No | None | | | Change Drop More |
| 7 | total_price | int(10) | | | No | None | | | Change Drop More |
| 8 | remaining_price | int(10) | | | No | None | | | Change Drop More |
| 9 | payment_status | tinyint(1) | | | No | None | | | Change Drop More |

table 4.3.1 BOOKING

CUSTOMER:

CUSTOMER table has attributes like is CUSTOMER_ID, CUSTOMER_NAME, CONTACT_NO, EMAIL, ID_CARD_TYPE_ID, ID_CARD_NO, ADDRESS used as primary key as shown in the table 4.3.2.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|
| 1 | customer_id | int(10) | | | No | None | | AUTO_INCREMENT | Change Drop More |
| 2 | customer_name | varchar(100) | latin1_swedish_ci | | No | None | | | Change Drop More |
| 3 | contact_no | bigint(20) | | | No | None | | | Change Drop More |
| 4 | email | varchar(100) | latin1_swedish_ci | | No | None | | | Change Drop More |
| 5 | id_card_type_id | int(10) | | | No | None | | | Change Drop More |
| 6 | id_card_no | varchar(20) | latin1_swedish_ci | | No | None | | | Change Drop More |
| 7 | address | varchar(100) | latin1_swedish_ci | | No | None | | | Change Drop More |

table 4.3.2 CUSTOMER

ID_CARD_TYPE:

ID_CARD_TYPE table has the attributes ID_CARD_TYPE_ID, ID_CARD_TYPE, and ID_CARD_TYPE_ID is the primary keys as shown in the table 4.3.3.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|
| 1 | id_card_type_id | int(10) | | | No | None | | AUTO_INCREMENT | Change Drop More |
| 2 | id_card_type | varchar(100) | latin1_swedish_ci | | No | None | | | Change Drop More |

table 4.3.3 ID_CARD_TYPE

ROOM:

ROOM table has the attributes ROOM_ID, ROOM_TYPE_ID, STATUS, CHECK_IN_STATUS, CHECK_OUT_STATUS, DELETESTATUS, and ROOM_ID is the primary key as shown in the table 4.4.4.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|
| 1 | room_id | int(10) | | | No | None | | AUTO_INCREMENT | Change Drop More |
| 2 | room_type_id | int(10) | | | No | None | | | Change Drop More |
| 3 | room_no | varchar(10) | latin1_swedish_ci | | No | None | | | Change Drop More |
| 4 | status | tinyint(1) | | | Yes | NULL | | | Change Drop More |
| 5 | check_in_status | tinyint(1) | | | No | None | | | Change Drop More |
| 6 | check_out_status | tinyint(1) | | | No | None | | | Change Drop More |
| 7 | deleteStatus | tinyint(1) | | | Yes | 0 | | | Change Drop More |

table 4.4.4 ROOM

ROOM_TYPE:

ROOM_TYPE table has the attributes ROOM_TYPE_ID, ROOM_TYPE, PRICE, MAX_PERSON, and ROOM_TYPE_ID is the primary key as shown in the table 4.4.5.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|
| 1 | room_type_id | int(10) | | | No | None | | AUTO_INCREMENT | Change Drop More |
| 2 | room_type | varchar(100) | latin1_swedish_ci | | No | None | | | Change Drop More |
| 3 | price | int(10) | | | No | None | | | Change Drop More |
| 4 | max_person | int(10) | | | No | None | | | Change Drop More |

table 4.4.5 ROOM_TYPE

USER:

USER table has the attributes ID, NAME, and USERNAME, EMAIL, PASSWORD, CREATED_AT and ID is the primary key as shown in the table 4.4.6.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|
| 1 | id | int(11) | | | No | None | | AUTO_INCREMENT | Change Drop More |
| 2 | name | varchar(100) | latin1_swedish_ci | | No | None | | | Change Drop More |
| 3 | username | varchar(15) | latin1_swedish_ci | | No | None | | | Change Drop More |
| 4 | email | varchar(100) | latin1_swedish_ci | | No | None | | | Change Drop More |
| 5 | password | varchar(100) | latin1_swedish_ci | | No | None | | | Change Drop More |
| 6 | created_at | timestamp | | | No | current_timestamp() | | | Change Drop More |

table 4.4.6 USER

TRIG:

TRIG table has the attributes TID, ROOM_NO, ACTION and TIMESTAMP and TID is the primary key as shown in the table 4.4.6.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|
| 1 | tid | int(10) | | | No | None | | AUTO_INCREMENT | Change Drop More |
| 2 | room_no | varchar(10) | latin1_swedish_ci | | No | None | | | Change Drop More |
| 3 | action | varchar(20) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 4 | timestamp | datetime | | | No | None | | | Change Drop More |

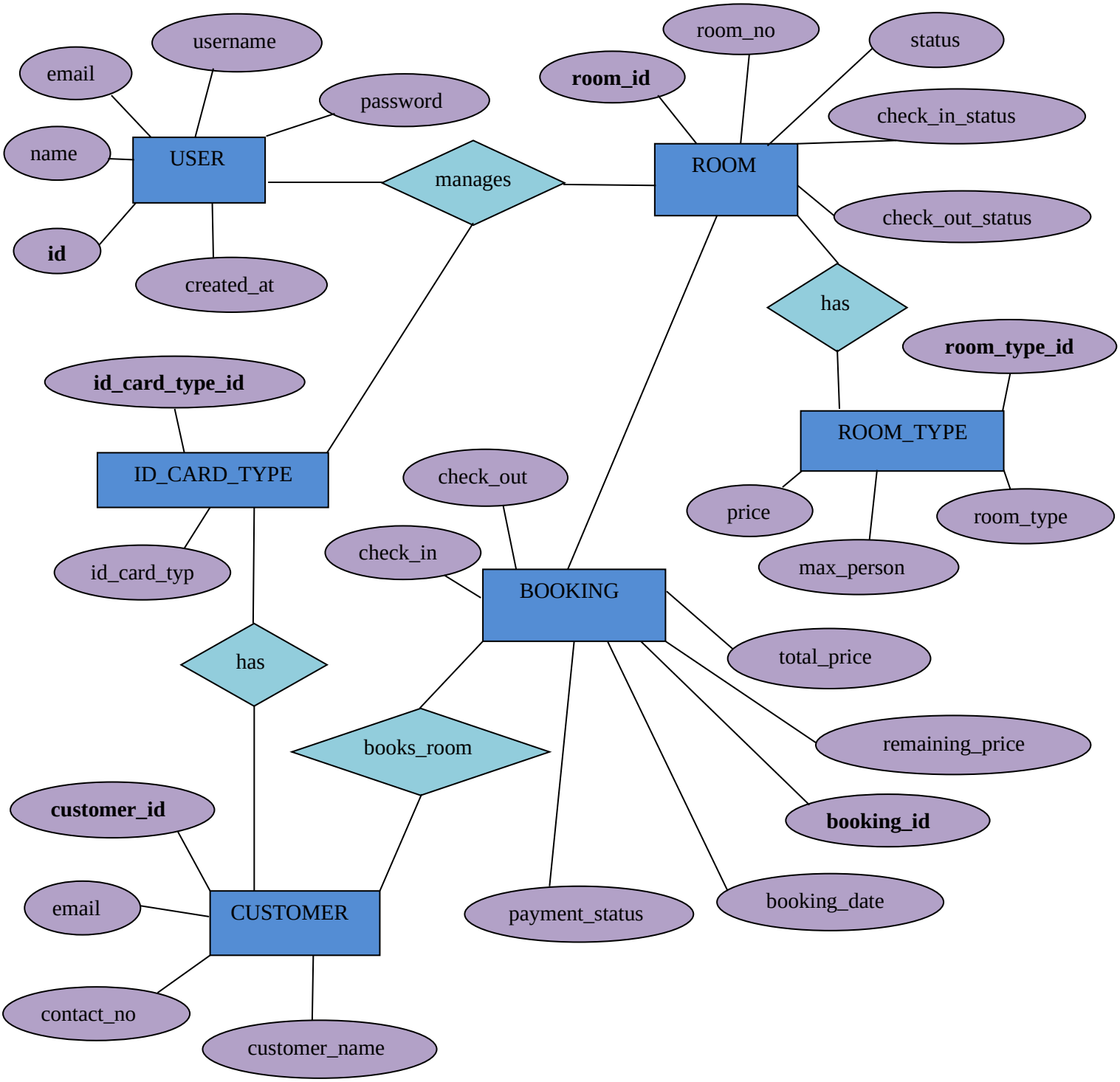
table 4.4.7 TRIG

TRIGGER:

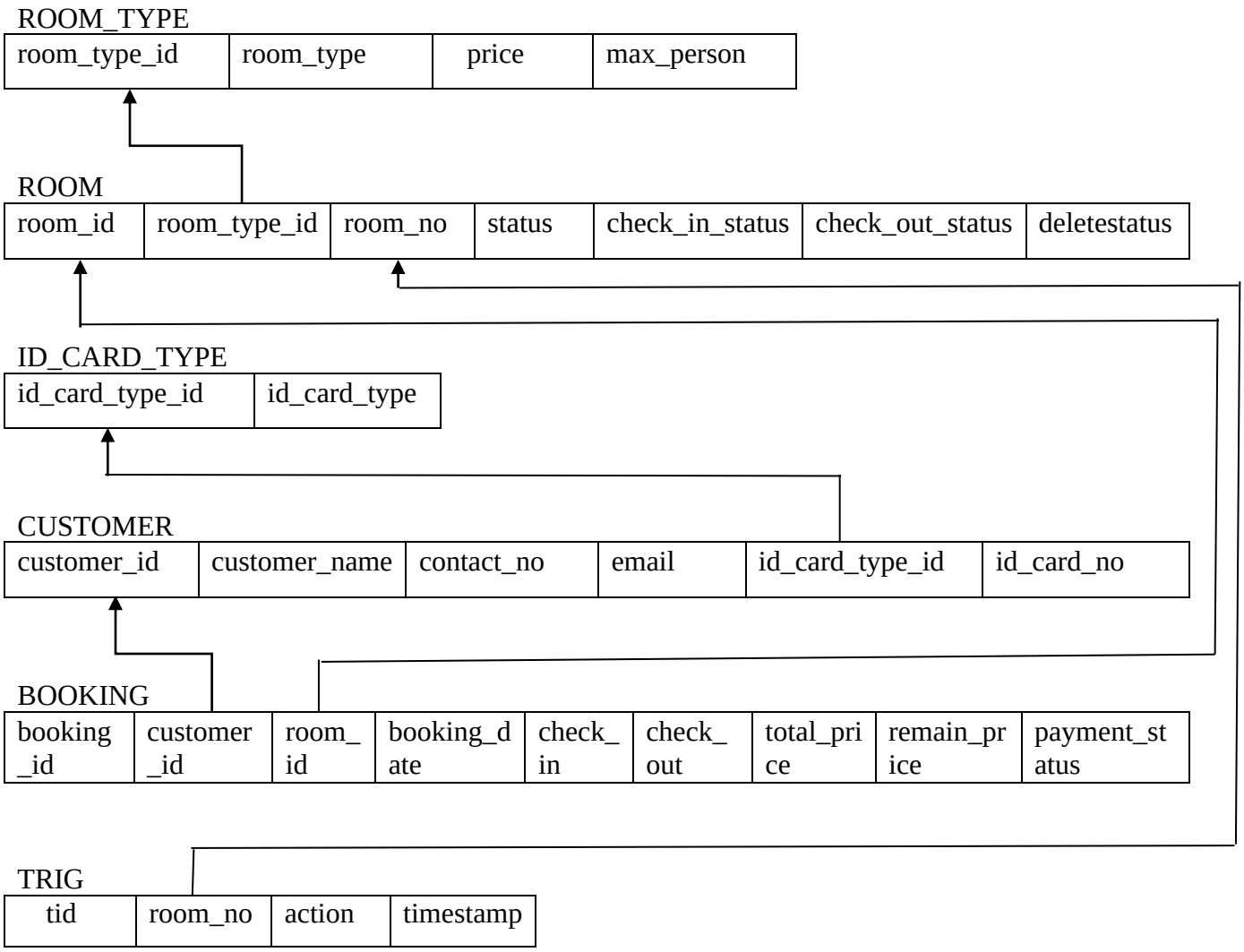
A trigger is a special type of stored procedure that automatically executes when an event occurs in the database server. In this project, one trigger called room as shown in table 4.4.8.

| Name | Action | Time | Event |
|---|---|---|---|
| DELETE | Edit Export Drop | BEFORE | DELETE |
| INSERT | Edit Export Drop | AFTER | INSERT |
| UPDATE | Edit Export Drop | AFTER | UPDATE |

## 4.3 ENTITY RELATIONSHIP DIAGRAM:



## 4.4 REFERENTIAL INTEGRITY DIAGRAM:



# CHAPTER 5

# IMPLEMENTATION

## 5.1 IMPLEMENTATION OF CODE:

Login.php

```
<?php
if (isset($_GET['empty'])){
echo '<div class="alert alert-danger">Enter Username or Password</div>';
}elseif (isset($_GET['loginE'])){
echo '<div class="alert alert-danger">Username or Password Don\'t Match</div>';
} ?>
```

db.php

```
<?php
$connection = mysqli_connect("localhost","root","","hotelms")
?>
```

```
include_once 'db.php';
session_start();

if (isset($_POST['login'])) {
    $email = $_POST['email'];
    $password = $_POST['password'];

    if (!$email && !$password) {
        header('Location:login.php?empty');
    } else {
        $password = md5($password);
        $query = "SELECT * FROM user WHERE username = '$email' OR email='$email' AND
password='$password'";
        $result = mysqli_query($connection, $query);
        if (mysqli_num_rows($result) == 1) {
            $user = mysqli_fetch_assoc($result);
            $_SESSION['username'] = $user['username'];
            $_SESSION['user_id'] = $user['id'];
            header('Location:index.php?dashboard');
        } else {
            header('Location:login.php?loginE');
        }
    }
}

if (isset($_POST['add_room'])) {
```

```
    $room_type_id = $_POST['room_type_id'];
    $room_no = $_POST['room_no'];

    if ($room_no != '') {
        $sql = "SELECT * FROM room WHERE room_no = '$room_no'";
        if (mysqli_num_rows(mysqli_query($connection, $sql)) >= 1) {
            $response['done'] = false;
            $response['data'] = "Room No Already Exist";
        } else {
            $query = "INSERT INTO room (room_type_id,room_no) VALUES
('$room_type_id','$room_no')";
            $result = mysqli_query($connection, $query);

            if ($result) {
                $response['done'] = true;
                $response['data'] = 'Successfully Added Room';
            } else {
                $response['done'] = false;
                $response['data'] = "DataBase Error";
            }
        }
    } else {
        $response['done'] = false;
        $response['data'] = "Please Enter Room No";
    }
     echo json_encode($response);
}

if (isset($_POST['room'])) {
    $room_id = $_POST['room_id'];

    $sql = "SELECT * FROM room WHERE room_id = '$room_id'";
    $result = mysqli_query($connection, $sql);
    if ($result) {
        $room = mysqli_fetch_assoc($result);
        $response['done'] = true;
        $response['room_no'] = $room['room_no'];
        $response['room_type_id'] = $room['room_type_id'];
    } else {
        $response['done'] = false;
        $response['data'] = "DataBase Error";
    }
    echo json_encode($response);
}

if (isset($_POST['edit_room'])) {
    $room_type_id = $_POST['room_type_id'];
```

```
    $room_no = $_POST['room_no'];
    $room_id = $_POST['room_id'];

    if ($room_no != '') {
        $query = "UPDATE room SET room_no = '$room_no',room_type_id = '$room_type_id' where
room_id = '$room_id'";
        $result = mysqli_query($connection, $query);

        if ($result) {
            $response['done'] = true;
            $response['data'] = 'Successfully Edit Room';
        } else {
            $response['done'] = false;
            $response['data'] = "DataBase Error";
        }
    } else {
        $response['done'] = false;
        $response['data'] = "Please Enter Room No";
    }
     echo json_encode($response);
}

if (isset($_GET['delete_room'])) {
    $room_id = $_GET['delete_room'];
    $sql = "UPDATE room set deleteStatus = '1' WHERE room_id = '$room_id' AND status IS
NULL";
    $result = mysqli_query($connection, $sql);
    if ($result) {
        header("Location:index.php?room_mang&success");
    } else {
        header("Location:index.php?room_mang&error");
    }
}

if (isset($_POST['room_type'])) {
    $room_type_id = $_POST['room_type_id'];

    $sql = "SELECT * FROM room WHERE room_type_id = '$room_type_id' AND status IS NULL
AND deleteStatus = '0'";
    $result = mysqli_query($connection, $sql);
    if ($result) {
        echo "<option selected disabled>Select Room Type</option>";
        while ($room = mysqli_fetch_assoc($result)) {
            echo "<option value='" . $room['room_id'] . "'>" . $room['room_no'] . "</option>";
        }
    } else {
        echo "<option>No Available</option>";
```

```
    }
}

if (isset($_POST['room_price'])) {
    $room_id = $_POST['room_id'];

    $sql = "SELECT * FROM room NATURAL JOIN room_type WHERE room_id = '$room_id'";
    $result = mysqli_query($connection, $sql);
    if ($result) {
        $room = mysqli_fetch_assoc($result);
        echo $room['price'];
    } else {
        echo "0";
    }
}

if (isset($_POST['booking'])) {
    $room_id = $_POST['room_id'];
    $check_in = $_POST['check_in'];
    $check_out = $_POST['check_out'];
    $total_price = $_POST['total_price'];
    $name = $_POST['name'];
    $contact_no = $_POST['contact_no'];
    $email = $_POST['email'];
    $id_card_id = $_POST['id_card_id'];
    $id_card_no = $_POST['id_card_no'];
    $address = $_POST['address'];

    $customer_sql = "INSERT INTO customer
(customer_name,contact_no,email,id_card_type_id,id_card_no,address) VALUES
('$name','$contact_no','$email','$id_card_id','$id_card_no','$address')";
    $customer_result = mysqli_query($connection, $customer_sql);

    if ($customer_result) {
        $customer_id = mysqli_insert_id($connection);
        $booking_sql = "INSERT INTO booking
(customer_id,room_id,check_in,check_out,total_price,remaining_price) VALUES
('$customer_id','$room_id','$check_in','$check_out','$total_price','$total_price')";
        $booking_result = mysqli_query($connection, $booking_sql);
        if ($booking_result) {
            $room_stats_sql = "UPDATE room SET status = '1' WHERE room_id = '$room_id'";
            if (mysqli_query($connection, $room_stats_sql)) {
                $response['done'] = true;
                $response['data'] = 'Successfully Booking';
            } else {
                $response['done'] = false;
                $response['data'] = "DataBase Error in status change";
```

```
            }
        } else {
            $response['done'] = false;
            $response['data'] = "DataBase Error booking";
        }
    } else {
        $response['done'] = false;
        $response['data'] = "DataBase Error add customer";
    }
     echo json_encode($response);
}

if (isset($_POST['cutomerDetails'])) {
    //$customer_result='';
    $room_id = $_POST['room_id'];

    if ($room_id != '') {
        $sql = "SELECT * FROM room NATURAL JOIN room_type NATURAL JOIN booking
NATURAL JOIN customer WHERE room_id = '$room_id' AND payment_status = '0'";
        $result = mysqli_query($connection, $sql);
        if ($result) {
            $customer_details = mysqli_fetch_assoc($result);
            $id_type = $customer_details['id_card_type_id'];
            $query = "select id_card_type from id_card_type where id_card_type_id = '$id_type'";
            $result = mysqli_query($connection, $query);
            $id_type_name = mysqli_fetch_assoc($result);
            $response['done'] = true;
            $response['customer_id'] = $customer_details['customer_id'];
            $response['customer_name'] = $customer_details['customer_name'];
            $response['contact_no'] = $customer_details['contact_no'];
            $response['email'] = $customer_details['email'];
            $response['id_card_no'] = $customer_details['id_card_no'];
            $response['id_card_type_id'] = $id_type_name['id_card_type'];
            $response['address'] = $customer_details['address'];
            $response['remaining_price'] = $customer_details['remaining_price'];
        } else {
            $response['done'] = false;
            $response['data'] = "DataBase Error";
        }
         echo json_encode($response);
    }
}

if (isset($_POST['booked_room'])) {
    $room_id = $_POST['room_id'];

    $sql = "SELECT * FROM room NATURAL JOIN room_type NATURAL JOIN booking
```

```
NATURAL JOIN customer WHERE room_id = '$room_id' AND payment_status = '0'";
    $result = mysqli_query($connection, $sql);
    if ($result) {
        $room = mysqli_fetch_assoc($result);
        $response['done'] = true;
        $response['booking_id'] = $room['booking_id'];
        $response['name'] = $room['customer_name'];
        $response['room_no'] = $room['room_no'];
        $response['room_type'] = $room['room_type'];
        $response['check_in'] = date('M j, Y', strtotime($room['check_in']));
        $response['check_out'] = date('M j, Y', strtotime($room['check_out']));
        $response['total_price'] = $room['total_price'];
        $response['remaining_price'] = $room['remaining_price'];
    } else {
        $response['done'] = false;
        $response['data'] = "DataBase Error";
    }
    echo json_encode($response);
}

if (isset($_POST['check_in_room'])) {
    $booking_id = $_POST['booking_id'];
    $advance_payment = $_POST['advance_payment'];

    if ($booking_id != '') {
        $query = "select * from booking where booking_id = '$booking_id'";
        $result = mysqli_query($connection, $query);
        $booking_details = mysqli_fetch_assoc($result);
        $room_id = $booking_details['room_id'];
        $remaining_price = $booking_details['total_price'] - $advance_payment;

        $updateBooking = "UPDATE booking SET remaining_price = '$remaining_price' where
booking_id = '$booking_id'";
        $result = mysqli_query($connection, $updateBooking);
        if ($result) {
            $updateRoom = "UPDATE room SET check_in_status = '1' WHERE room_id =
'$room_id'";
            $updateResult = mysqli_query($connection, $updateRoom);
            if ($updateResult) {
                $response['done'] = true;
            } else {
                $response['done'] = false;
                $response['data'] = "Problem in Update Room Check in status";
            }
        } else {
            $response['done'] = false;
            $response['data'] = "Problem in payment";
```

```
        }
    } else {
        $response['done'] = false;
        $response['data'] = "Error With Booking";
    }
    echo json_encode($response);
}

if (isset($_POST['check_out_room'])) {
    $booking_id = $_POST['booking_id'];
    $remaining_amount = $_POST['remaining_amount'];
        if ($booking_id != '') {
        $query = "select * from booking where booking_id = '$booking_id'";
        $result = mysqli_query($connection, $query);
        $booking_details = mysqli_fetch_assoc($result);
        $room_id = $booking_details['room_id'];
        $remaining_price = $booking_details['remaining_price'];
            if ($remaining_price == $remaining_amount) {
            $updateBooking = "UPDATE booking SET remaining_price = '0',payment_status = '1' where
booking_id = '$booking_id'";
            $result = mysqli_query($connection, $updateBooking);
            if ($result) {
                $updateRoom = "UPDATE room SET status = NULL,check_in_status =
'0',check_out_status = '1' WHERE room_id = '$room_id'";
                $updateResult = mysqli_query($connection, $updateRoom);
                if ($updateResult) {
                    $response['done'] = true;
                } else {
                    $response['done'] = false;
                    $response['data'] = "Problem in Update Room Check in status";
                }
            } else {
                $response['done'] = false;
                $response['data'] = "Problem in payment";
            }
        } else {
            $response['done'] = false;
            $response['data'] = "Please Enter Full Payment";
        }
    } else {
        $response['done'] = false;
        $response['data'] = "Error With Booking";
    }
    echo json_encode($response);
}
```

# SNAPSHOTS

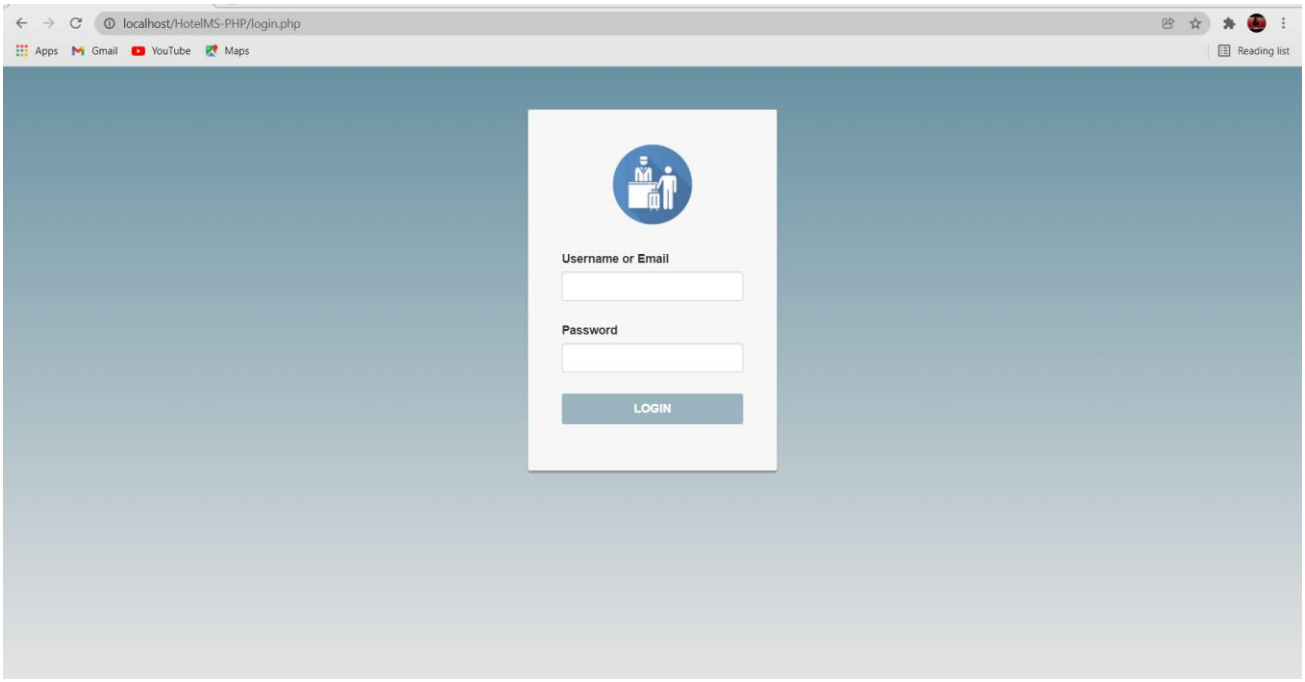



Fig (i): Login page

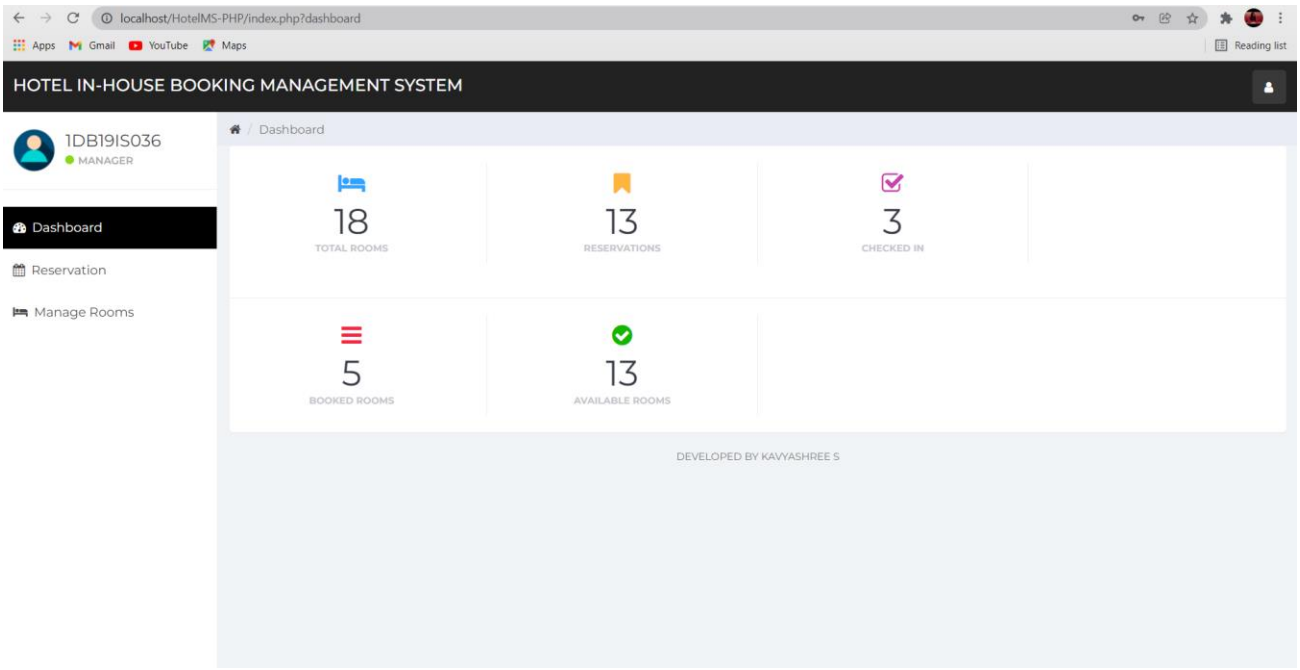



Fig (ii): Dashboard page

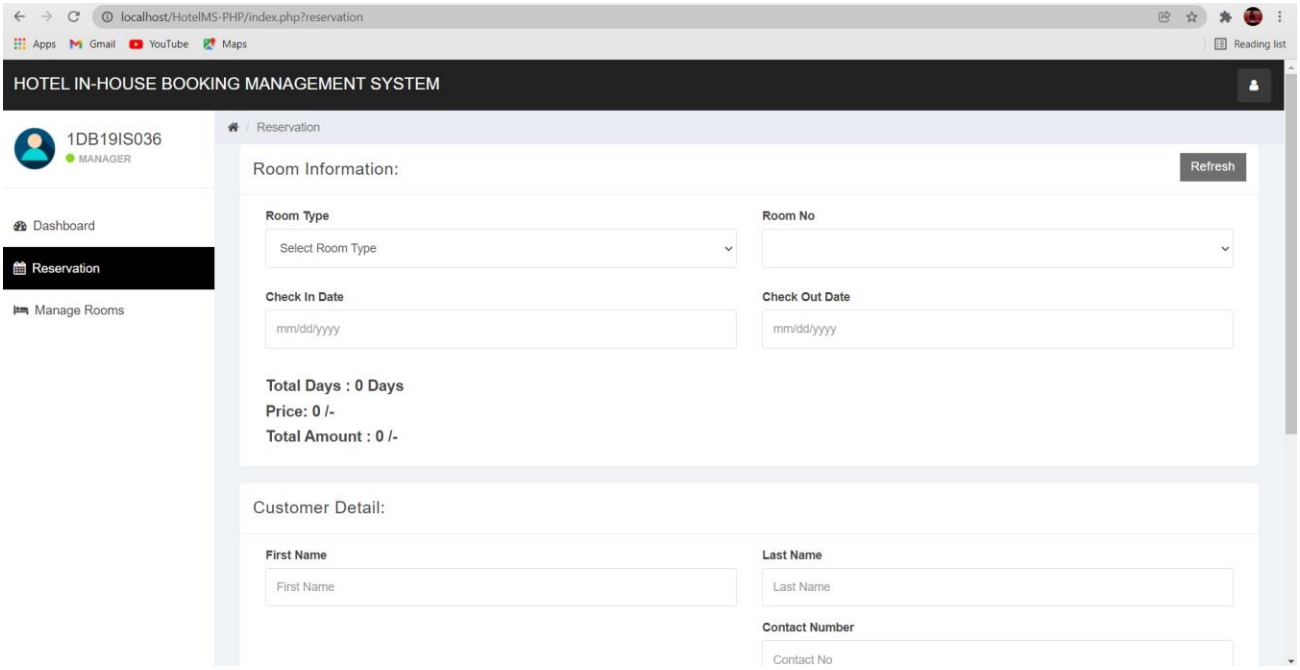



Fig (iii): Reservation page

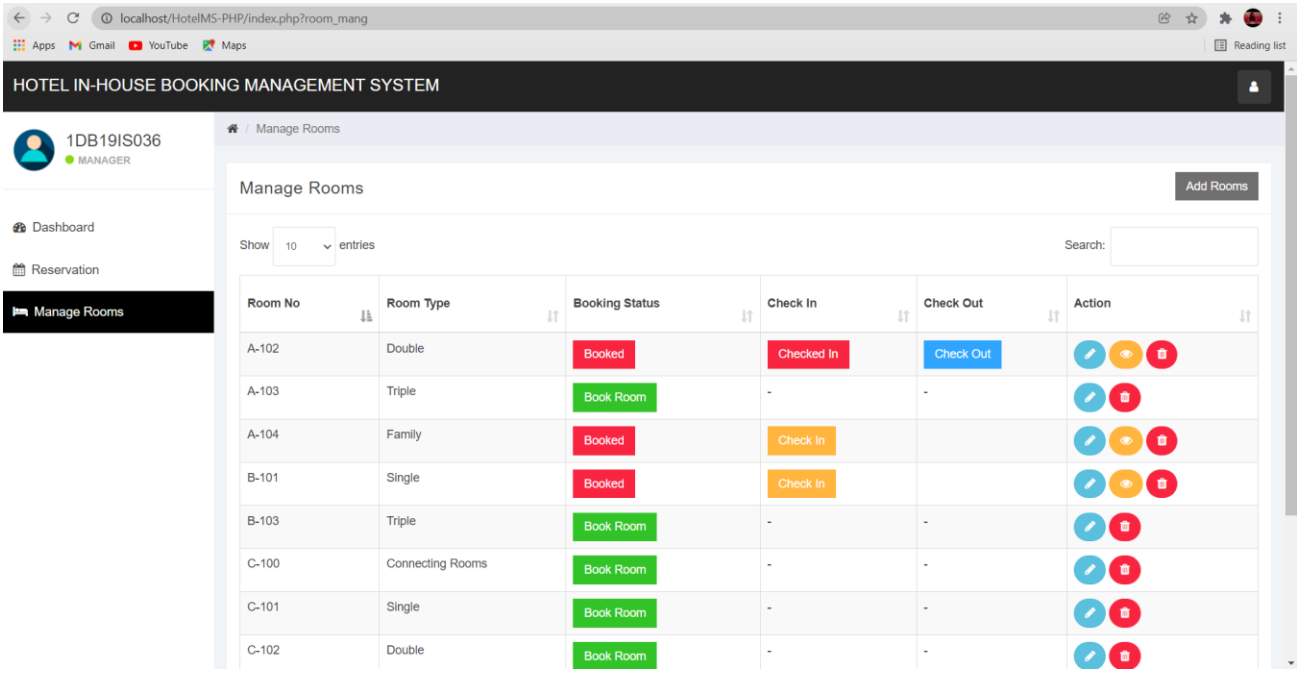



Fig (iv): Manage Rooms

# CONCLUSION

This is my first attempt in developing a desktop application, which gave me a basic understanding of development and challenges of web application development. The main aim of this project is to overcome the time required to search for any information and add mobility and automation to the process of managing customers and room's information. This application has been implemented and tested on real devices.

## REFERENCES:


- HTML, CSS: The Complete Reference, Fifth Edition
- Learning PHP, MYSQL, JavaScript & Bootstrap

**Website:**

- www.researchgate.com
- www.simplilearn.com
- www.codeastro.com
- www.geeksforgeeks.org
- www.iNetTutor.com