

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI – 590 018



A DBMS Mini Project Report on

HOTEL 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 Information Science and Engineering** of Visvesvaraya Technological University, Belagavi

Submitted by

Sameer Singh

- 1RN20IS131

Sharanya RP

- 1RN20IS143

Under the Guidance of

Faculty Incharge

Mr. T S Bhagvath Singh

Associate Professor
Dept. of ISE, RNSIT

Lab Incharge

Dr. R Rajkumar

Associate Professor
Dept. of ISE, RNSIT



An Institute with a Difference

Department of Information Science and Engineering

RNS Institute of Technology

Channasandra, Dr. Vishnuvardhan Road, RR Nagar Post

Bengaluru – 560 098

2022 – 2023

RNS INSTITUTE OF TECHNOLOGY

Channasandra, Dr.Vishnuvardhan Road, RR Nagar Post,
Bengaluru – 560 098

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING



ESTD:2001

An Institute with a Difference

CERTIFICATE

This is to certify that the Mini project report entitled **HOTEL MANAGEMENT SYSTEM** has been successfully completed by **SAMEER SINGH** bearing **USN 1RN20IS131** and **SHARANYA RP** bearing **USN 1RN20IS143**, presently V semester student of **RNS Institute of Technology** in partial fulfillment of the requirements as a part of the DBMS Laboratory for the award of the degree **Bachelor of Engineering in Information Science and Engineering** under **Visvesvaraya Technological University, Belagavi** during academic year 2022 – 2023. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements as a part of DBMS Laboratory for the said degree.

Mr. T S Bhagavath Singh
Faculty Incharge

Dr. R Rajkumar
Lab Incharge

Dr. Suresh L
Professor and HOD

External Viva

Name of the Examiners

Signature with date

1. _____

2. _____

DECLARATION

We, **SAMEER SINGH [USN: 1RN20IS0131]** and **SHARANYA RP [USN: 1RN20IS143]**, students of V Semester BE, in Information Science and Engineering, RNS Institute of Technology hereby declare that the Mini project entitled **HOTEL MANAGEMENT SYSTEM** has been carried out by us and submitted in partial fulfilment of the requirements for the V Semester degree of Bachelor of Engineering in Information Science and Engineering of Visvesvaraya Technological University, Belagavi during the academic year 2022-2023.

Place: Bengaluru

SAMEER SINGH
[1RN20IS131]

SHARANYA RP
[1RN20IS143]

ABSTRACT

The purpose of Hotel Management System is to automate the existing manual system by the help of computerized equipment 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 focus of the project is to a good hotel management system for a particular Hotel. It should be controlled by the admin and the users can very easily book the hotel rooms.

Administrator Panel Account Manager – Administrator – Administrator can add / edit and manage administrator accounts. Hotels Manager – Hotels – Administrator can manage hotels that will appear on the site with the hotel name, description, facilities, phone and fax – Room Types – Administrators can define the type of rooms in the hotels, rooms prices and upload an image for each room. Hotel Rooms – For each Hotel the administrator can define the rooms available, rooms number, max occupants and remarks on the specific room. Booking Statistics – Administrators can view statistics of booking on bar charts that show the difference in bookings according to months.

ACKNOWLEDGEMENT

The fulfillment and rapture that go with the fruitful finishing of any assignment would be inadequate without the specifying the people who made it conceivable, whose steady direction and support delegated the endeavors with success.

We would like to profoundly thank **Management** of **RNS Institute of Technology** for providing such a healthy environment to carry out this Project work.

We would like to express our thanks to our Principal **Dr. M K Venkatesha** for his support and inspired us towards the attainment of knowledge.

We wish to place on record our words of gratitude to **Dr. Suresh L**, Professor and Head of the Department, Information Science and Engineering, for being the enzyme and master mind behind our Project work.

We would like to express our profound and cordial gratitude to our Laboratory Incharge **Dr. R Rajkumar**, Associate Professor, Department of Information Science and Engineering for his valuable guidance, constructive comments and continuous encouragement throughout the Project work.

We would like to express our profound and cordial gratitude to our Faculty Incharge **Mr. T S Bhagavath Singh**, Associate Professor, Department of Information Science and Engineering for his valuable guidance in preparing Project report.

We would like to thank all other teaching and non-teaching staff of Information Science & Engineering who have directly or indirectly helped us to carry out the project work.

And lastly, we would hereby acknowledge and thank our parents who have been a source of inspiration and also instrumental in carrying out this Project work.

SAMEER SINGH (1RN20IS131)

SHARANYA RP (1RN20IS143)

TABLE OF CONTENTS

| | |
|---|-------------|
| CERTIFICATE | II |
| DECLARATION | III |
| ABSTRACT | IV |
| ACKNOWLEDGEMENT | V |
| TABLE OF CONTENTS..... | VI |
| LIST OF FIGURES | VII |
| LIST OF TABLES | VIII |
| CHAPTER 1 | 1 |
| INTRODUCTION..... | 1 |
| CHAPTER 2 | 2 |
| LITRATURE SURVEY | 2 |
| 2.1 Traditional File System | 2 |
| CHAPTER 3 | 7 |
| SYSTEM REQUIREMENTS | 7 |
| 3.1 Hardware Requirements | 7 |
| 3.2 Software Requirements | 7 |
| CHAPTER 4 | 8 |
| SYSTEM DESIGN..... | 8 |
| 4.1 Entity Relation Diagram..... | 8 |
| 4.2 Description of ER Diagram..... | 9 |
| 4.2 Schema Diagram | 10 |
| CHAPTER 5 | 11 |
| IMPLEMENTATION | 11 |
| 5.1 HTML5..... | 11 |
| 5.2 PHP..... | 11 |
| 5.3 SQL (Structured Query Language) | 12 |
| 5.4 Code Snippets..... | 13 |
| 5.5 Stored Procedures..... | 16 |
| 5.6 Triggers | 17 |
| CHAPTER 6 | 19 |
| SNAPSHOTS | 19 |
| CHAPTER 7 | 28 |
| CONCLUSIONS AND FUTURE ENHANCEMENTS..... | 28 |
| REFERENCES..... | 29 |

LIST OF FIGURES

| Figure No. | Description | Page |
|-------------------|--------------------------------------|-------------|
| Figure 2.1 | Components of DBMS | 6 |
| Figure 4.1 | Symbols and Notations for ER Diagram | 8 |
| Figure 4.2 | ER Diagram | 9 |
| Figure 4.3 | Schema Diagram | 10 |
| Figure 5.1 | Stored Procedure | 16 |
| Figure 5.2 | Data in stored procedure | 17 |
| Figure 5.3 | Trigger | 18 |
| Figure 5.4 | Trigger execution on webpage | 18 |
| Figure 6.1 | Home page of user | 19 |
| Figure 6.2 | Hotel rooms | 20 |
| Figure 6.3 | Hotel rooms with prices | 20 |
| Figure 6.4 | Detailed prices | 21 |
| Figure 6.5 | Booked page of admin | 21 |
| Figure 6.6 | Check-in page of admin | 22 |
| Figure 6.7 | Check-out page of admin | 22 |
| Figure 6.8 | Room category list page in admin | 23 |
| Figure 6.9 | Rooms page in admin | 23 |
| Figure 6.10 | Users page in admin | 24 |
| Figure 6.11 | Site settings page in admin | 24 |

LIST OF TABLES

| Table No. | Description | Page |
|------------------|-----------------------|-------------|
| Table 6.1 | Checked table | 25 |
| Table 6.2 | Rooms table | 25 |
| Table 6.3 | Room_categories table | 26 |
| Table 6.4 | System_settings | 26 |
| Table 6.5 | Users table | 27 |

Chapter 1

INTRODUCTION

Our project is a to create a hotel management system which will make booking a hotel a seamless and hassle-free experience and the purpose of this project is to make a hotel management system to make it easier to manage rooms in a hotel. It also provides a smooth experience to the admin who is able to view all the information in regards to rooms, reservations and users currently using the system.

With the increasing trend of “all things online” our system enables that and also reduces person to person contact through a digital method. This system is designed to not only store and display the result of the student, but also can be used to store the basic information of the student.

Rooms can be booked based on their category and the capacity of people.

When booking the system checks the check-in and the check-out dates to see if there the room is booked. Users can book a hotel easily and with a seamless experience. An admin user can also manage the details of rooms, view bookings, view users, add rooms and delete rooms.

Chapter 2

LITERATURE SURVEY

2.1 Traditional File System

In the early days of computing, data management and storage were a very new concept for organizations. The traditional approach to data handling offered a lot of the convenience of the manual approach to business processes (e.g., handwritten invoices & account statements, etc.) as well as the benefits of storing data electronically.

The traditional approach usually consisted of custom-built data processes and computer information systems tailored for a specific business function. An accounting department would have their own information system tailored to their needs, where the sales department would have an entirely separate system for their needs.

Initially, these separate systems were very simple to set up as they mostly mirrored the business process that departments had been doing for years but allowed them to do things faster with less work. However, once the systems were in use for so long, they became very difficult for individual departments to manage and rely on their data because there was no reliable system in place to enforce data standards or management.

Separate information systems for each business function also led to conflicts of interest within the company. Departments felt a great deal of ownership for the data that they collected, processed, and managed which caused many issues among company-wide collaboration and data sharing.

2.2 Pros and Cons of the Traditional Approach

2.2.1 Pros

➤ **Simple**

- Matched existing business processes and functions.
- Companies were not as interested in funding complicated information systems

➤ **Initially low-cost**

- Early computing was not viewed as beneficial for large funding.
- Systems were designed to be cheap in order to save on cost.

2.2.2 Cons

➤ **Separated ownership**

- Business functions had a high sense of data ownership.
- Departments unwilling to share data for fear of minimizing their superiority.

➤ **Unmanaged redundancy**

- Multiple instances of the same data appeared throughout various files, systems, and databases.
- Information updated in one place was not replicated to the other locations.
- Disk space was very expensive, and redundancy had a big impact on storage.

➤ **Data in consistency**

- Redundant data stored in various locations was usually never stored the same way.
- Formatting was not centrally managed.

➤ **Lack of data sharing**

- Same data stored in multiple locations.
- Caused unnecessary doubling of efforts for processing and managing data.

➤ **High costs in the long run**

- Hiring data processors for each department was very expensive, and each position was typically working on the same thing just for a different area.
- Doubling of work as well as excessive maintenance cost.

2.3 Downfall of Traditional Management System

Conceived in a relatively centralized era when software was deployed in static environments, legacy database architectures fail to support an increasingly mobile world where applications are accessed anytime, anywhere.

Today software users want consistent improvements in usability and expect SaaS vendors to deliver new features and functionalities needed to achieve their business objectives. However, legacy database technologies fall short in serving the needs of today's distributed and cloud environments for the following reasons:

- Inadequate fail over capabilities
- Insufficient provisions during peak demands Latency issues
- Lack of high availability at all times Increasing operational costs
- Inability to meet the demands of global markets

For all of these reasons, traditional databases are unable to deliver results in a rapidly growing environment where the workload is geographically distributed across heterogeneous datacenters. Upgrading to a more distributed data model is costly and complicated and your DBAs can't just sit back and give up on this situation. Hence, due to these various reasons, the downfall of the traditional system was inevitable.

2.4 Introduction to the Database Management System

A database management system (DBMS) refers to the technology for creating and managing databases. Basically, a DBMS is a software tool to organize (create, retrieve, update and manage) data in a database.

The main aim of a DBMS is to supply a way to store and retrieve database information that is both convenient and efficient. By data, we mean known facts that can be recorded and that have embedded meaning. Normally people use software such as DBASE IV or V, Microsoft ACCESS, or EXCEL to store data in the form of database. A datum is a unit of Data. Meaningful data combines to form Information. Hence, information is interpreted data- data provided with semantics. MS ACCESS is one of the most common examples of database management software. Database systems are meant to handle large collection of information Management of data

involves both defining structures for storage of information and providing mechanisms that can do the manipulation of those stored information. Moreover, the database system must ensure the safety of the information stored, despite system crash or attempts at unauthorized access.

2.5 Indicative areas for the use of a DBMS

Airlines: reservations, schedules etc.

Telecom: calls made, customer details, network usage etc. Universities registration, results, grades, etc.

Sales: products, purchases, customers etc. Banking: all transactions.

2.6 Advantages of a DBMS

A Database Management System has many advantages over the traditional file system used in the earlier days, such as:

Data independence: Application programs should be as free or independent as possible from details of data representation and storage. DBMS can supply an abstract view of the data for insulating application code from such facts.

Efficient data access: DBMS utilize a mixture of sophisticated concepts and techniques for storing and retrieving data competently and this feature becomes important in cases where the data is stored on external storage devices.

Data integrity and Security: If data is accessed through the DBMS, the DBMS can enforce integrity constraint on the data

Data administration: When several users share the data, integrating the administration of data can offer major improvements. Experienced professionals understand the nature of the data being managed and can be responsible for organizing the data representation to reduce redundancy and make the data to retrieve efficiently

2.7 Components of a DBMS

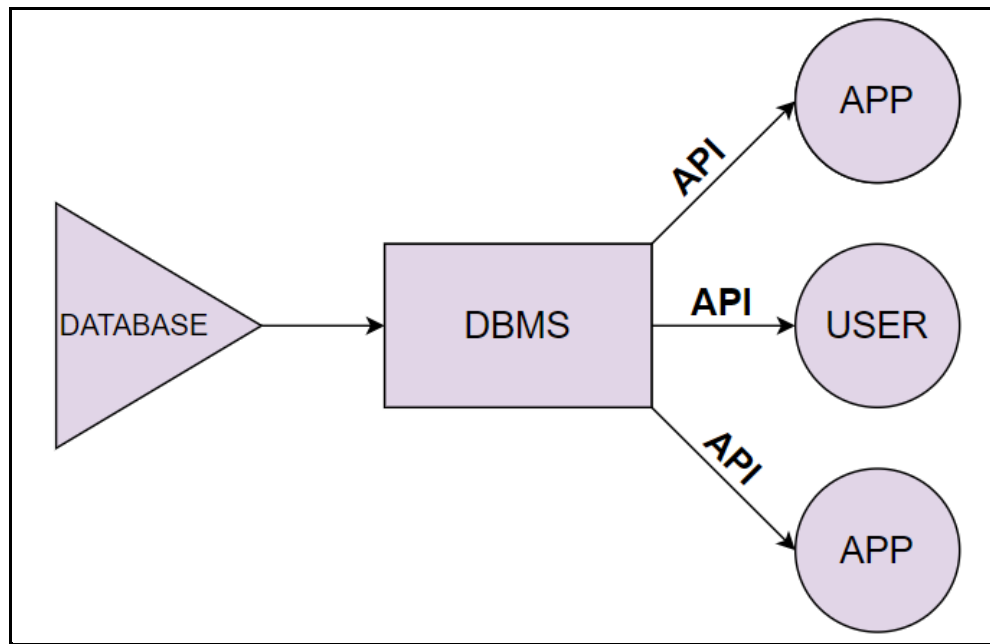


Figure 2.1: Components of DBMS

- **Users:** Users may be of any kind, such as data base administrators, system developers or database users.
- **Database application:** Database application may be Departmental, Personal, Organizational and /or Internal
- **DBMS:** Software that allows users to create and manipulate database access.
- **Database:** Collection of logical data as a single unit.

Chapter 3

SYSTEM REQUIREMENTS

The main purpose of this SRS document is to illustrate the requirements of the project Student information System and is intended to help any organization to maintain and manage its student's personal data.

3.1 Hardware Requirements

- Processor: Qualcomm Snapdragon processor
- Processor Speed: 1.4 GHz
- RAM: 2 GB
- Storage Space: 10 GB
- Display Resolution: 1024*768
- I/O Elements: Camera, Speaker, Microphone, GPS Network: 5 Mbps

3.2 Software Requirements

- A PHP engine: Version 5. Included in XAMPP-Windows.
- A web server: Apache HTTP Server 2.2 is recommended. Included in XAMPP Windows.
- A database server: MySQL Server 5.0 is recommended. Included in XAMPP Windows.
- A PHP debugger(optional): XDebug 2.0 or later.

Chapter 4

SYSTEM DESIGN

System design is the process of designing the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system.

4.1 Entity Relation Diagram

An entity–relationship model (ER model) describes the relationship of empty set stored in a database. An entity in this context is a component of data. In other words, ER diagrams illustrate the logical structure of databases. At first glance an entity relationship diagram looks very much like a flowchart. It is the specialized symbols, and the meanings of those symbols, that make it unique.

The figure below shows the different symbols and notations for ER Diagram.

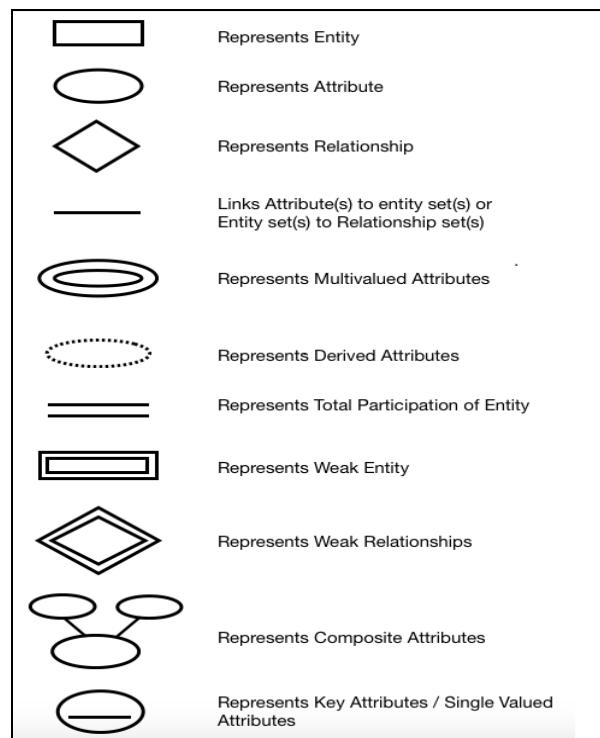


Figure 4.1: Symbols and Notations for ER Diagram

4.2 Description of ER Diagram

The figure below shows the ER Diagram of Hotel management which has 5 main entities which are:

- Checked
- Room_categories
- System_settings
- Rooms
- Users

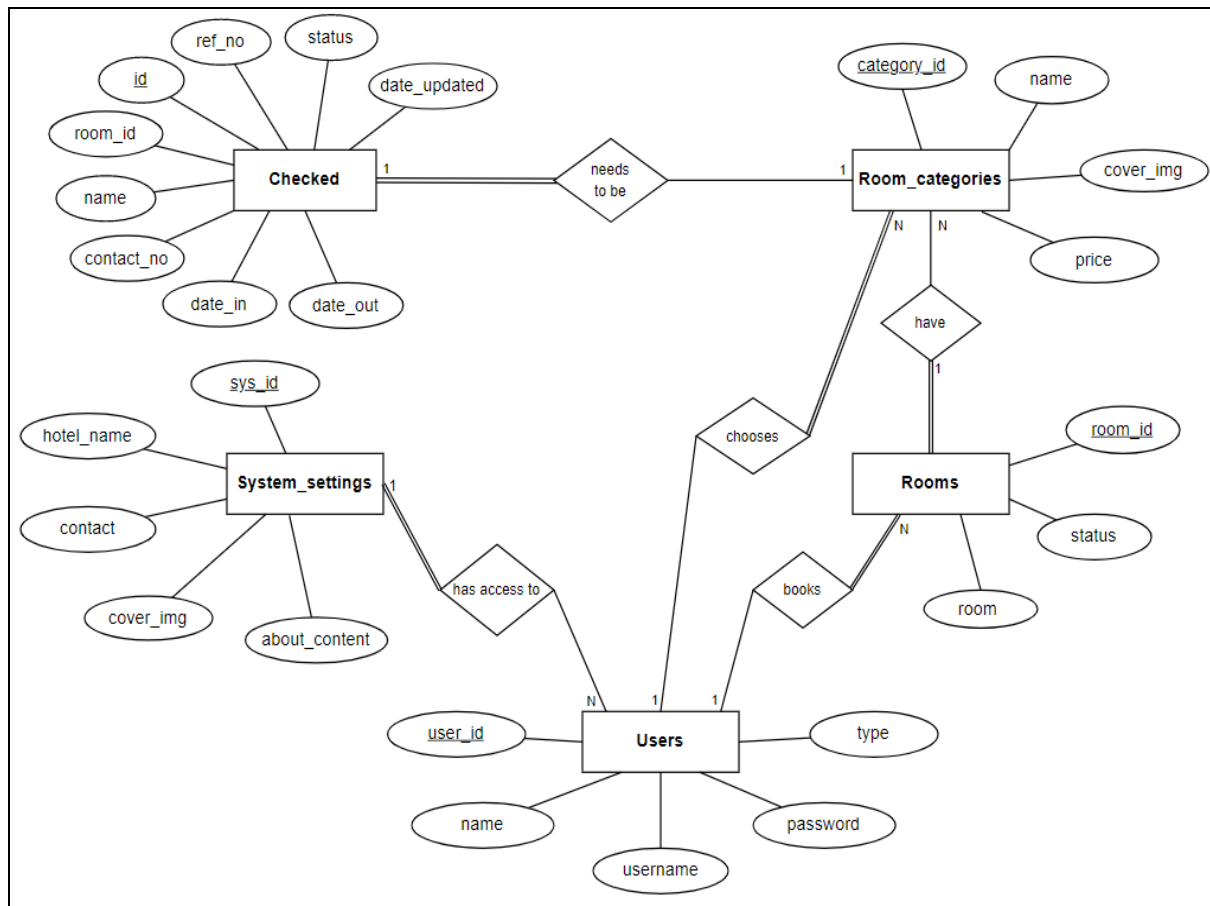


Figure 4.2: ER Diagram

4.2 Schema Diagram

A schema contains schema objects, which could be tables, columns, data types, store procedures, relationships, primary keys, foreign keys. A database schema can be represented in a visual diagram, which shows the database objects and their relationship with each other.

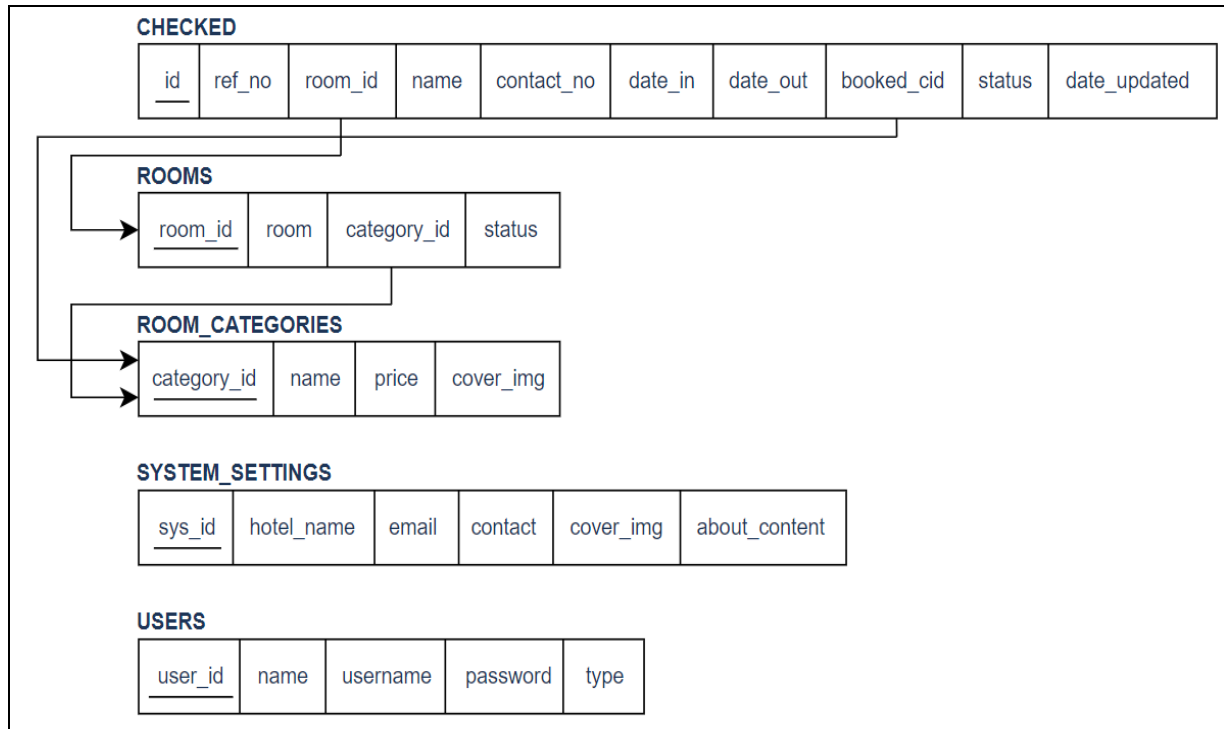


Figure 4.3: Schema Diagram

The above figure shows the schema diagram of the hotel management system in which it has 6 tables namely checked, rooms, room_categories, system_settings and users

Chapter 5

IMPLEMENTATION

5.1 HTML5

HTML5 is a markup language used for structuring and presenting content on the World Wide Web. It is the fifth and current major version of the HTML standard. It was published in October 2014 by the World Wide Web Consortium (W3C) to improve the language with support for the latest multimedia, while keeping it both easily readable by humans and consistently understood by computers and devices such as web browsers, parsers, etc. HTML5 is intended to subsume not only HTML 4, but also XHTML 1 and DOM Level 2 HTML.

HTML5 includes detailed processing models to encourage more interoperable implementations; it extends, improves and rationalizes the markup available for documents, and introduces markup and application programming interfaces (APIs) for complex web applications. Many new syntactic features are included. To natively include and handle multimedia and graphical content, the new <video>, <audio> and <canvas> elements were added, and support for scalable vector graphics (SVG) content and MathML for mathematical formulas. To enrich the semantic content of documents, new page structure elements such as <main>, <section>, <article>, <header>, <footer>, <aside>, <nav> and <figure>, are added. New attributes are introduced, some elements and attributes have been removed, and others such as <a>, <cite> and <menu> have been changed, redefined or standardized.

The APIs and Document Object Model (DOM) are now fundamental parts of the HTML5 specification and HTML5 also better defines the processing for, any invalid documents.

5.2 PHP

PHP is a servlet-side scripting language designed primarily for web development but also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994, the PHP reference implementation is now produced by The PHP Development Team. PHP original stood for Personal Home Page, but it now stands for the recursive acronym PHP: Hypertext Pre-processor.

PHP code may be embedded into HTML or HTML5 markup, or it can be used in combination with various web template systems, web content management systems and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the webserver.

Common Gateway interface (CGI) executable. PHP code may also be executed with a command line interface (CLI) and can be used to implement standalone graphical applications.

The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

The PHP language evolved without a written formal specification or standard until 2014, leaving the canonical PHP interpreter as a de facto standard. Since 2014 work has gone on to create a formal PHP specification. PHP is a scripting language that helps people make web pages more interactive by allowing them to do more things.

A website programmed with PHP can have pages that are password protected. A website with no programming cannot do this without other complex things. Standard PHP file extensions are: php, php3or, but a web server can be set up to use any extension.

5.3 SQL (Structured Query Language)

SQL (Structured Query Language) is a domain-specific language used in programming and designed for managing data held in a relational database Management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS). In comparison to older read/write APIs like [SAM or VSAM, SQL offers two main advantages: first, it introduced the concept of accessing many records with one single command; and second, it eliminates the need to specify how to reach a record, e.g. with or without an index.

Originally based upon relational algebra and tuple relational calculus, SQL consists of a data definition language, data manipulation language, and data control language. Although SQL is often described as, and to a great extent is, a declarative language (4GL), it also includes procedural elements.

SQL became a standard of the American National Standards Institute (ANSI) in 1986 and of the International Organization for Standardization (ISO) in 1987. Since then, the standard

has been revised to include a larger set of features. Despite the existence of such standards, most SQL code is not completely portable among different data base systems without adjustments.

5.4 Code Snippets

5.4.1 Pseudo Code for user

Check if a `date_in` value is set in the POST data, if yes, assign it to the variable `$date_in`. If not, set the current date as the value of `$date_in`.

Check if a `date_out` value is set in the POST data, if yes, assign it to the variable `$date_out`. If not, set the current date + 3 days as the value of `$date_out`.

Display the header section with a title "Rooms" and a horizontal line.

Start a form with action to "index.php?page=list" and method "POST"

Create a row with 3 columns

In the first column, display a label "Check-in Date" and an input field for the check-in date with a class "form-control datepicker" and name "date_in" and autocomplete off, and set the value to the current `date_in`.

In the second column, display a label "Check-out Date" and an input field for the check-out date with a class "form-control datepicker" and name "date_out" and autocomplete off, and set the value to the current `date_out`.

In the third column, display a button with class "btn btn-block btn-primary mt-3" and text "Check Availability"

Close the form

Display a horizontal line

Fetch all room categories from the "room_categories" table and store them in an array

Fetch all available rooms that are not booked for the selected check-in and check-out dates, and display them in a card layout

Close the card layout

5.4.2 Connectivity to database

```
<?php
//      DB      credentials. define('DB_HOST','localhost'
); define('DB_USER','root');
define('DB_PASS','');
define('DB_NAME','srms');
// Establish database connection. try
{
$dbh  =                new
PDO("mysql:host=".DB_HOST.";dbname=".DB_NAME,DB_USER,
DB_PASS,array(PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES
'utf8'"));
}
catch (PDOException $e)
{
exit("Error: " . $e->getMessage());
}
?>
```

5.4.3 Pseudo code for check-in page

Include the database connection file (db_connect.php)

Check if the GET variable 'id' is set

If GET variable 'id' is set, assign its value to the variable '\$id'

Run a query to select all data from the 'checked' table where the 'id' column is equal to the value of '\$id'

If the query returns any data, store each column value in a variable with the same name as the column

Check if the value of '\$room_id' is greater than 0

If the value of '\$room_id' is greater than 0, run a query to select all data from the 'rooms' table

where the 'id' column is equal to the value of '\$room_id' and store the result in the '\$room' variable. Then run a query to select all data from the 'room_categories' table where the 'id' column is equal to the 'category_id' of the '\$room' variable and store the result in the '\$cat' variable.

If the value of '\$room_id' is not greater than 0, run a query to select all data from the 'room_categories' table where the 'id' column is equal to the value of '\$booked_cid' and store the result in the '\$cat' variable.

Calculate the number of days between the '\$date_out' and '\$date_in' variable and store the result in the '\$calc_days' variable

Output the data in HTML format with PHP code interleaved to display the values of the variables

Check if the GET variable 'checkout' is set and the value of the 'status' variable is not equal to 2

If the condition is true, output HTML buttons for 'checkout' and 'edit_checkin'

Always output a 'Close' button in HTML

Add a 'click' event listener to the 'edit_checkin' button to open a modal and load the 'manage_check_in.php' file with GET variables 'id' and 'rid'

Add a 'click' event listener to the 'checkout' button to make an AJAX call to the 'ajax.php' file with the action 'save_checkout' and the GET variable 'id' with the value of '\$id'

On success of the AJAX call, show a success message and reload the page

On failure of the AJAX call, show an error message.

5.5 Stored Procedures

A procedure in SQL (often referred to as stored procedure), is a reusable unit that encapsulates the specific business logic of the application. A SQL procedure is a group of SQL statements and logic, compiled and stored together to perform a specific task.

Syntax :- `CREATE PROCEDURE procedure_name AS
 sql_statement
 GO;`

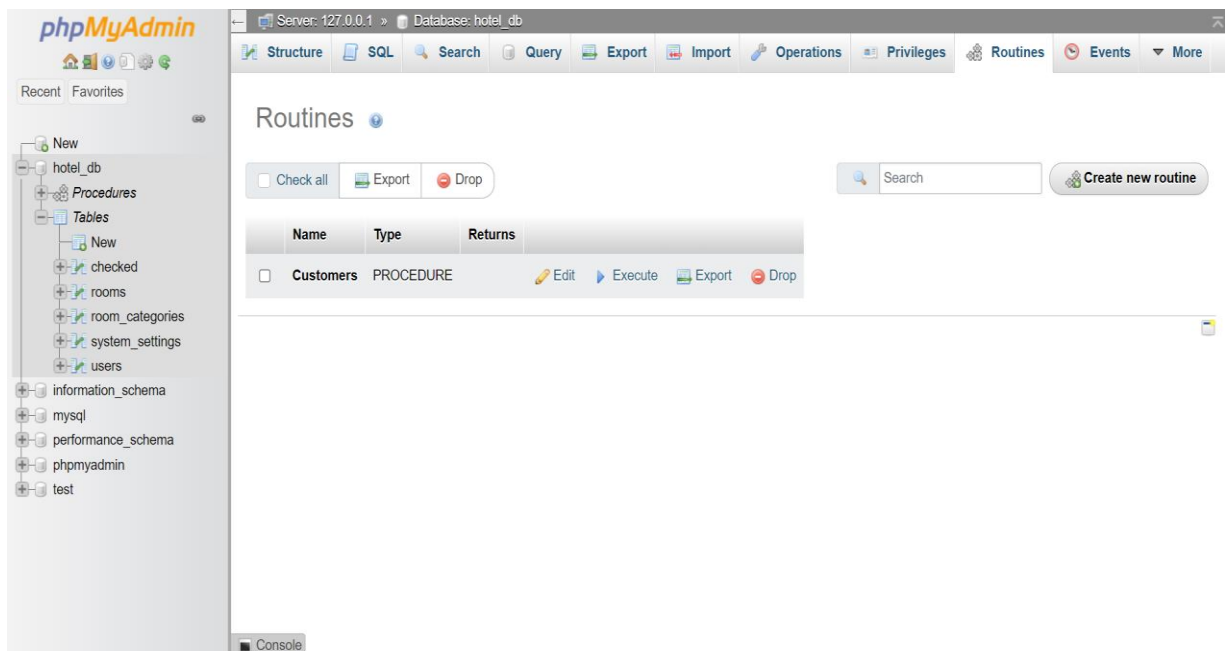


Figure 5.1: Stored procedure

The above figure 5.5.1 shows a stored procedure called customers where we can store the data of all the customers who have booked rooms in the hotel.

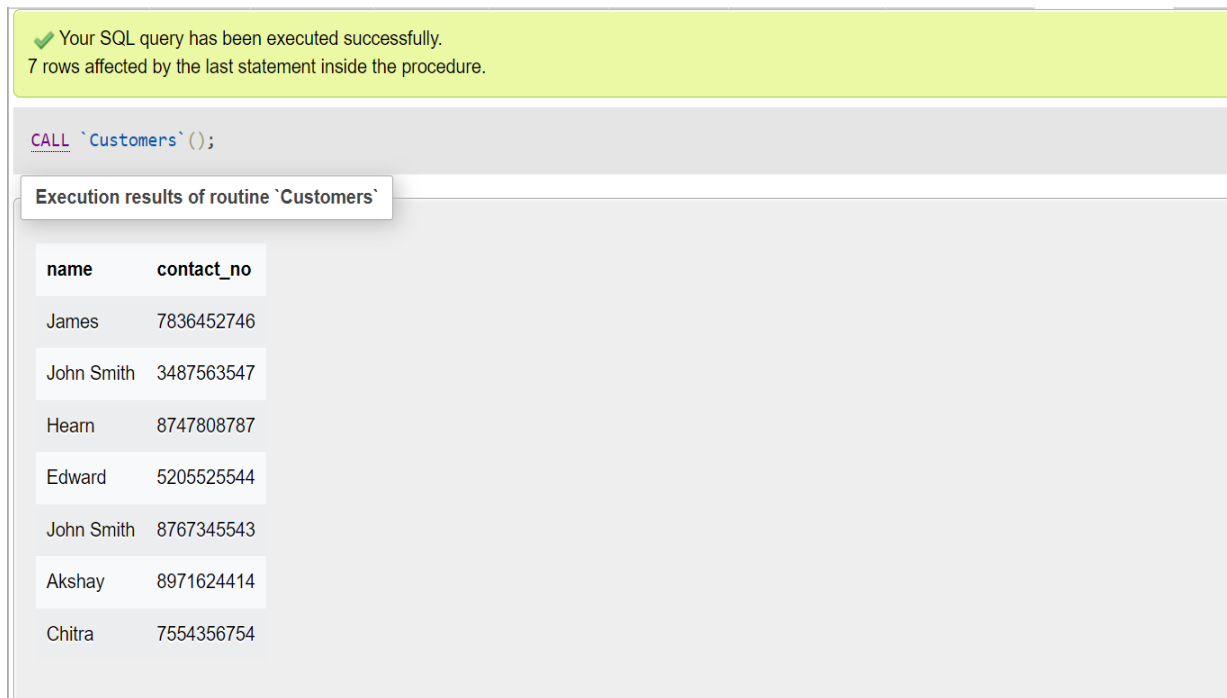


Figure 5.2 Data in stored procedure

The figure 5.5.2 shows all the details i.e., the name and the contact number of the customers who have booked a room in the hotel.

5.6 Triggers

Trigger is a statement that a system executes automatically when there is any modification to the database. In a trigger, we first specify when the trigger is to be executed and then the action to be performed when the trigger executes. Triggers are used to specify certain integrity constraint and referential constraints that cannot be specified using the constraint mechanism of SQL.

Syntax:

```
CREATE TRIGGER schema.trigger_name
ON table_name AFTER {INSERT, UPDATE, DELETE}
[NOT FOR REPLICATION]

AS

{SQL_Statements}
```

Trigger code:

```
DROP TRIGGER IF EXISTS `checked_before_time`;
```

```
CREATE DEFINER=`root`@`localhost` TRIGGER `check_before_time` BEFORE INSERT  
ON `checked` FOR EACH ROW IF (CAST(DATE_FORMAT(NEW.date_in, "%H") AS INT)  
< 11) THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Checkin before time';  
END IF
```

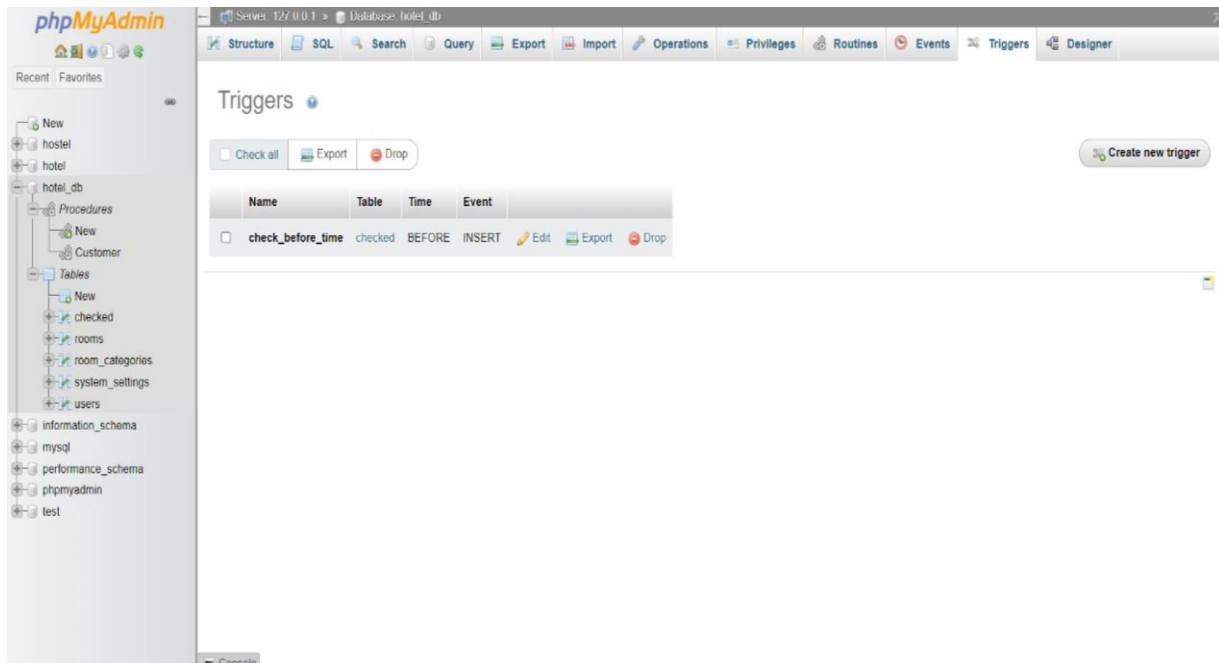


Figure 5.3: Trigger

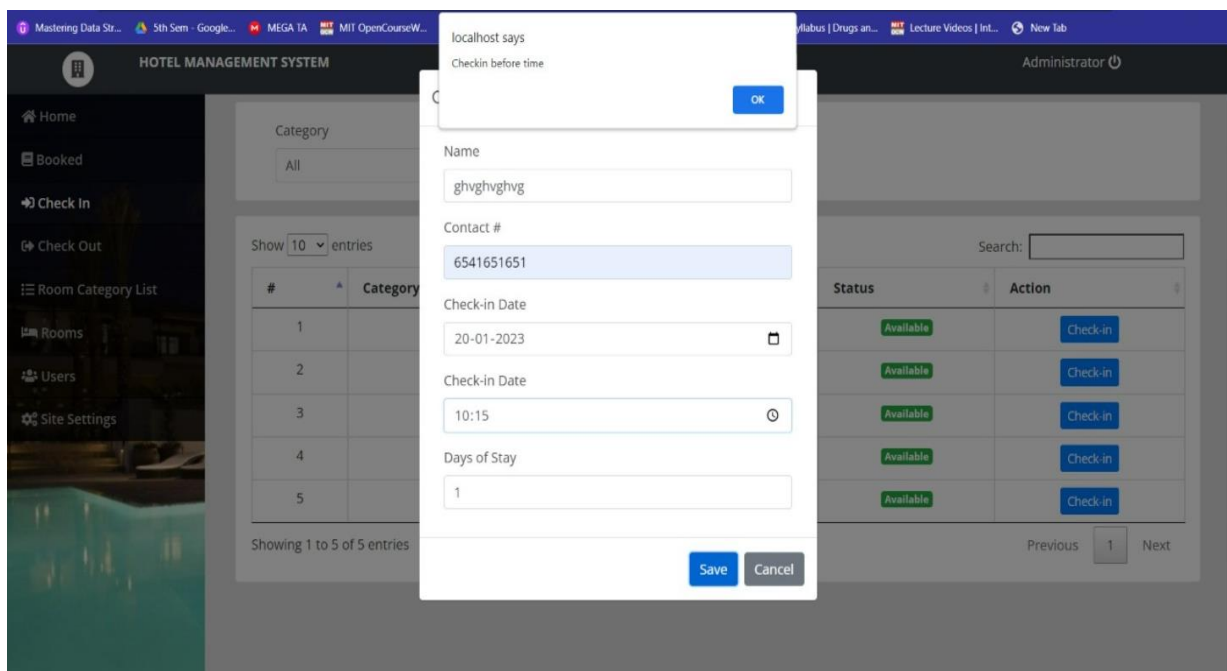


Figure 5.4: Trigger execution on webpage

Chapter 6

SNAPSHOTS

6.1 Home page of user

This is the user's home page and of the hotel management system where the user can enter the check-in date and check-out date and then check the availability of the rooms.

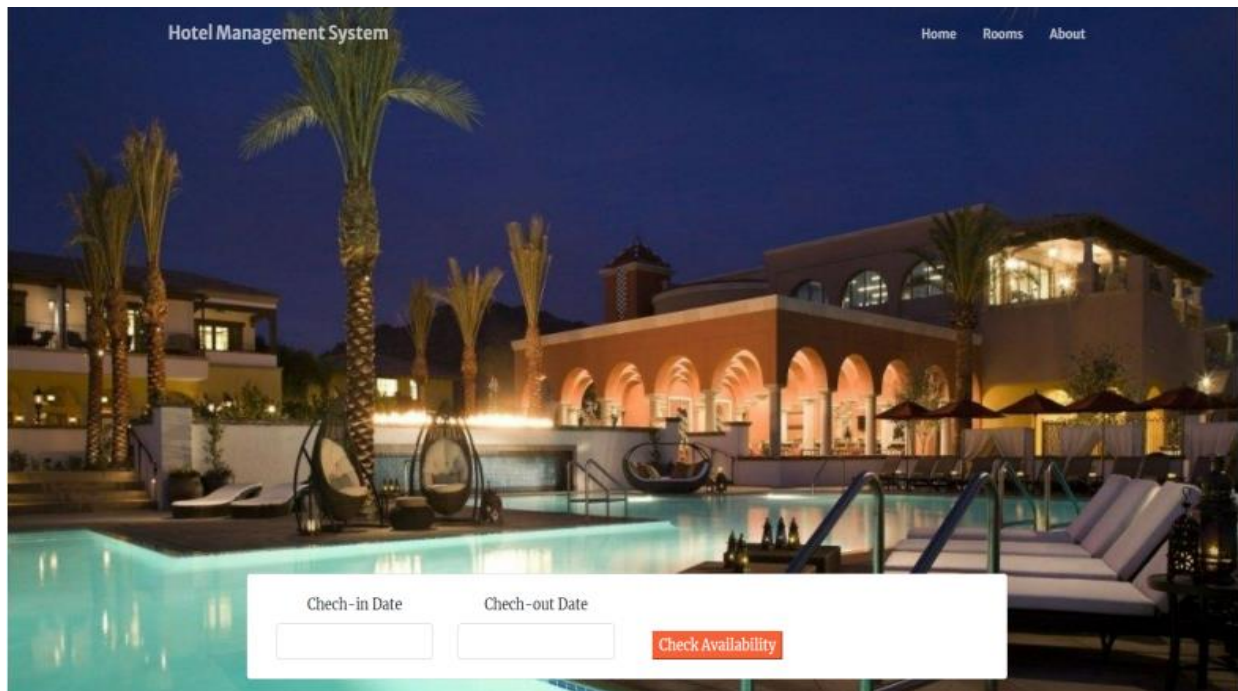


Figure 6.1: Home page of user

6.2 Hotel rooms

This is a page which displays the different hotel rooms that the users can book according to their given dates of stay.

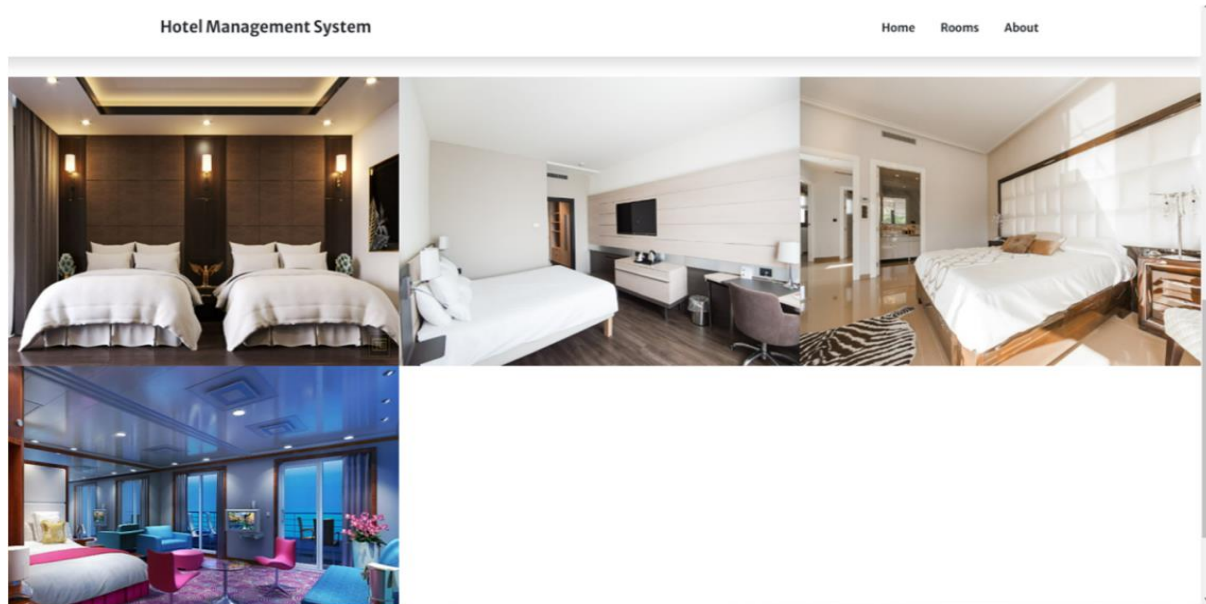


Figure 6.2: Hotel rooms

6.3 Hotel rooms with prices

This page shows all the hotel rooms along with prices of the particular room that the user can book accordingly.

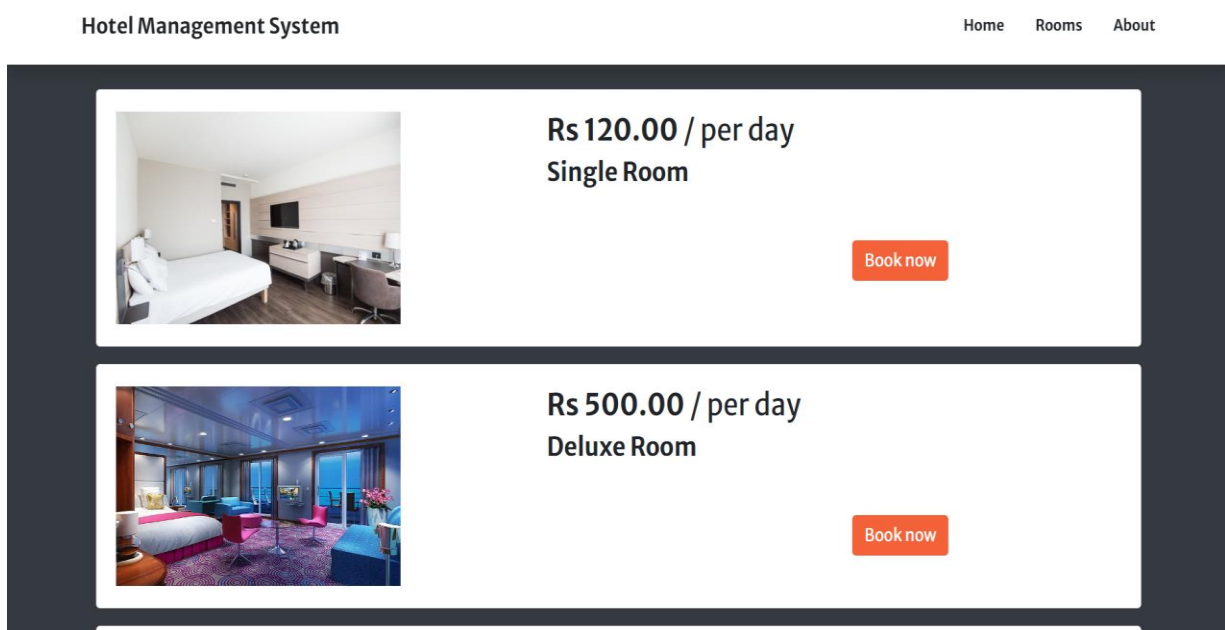


Figure 6.3: Hotel rooms with prices

6.4 Detailed section

This page shows the details that user will be entering after selecting the type of room to be booked.

Hotel Management System

Book

Name: Sharanya

Contact #: 8793453442

Check-in Date: 26-01-2023

Check-in Time: 18:12

Days of Stay: 4

Save **Cancel**

Home Rooms About

Book now

Book now

Figure 6.4: Detailed section

6.5 Booked page of admin

This page shows all the booked rooms along with the details, reference and the current status of the room.

HOTEL MANAGEMENT SYSTEM

Administrator

Show 10 entries

Search:

| # | Category | Reference | Status | Action |
|---|---------------|------------|--------|----------------------|
| 1 | Deluxe Room | 6184351813 | Booked | View |
| 2 | Single Room | 9693823437 | Booked | View |
| 3 | Twin Bed Room | 6680841678 | Booked | View |
| 4 | Twin Bed Room | 7139446439 | Booked | View |
| 5 | Single Room | 2270611868 | Booked | View |

Showing 1 to 5 of 5 entries

Previous 1 Next

Home Booked Check In Check Out Room Category List Rooms Users Site Settings

Figure 6.5: Booked page of admin

6.6 Check-in page in admin

This page shows the check-in details about the room, room category, room number and also the status of the room.

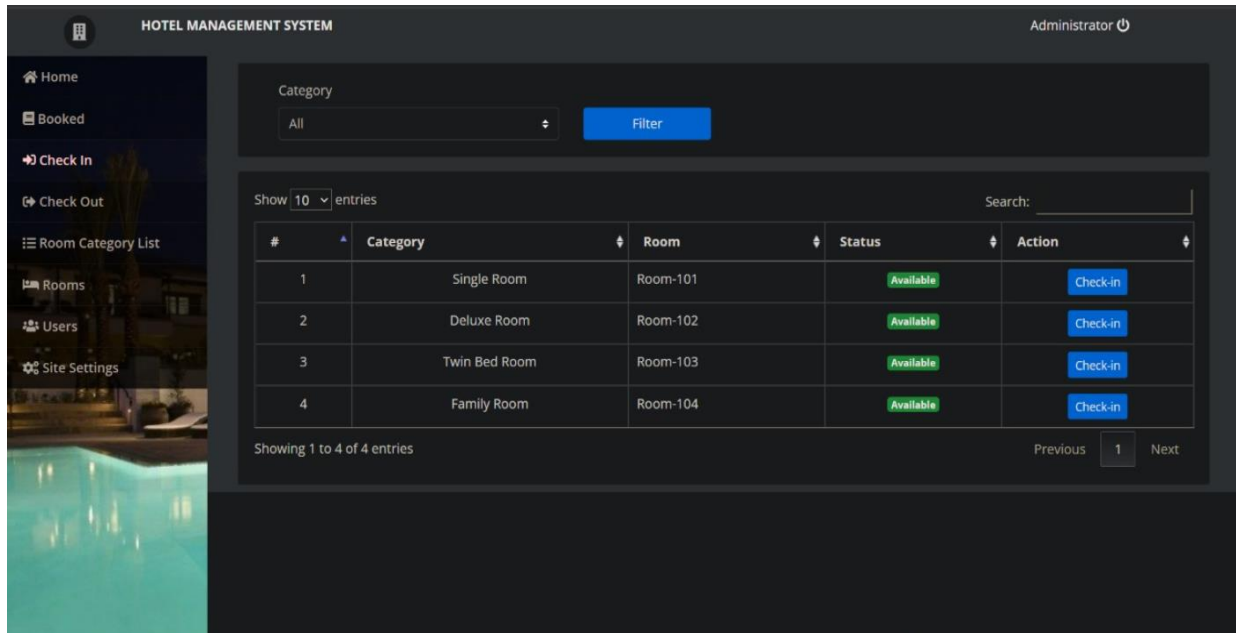


Figure 6.6: Check-in page of admin

6.7 Check out page in admin

This page shows the check-out details about the room, room category, room number, status of the room and he reference number.

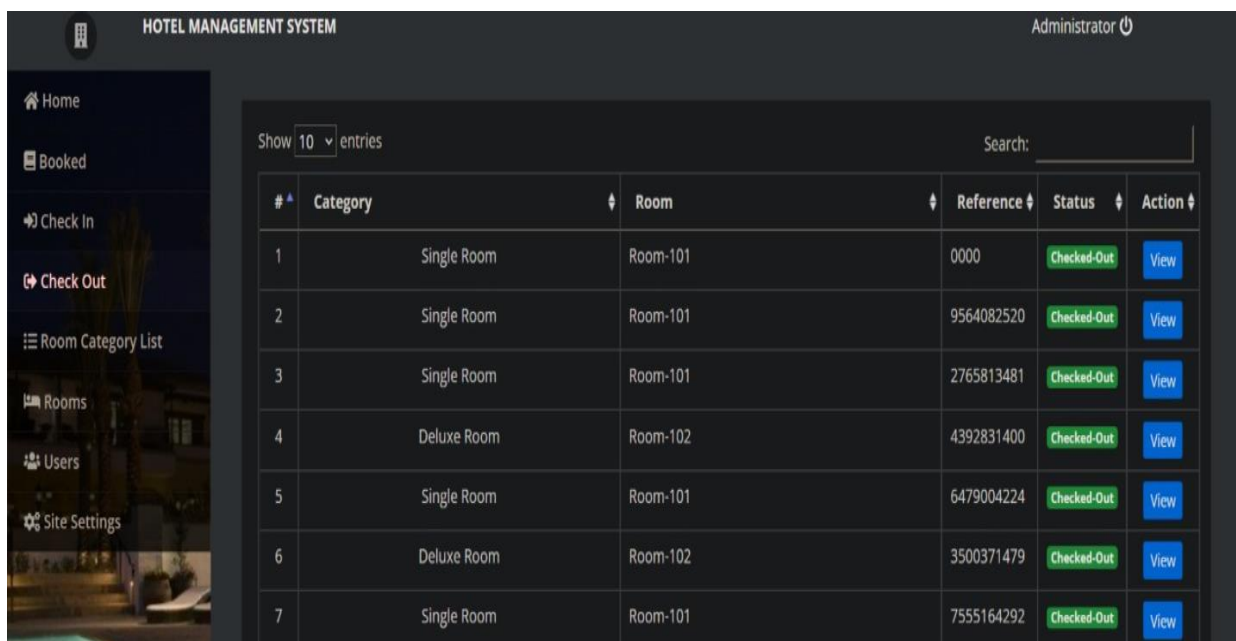


Figure 6.7: Check-out page in admin

6.8 Room category list page in admin

This page shows that the admin can add more room categories and also edit the existing ones along with the prices.

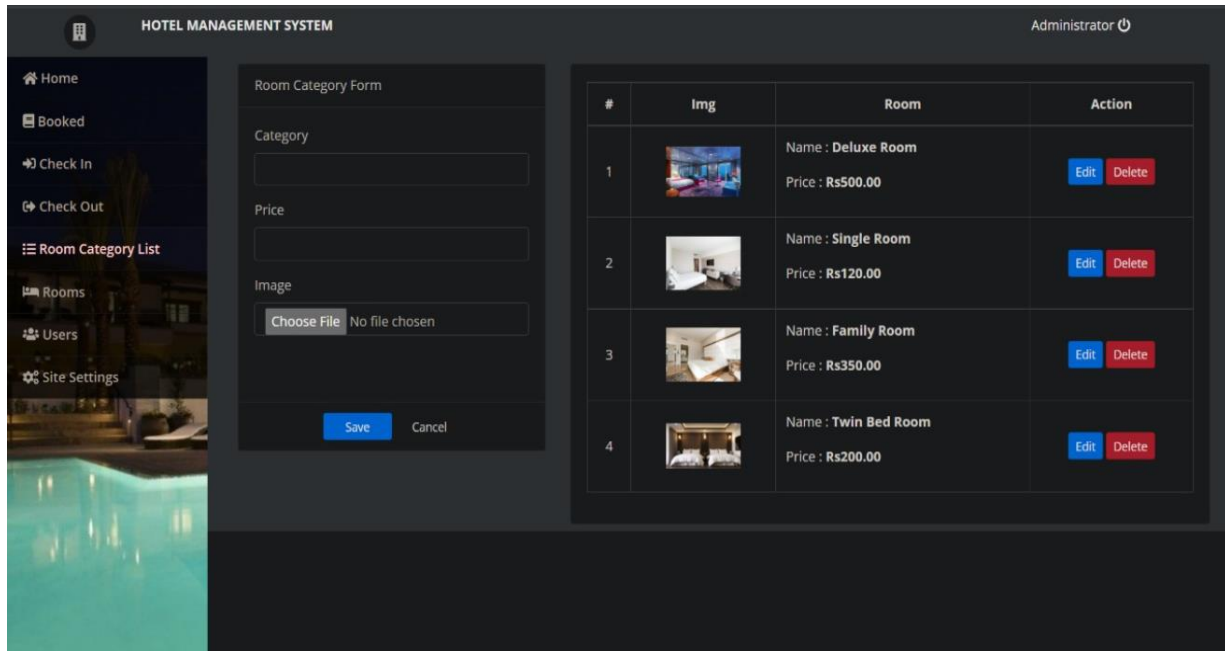


Figure 6.8: Room category list page in admin

6.9 Rooms List page in admin

This page shows that the admin can add more rooms in a particular type of room category and also edit the previously created rooms.

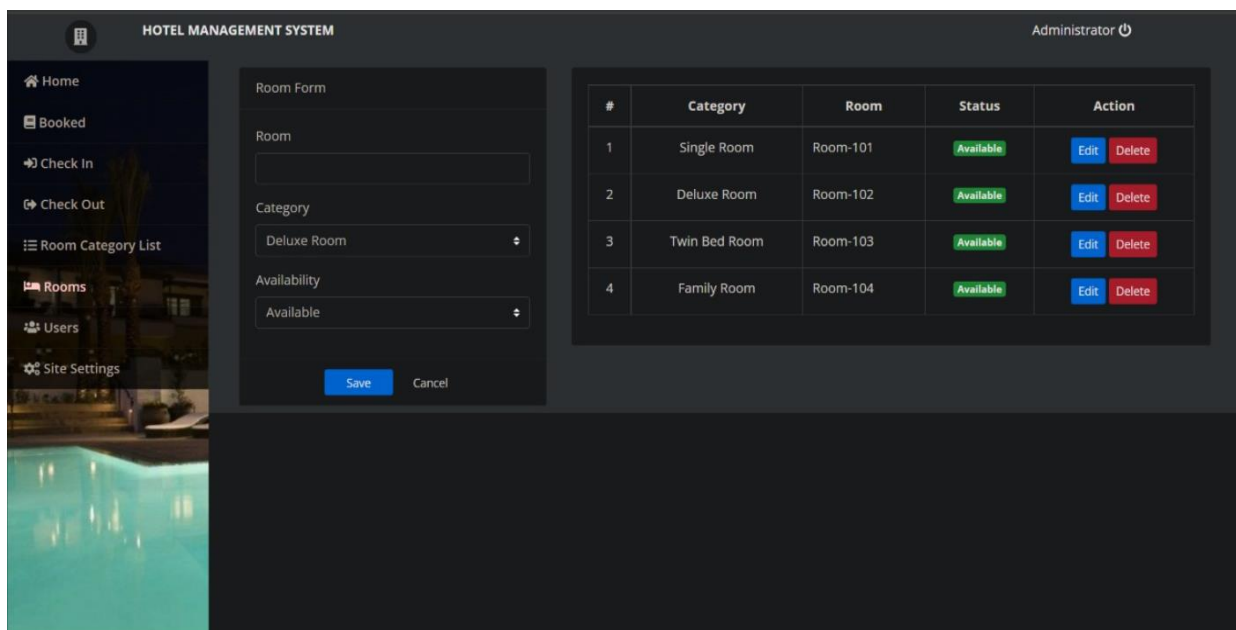


Figure 6.9: Rooms page in admin

6.10 Users page in admin

This page shows the name, username and the actions of the admin. Here we can add more users of different ions like admin or just the staff. If the new user is the admin then password can also be set in this page

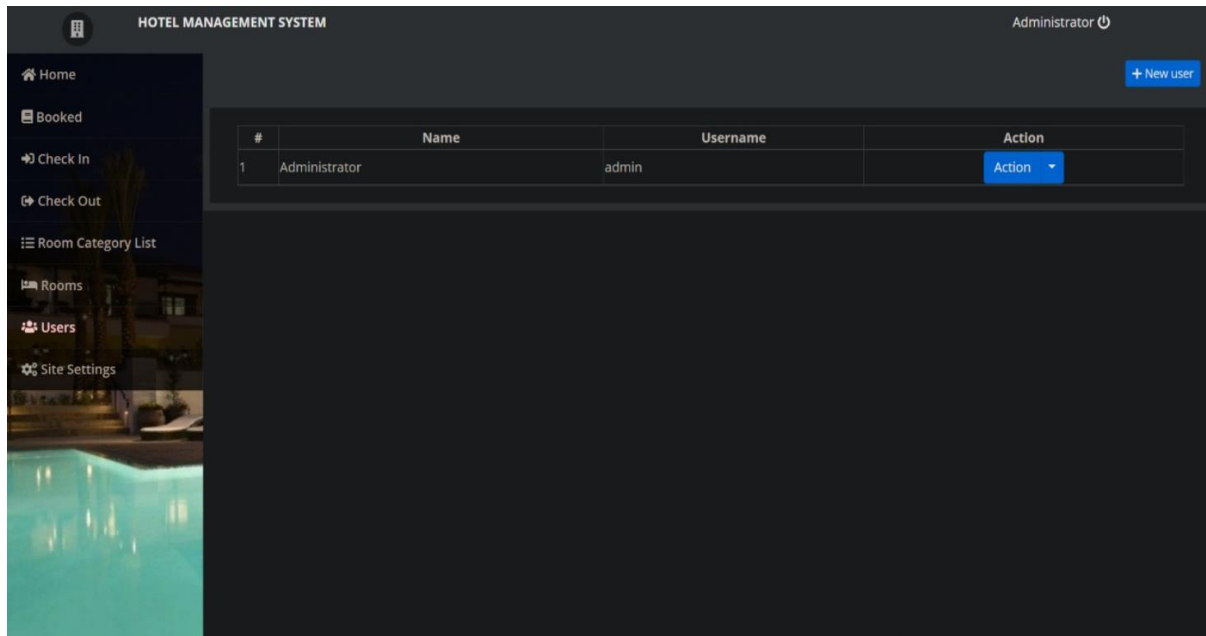


Figure 6.10: User page in admin

6.11 Site settings page in admin

This page shows that in this page the site name, email address and contact can be changed. We can also write about the website here which will be displayed on the actual website.

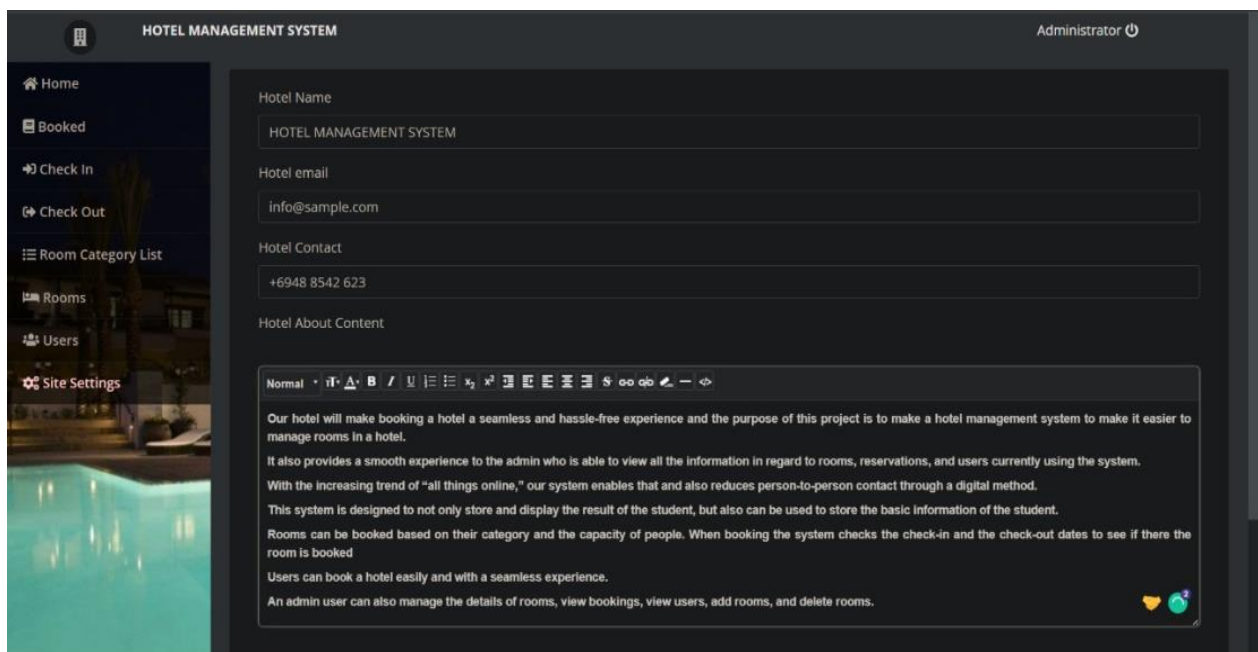


Figure 6.11: Site settings page in admin

6.12 Table Structure

6.12.1 Checked table

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|----|--------------|--------------|--------------------|------------|------|---------------------|---|-------------------------------|--------------------|
| 1 | id | int(30) | | | No | None | | AUTO_INCREMENT | Change Drop More |
| 2 | ref_no | varchar(100) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 3 | room_id | int(30) | | | No | None | | | Change Drop More |
| 4 | name | text | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 5 | contact_no | varchar(20) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 6 | date_in | datetime | | | No | None | | | Change Drop More |
| 7 | date_out | datetime | | | No | None | | | Change Drop More |
| 8 | booked_cid | int(30) | | | No | None | | | Change Drop More |
| 9 | status | tinyint(1) | | | No | 0 | 0 = pending, 1=checked in , 2 = checked out | | Change Drop More |
| 10 | date_updated | datetime | | | No | current_timestamp() | | ON UPDATE CURRENT_TIMESTAMP() | Change Drop More |

Table 6.1: Checked table

Table 6.12.1 shows the checked table which contains 10 attributes: id, ref_no, room_id, name, contact_no, date_in, date_out, booked_cid, status, date_updated.

Here, id of the customer is the primary key, room_id and booked_cid are the foreign keys.

6.12.2 Rooms table

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|-------------|-------------|--------------------|------------|------|---------|---------------------------------|----------------|--------------------|
| 1 | id | int(30) | | | No | None | | AUTO_INCREMENT | Change Drop More |
| 2 | room | varchar(30) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 3 | category_id | int(30) | | | No | None | | | Change Drop More |
| 4 | status | tinyint(1) | | | No | 0 | 0 = Available , 1= Unavailables | | Change Drop More |

Table 6.2: Rooms table

This table 6.12.2, consists of 4 attributes: id, room, category_id, status.

In this table id of the room is the primary key and category_id is the foreign key.

6.12.3 Room_categories table














| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|--|---------|--------------------|------------|------|---------|----------|----------------|--|
| 1 | id  | int(30) | | | No | None | | AUTO_INCREMENT |  Change  Drop  More |
| 2 | name | text | utf8mb4_general_ci | | No | None | | |  Change  Drop  More |
| 3 | price | float | | | No | None | | |  Change  Drop  More |
| 4 | cover_img | text | utf8mb4_general_ci | | No | None | | |  Change  Drop  More |

Table 6.3: Room_categories table

This table 6.12.3, consists of 4 attributes: id, name, price, cover_img.

Here id of the category of the room is the primary key.

6.12.4 System_settings table




















| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|--|--------------|--------------------|------------|------|---------|----------|----------------|--|
| 1 | id  | int(30) | | | No | None | | AUTO_INCREMENT |  Change  Drop  More |
| 2 | hotel_name | text | utf8mb4_general_ci | | No | None | | |  Change  Drop  More |
| 3 | email | varchar(200) | utf8mb4_general_ci | | No | None | | |  Change  Drop  More |
| 4 | contact | varchar(20) | utf8mb4_general_ci | | No | None | | |  Change  Drop  More |
| 5 | cover_img | text | utf8mb4_general_ci | | No | None | | |  Change  Drop  More |
| 6 | about_content | text | utf8mb4_general_ci | | No | None | | |  Change  Drop  More |

Table 6.4: System_settings table

This table 6.12.4, consists of 6 attributes: id, hotel_name, email, contact, cover_img, about_content.

Here id of the system is the primary key.

6.12.5 Users table

















| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|--------------|--------------------|------------|------|---------|---------------------|----------------|--|
| 1 | id  | int(30) | | | No | None | | AUTO_INCREMENT |  Change  Drop  More |
| 2 | name | varchar(200) | utf8mb4_general_ci | | No | None | | |  Change  Drop  More |
| 3 | username | varchar(100) | utf8mb4_general_ci | | No | None | | |  Change  Drop  More |
| 4 | password | varchar(200) | utf8mb4_general_ci | | No | None | | |  Change  Drop  More |
| 5 | type | tinyint(1) | | | No | 2 | 1=admin , 2 = staff | |  Change  Drop  More |

Table 6.5: Users table

This table 6.12.5, consists of 5 attributes: id, name, username, password, type.

Here id of the user is the primary key.

Chapter 7

CONCLUSIONS AND FUTURE ENHANCEMENTS

Hotel Management System is a Web-portal Development Company specializing in providing custom solutions for small businesses. We strive to build solutions to your specific needs to get the job done right the first time. We pay special attention to the ease of use and utilize the latest technology. We can add printers in future. We can give more advance software for Hotel Management System including more facilities. We will host the platform on online servers to make it accessible worldwide Integrate multiple load balancers to distribute the loads of the system.

Implement the backup mechanism for taking backup of codebase and database. on regular basis on different servers. The above-mentioned points are the enhancements which can be done to increase the applicability and usage of this project. Here we can maintain the records of Room and Customer. Also, as it can be seen that now-a-days the players are versatile.

So, there is a scope for introducing a method to maintain the Hotel Management System. Enhancements can be done to maintain all the Room, Customer, Booking, Tariffs, Food Bill.

REFERENCES

- [1] <http://www.phpknowhow.com/basics/working-with-xampp>
- [2] <https://www.geeksforgeeks.org/dbms/>
- [3] <http://www.w3schools/php.com>
- [4] <https://github.com/mpdf/mpdf>
- [5] <https://devdocs.io/>
- [6] <https://mobiosolutions.com/create-or-execute-stored-procedures-in-phpmyadmin/>