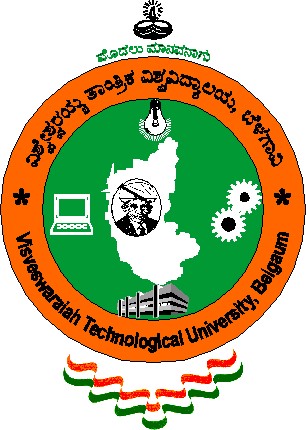
**VISVESVARAYA TECHNOLOGICAL UNIVERSITY,**

**BELAGAVI**



**Mini Project Report**

**On**

**“HOTEL MANAGEMENT SYSTEM”**

Submitted by

**HARSHITHA S (1EE21CS019)**

In partial fulfilment of

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

Under the guidance of

|  |  |
| --- | --- |
|  |  |
| Guide:  **Dr. Kumuda N S**  Associate Professor,  Dept. of CSE | HOD:  **Dr. Lavanya N L** Head,  Dept. of CSE |

# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

# EAST WEST COLLEGE OF ENGINEERING

Yelahanka New Town, Bangalore-560064

**2023-24**

## **ACKNOWLEDGEMENT**

Inspiration and guidance are valuable in all aspects of life, especially what is

academic. “Experience is the best teacher”, is an old saying. The satisfaction and pleasure that accompany the gain of experience would be incomplete without mentioning the people who made it possible.

We are extremely thankful and grateful to our guide **Dr. Kumudha N S, Professor, and Department of Computer Science and Engineering**. They being our guide has taken keen interest in the progress of the Project work by providing facilities and guidance. We are indebted to our guide for her inspiration, support and kindness showered throughout the course.

We express our profound sense of gratitude to **Dr. Lavanya N L HOD, Department of Computer Science and Engineering** for giving us the opportunity to pursue our interest in our Project work.

We express our special gratitude to our **Principal Dr. Santhosh Kumar G and Management** for providing the resources and support for making our Project work. Nevertheless, we express heartfelt thanks towards our parents, friends and teaching and non-teaching staff of our college for their kind co-operation and encouragement which helped us in the completion of our Project work.

**HARSHITHA S (1EE21CS019)**

# EAST WEST COLLEGE OF ENGINEERING

Yelahanka New Town, Bangalore-560064

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



### **CERTIFICATE**

### This is to certify thatHARSHITHA S (1EE21CS019) of 5th semester B.E (Computer Science & Engineering) has completed the Mini Project report on “HOTEL MANAGEMENT SYSTEM” in the partial fulfilment of MINI PROJECT prescribed by Visvesvaraya Technological University during the academic year 2023-24.

#### **Signature of Guide Signature of HOD**

**Dr. Kumuda N S Dr. Lavanya N L**

Associate Professor, Head,

Dept. of CSE Dept. of CSE

Signature of External

1)

2)

**ABSTRACT**

The Hotel management System is a Database Management System (DBMS) mini-project designed to facilitate efficient booking and management of rooms in various establishments such as hotels, conference centers, or educational institutions. This system incorporates both online and offline booking capabilities to cater to a diverse range of users.

The system consists of two main components: the user interface for booking rooms and an administrative panel for managing bookings, room availability, and user accounts. Users can access the system through a web-based interface or via an offline application, providing flexibility in booking options.

The Room Booking System is designed to streamline the booking process, enhance user experience, and improve operational efficiency for both administrators and users. By offering both online and offline booking options, the system caters to a wide range of users with varying preferences and accessibility needs.

I

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **SL.NO** | **CHAPTER** | **PAGE**  **NO** |
| 1 | Introduction | 1 |
| 2 | Analysis | 2-4 |
|  | 2.1 System Analysis | 2 |
|  | 2.2 System Specification | 3 |
| 3 | Design Approach | 5-10 |
|  | 3.1 Introduction To Design | 5 |
|  | 3.2 Dataflow Diagram | 6-7 |
|  | 3.3 Er Diagram | 7-9 |
|  | 3.4 Schema Diagram | 10 |
| 4 | Project Module | 11-14 |
| 5 | Implementation | 15-22 |
| 6 | Testing | 23-25 |
|  | 6.1 Test Cases | 25 |
| 7 | Result Analysis | 26-29 |
| 8 | Conclusion | 30 |
| 9 | Future Enhancement | 31 |
| 10 | Bibliography | 32 |

**II**

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **SL.NO** | **PARTICULARS** | **PAGE**  **NO** |
| 3.1 | Dataflow Diagram | 6 |
| 3.2 | Sequence diagram | 7 |
| 3.3 | Flow chart | 8 |
| 3.4 | Schema Diagram | 9 |
| 7.1 | User panel | 26 |
| 7.2 | Amenities page | 26 |
| 7.3 | Services page | 26 |
| 7.4 | Rooms Booking Page | 27 |
| 7.5 | Contact us Page | 27 |
| 7.6 | Admin login Page | 27 |
| 7.7 | Admin home Page | 28 |
| 7.8 | Profit Page | 28 |
| 7.9 | Reservation Page | 28 |
| 7.10 | Payment Page | 29 |
| 7.11 | Add Rooms Page | 29 |
| 7.12 | Invoice | 29 |
| 7.13 | Backend | 30 |

**III**

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **SL.NO** | **NAMES** | **PAGE**  **NO** |
| 1 | Room | 13 |
| 2 | Newsletterlog | 13 |
| 3 | Roombook | 14 |
| 4 | Contact | 14 |
| 5 | login | 15 |

**IV**

**CHAPTER 1**

**INTRODUCTION**

In the bustling hospitality industry, effective management of hotel operations is paramount to ensuring guest satisfaction and maximizing revenue. The Hotel Management System (HMS) serves as a comprehensive solution designed to streamline various facets of hotel operations through a robust Database Management System (DBMS). This project aims to address the complexities of hotel management by offering a user-friendly interface coupled with powerful functionalities.

The “Hotel Management System” is a hotel reservation site script where site users will be able to search rooms availability with an online booking reservations system. Site users can also browse hotels, view room inventory, check availability, and book reservations in real-time. Site users enter check in date and check out date then search for availability and rates. After choosing the right room in the wanted hotel – all booking and reservation process is done on the site and an SMS is sent to confirm the booking. Administrator Panel.

This system will help administration to work easily. Because of its easy access and less time consuming administration can get the information of the Users, Rooms, Payments, etc. They do not have to search in the paper file for the log time. Members an easily handle the system. The “HOTEL MANAGEMENT SYSTEM” is designed with user-friendliness in mind, ensuring that both hotel staff and guests can navigate the system with ease. The intuitive interface offers seamless navigation and accessibility features, making it effortless for users to perform tasks such as making reservations, checking room availability, updating guest profiles, and generating reports.

Moreover, the system supports multi-platform access, allowing users to interact with the HMS through web browsers, mobile applications, and desktop clients. This flexibility ensures that users can access the system anytime, anywhere, using their preferred devices, whether they are on-site or remotely. In conclusion, the Hotel Management System offers a comprehensive suite of features coupled with user-friendly accessibility, empowering hotels to optimize their operations, enhance guest experiences, and achieve sustainable growth in the competitive hospitality industry.

**MYSQL:**

MySQL Workbench is a unified visual tool for database architects, developers, and DBAs. MySQL Workbench provides data modeling, SQL development, and comprehensive administration tools for server configuration, user administration, backup, and much more. MySQL Workbench is available on Windows, Linux and

Mac OS X.

**PHP:**

Hypertext Preprocessor (or simply PHP) is a server-side scripting language designed for Web development, but also used as a general-purpose programming language. It was originally created by Rasmus Lerdorf in 1994,] the PHP reference implementation is now produced by The PHP Group. PHP originally stood for Personal Home Page,] but it now stands for the recursive acronym PHP: Hypertext Preprocessor.

**XAMPP:**

XAMPP is a free and open source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes.

**CHAPTER 2**

**ANALYSIS**

**2.1 SYSTEM ANALYSIS**

It is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components. System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose.

The objective of the system analysis activity is to develop structured system specification for the proposed system. The structured system specification should describe what the proposed system would do; independent of the technology, which will be used to implement these requirements. The structured system specification will be used to implement these requirements.

The essential model may itself consist of multiple models, modeling different aspect of the system. The data flow diagrams may model the data and there relationships and the state transition diagram may model time dependent behavior of the system. The essential model thus consists of the following.

* Context diagram
* Leveled data flow diagrams
* Process specification for elementary bubbles
* Data dictionary for the flow and stores on the DFDs.

It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose.

**2.1.1 Existing system**

Existing system is manual one in which we have developed with different protocol to store the information like user details, room details and class details who are willing to book the room . It is very difficult to maintain historical data.

This activity is also known as the feasibility study. Perform and evaluate feasibility studies like cost-benefit analysis, technical feasibility, time feasibility and operational feasibility for the project. It begins with a request from the user for a new system. It involves the following:

* Identify the responsible user for a new system
* Clarify the user request
* Identify deficiencies in the current system
* Prepare a project charter that will be used to guide the remainder of the Project.

**2.1.2 Proposed System**

This system will help administration to work easily. Because of its easy access and less time consuming administration can get the information of the Users, Rooms, Payments, etc. They do not have to search in the paper file for the log time. Members an easily handle the system.

**ADVANTAGES :**

* The advantages of booking a hotel online add up long before your arrival. Our legendary customer service extends to the web.
* One advantage of booking with the hotel directly is the use of the hotel's full cancellation policy as well as not needing a deposit in most situations.
* Read reviews and compare prices for Online Hotel Booking.
* The most important advantage of online hotel booking is convenience, you can book your room by simply sitting in home.

**2.1.3 OBJECTIVES OF THE SYSTEM**

* Reservation Management: Implement a system to manage room reservations including booking, modification, and cancellation.
* Guest Management: Maintain a database of guest information including personal details, contact information, and preferences.
* Room Management: Keep track of room availability, status (clean, dirty, under maintenance), and assign rooms to guests based on their preferences and availability.
* Billing and Invoicing: Develop functionalities to generate bills for guests including room charges, additional services, taxes, and any other applicable fees.
* Security and Access Control: Ensure data security by implementing appropriate access controls and authentication mechanisms to prevent unauthorized access to sensitive information.

**2.2 SYSTEM SPECIFICATIONS**

**2.2.1 HARDWARE REQUIRMENT**

* Pentium IV Processor
* 256 MB Ram
* 512 KB Cache Memory
* Hard disk 10 GB

**2.2.2 SOFTWARE REQUIRMENT**

* Operating System: Windows
* Web browser : Google Chrome, Mozilla, Windows Explorer
* Web-Technology : PHP
* Front-End : Html, CSS, Bootstrap, Sublime Text, Swiper js, Java Script
* Back-End : MY SQL yog
* Web Server : Apache SERVER, XAMPP SERVER

**CHAPTER 3**

**DESIGN**

**3.1 INTRODUCTION TO DESIGN**

Design is the first step in the development phase for any techniques and principles for the purpos3.e of defining a device, a process or system in sufficient detail to permit its physical realization. Once the software requirements have been analyzed and specified the software design involves three technical activities-design, coding, implementation and testing that are required to build and verify the software.

**3.2 Methodology Development Model**

The sequential phases in Waterfall model are –

* **Requirement Gathering and analysis** − All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
* **System Design** − The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
* **Implementation** − With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
* **Integration and Testing** − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
* **Deployment of system** − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
* **Maintenance** − There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

**3.2 DATA FLOW DIAGRAMS**

The DFD takes an input-process-output view of a system i.e., data objects flow into the software, are transformed by processing elements, and resultant data objects flow out of the software. Data objects represented by labelled arrows and transformation are represented by circles also called as bubbles. DFD is presented in hierarchical fashion i.e., the first data flow model represents the system as a whole.

A context-level DFD for the system the primary external entities produce information for use by the system and consume information generated by the system. The labelled arrow represents data objects or object hierarchy. The DFD enables the software engineer to develop models of the information domain and functional domain at the same time. As the DFD is refined into grater levels of details, the analyst performs an implicit functional decomposition of the system. At the same time, DFD refinement results in a corresponding refinement of the data as it moves through the process that embody the applications.

**RULES OF DFD:**

* Fix the scope of the system by means of context diagrams.
* Organize the DFD so that the main sequence of the actions.
* Reads left to right and top to bottom.
* Identify all inputs and outputs.
* Identify and label each process internal to the system with rounded circles.

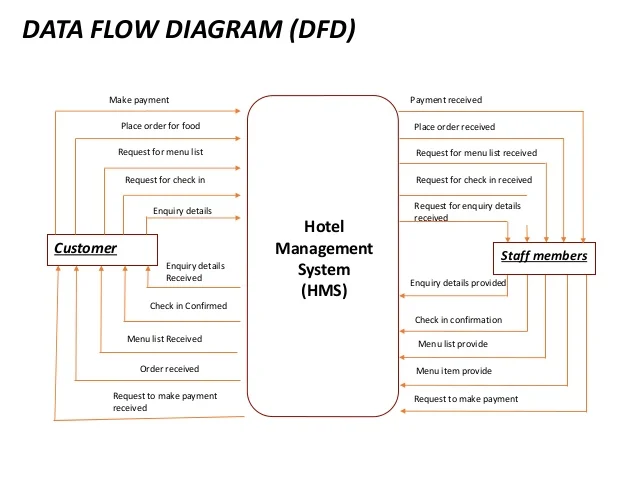
****

Figure.3.1: Dataflow/Context diagram of Hotel Management System.

* The dataflow or the context diagram contains the information on schedules and also contains a database of reservations (passengers records) and of ticket issued.

**3.3 E-R DIAGRAMS**

The entity relationship(ER) model was originally was proposed by Peter in 1976 [Chen76] as was to unify the network and relational database views. Simply stated the ER model is a conceptual data model that views the real world as entities and relationships. An Entity Relationship Model (ERM), in software engineering is an abstract and conceptual representation of data.

**Connectivity and Cardinality:**

The basic types of connectivity for relations are:

• One-to-one

• One-to-many

• Many-to-many

A **one-to-one** (1:1) relationship is when at most one instance of an entity A is associated with one instance of entity B. For example, “employees in the company are each assigned their own office. For each employee there exists a unique office and for each office their exists a unique employee.”

The implementation of this one-to-one relationship in airline reservation system is:

* The Passengers looks for the Room\_ details.
* The Passengers makes the Payment.

A **one-to-many** (1:N) relationships is when for one instance of entity A, there are zero, one, or many instances of entity B, but for one instance of entity B, there is only one instance of entity A. For example,” a department has many employees, each employee is assigned to one department.”

The implementation of this one-to-many relationship in airline reservation system is:

* The Admin checks for the Frequent\_ Customer\_ Details.
* The Admin edits the Room\_ Details.
* The Admin views Customers.

A **many-to-many** (M:N) relationship, sometimes called non-specific, is when for on instance of entity A, there are zero, one, or many instance of entity B and for one instance of entity B there are zero, one, or many instance of entity. The connectivity of a relationship describes the mapping of associated.

* M no of customers can view N no of room details.

**COMPONENTS OF ER MODEL**:

This model is based on 3 concepts:

* Entity is the thing which we want to store the information. It is an elementary basic building block of storing information. About business process. An entity represents and object defined within the information system about which you want to store information. Entities are distinct things in the enterprise. Entities are represented by labeled rectangles.

* A relationship is a named collection or association between entities or used to relate two or more entities attributes or meaningful interaction between the objects.
* Attributes are the properties of the entities and relationship, descriptor of the entity. Attributes are elementary pieces of information attached to an entity

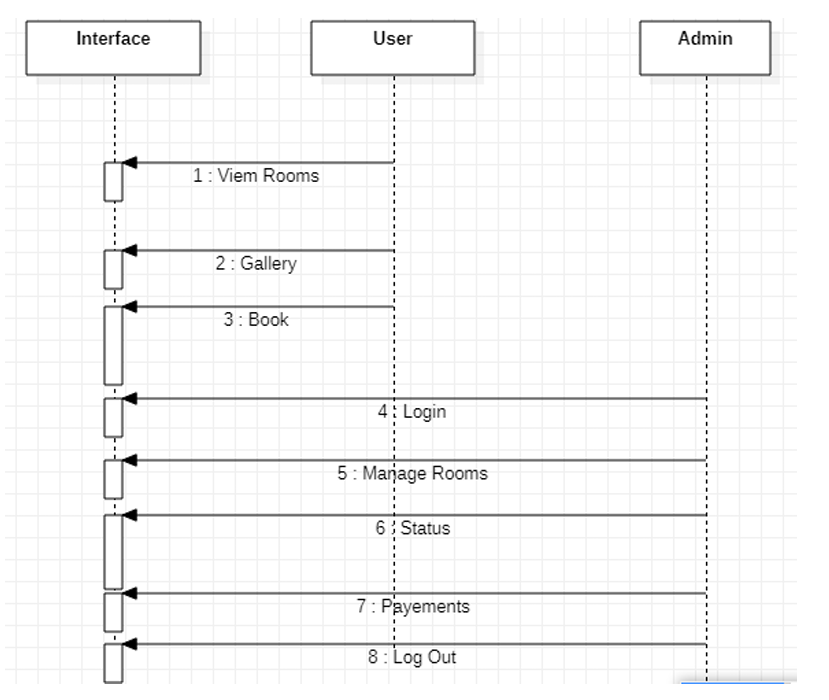


Figure 3.2 Sequence Diagram

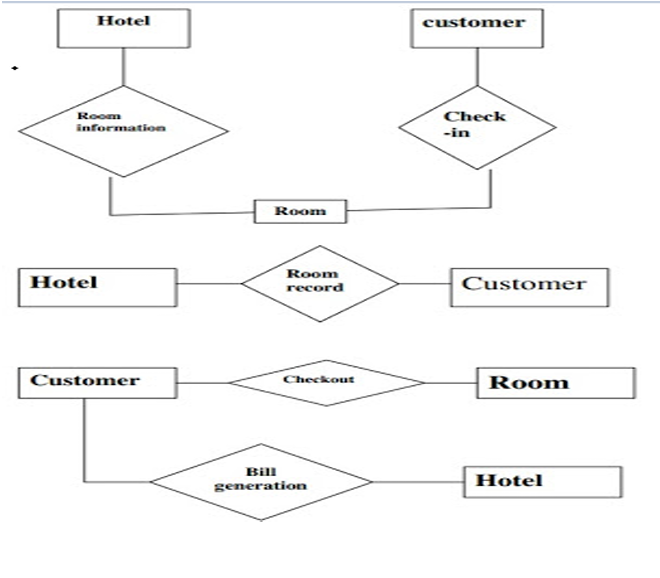


Figure 3.3 Flowchart

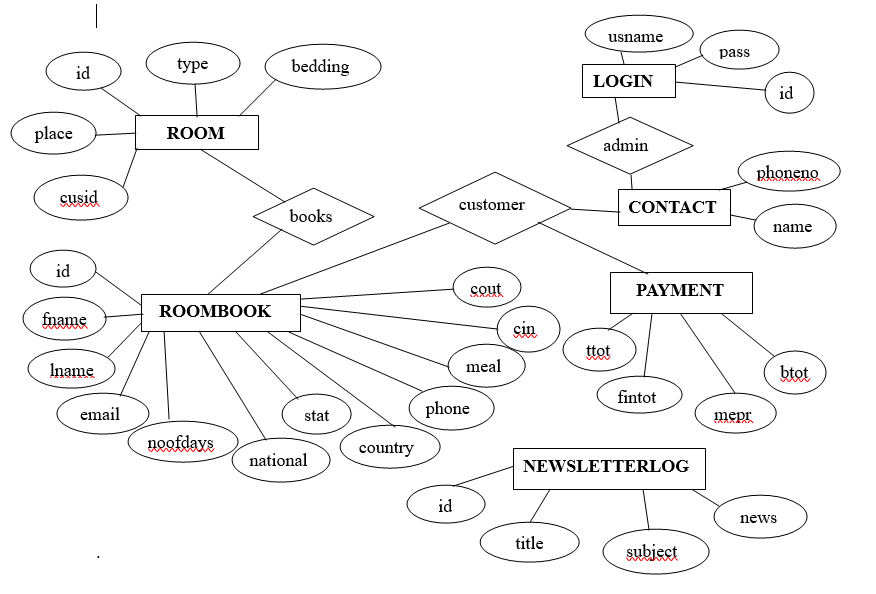


Figure 3.3 E-R Diagram

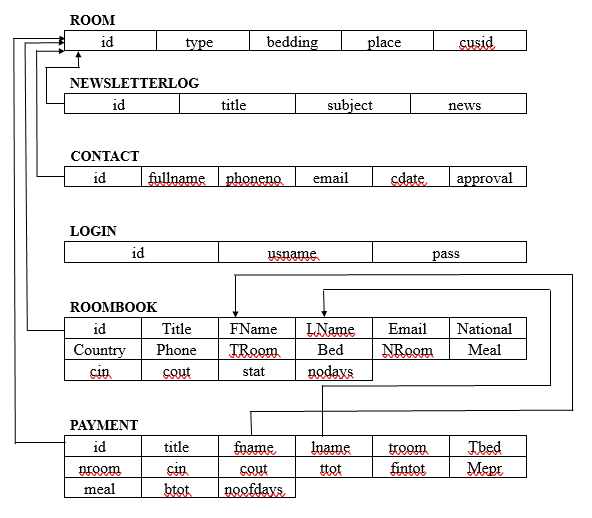


Figure 3.4 Schema Diagram

**DESCRIPTION**:

A schema diagram is a diagram which contains entities and the attributes that will define that schema. A schema diagram only shows us the database design. It does not show the actual data of the database. Schema can be a single table or it can have more than one table which is related.

**Chapter 4**

**PROJECT MODULES**

**4.1 Modules**

* Admin panel
* User panel

**Admin Panel:**

* Status
* View Booking
* View Rooms
* View payment’s
* News Letter
* Logout

**User Panel**

* View rooms
* View Gallery
* View services
* Book

Status

you can check the status (empty/filled) of rooms.

View Booking

Employees can conform the reservation by customers.

View Rooms

Check availability of rooms.

View Payment’s

Generate Invoice for rooms.

View Rooms

User can check the availability of rooms.

View Galary

Shows all the amenities and the images of the hotel.

View Service

Shows all the services provided.

Book Rooms

Reservations of room can be done here.

**4.2 Database Table**

1. This is Hotel room table

|  |  |  |
| --- | --- | --- |
| **#** | **Name** | **Type** |
| 1 | id | Int(10) |
| 2 | type | Varchar(15) |
| 3 | bedding | Varchar(10) |
| 4 | place | Varchar(10) |
| 5 | cusid | Int(11) |

1. This is Hotel newsletter log table

|  |  |  |
| --- | --- | --- |
| **#** | **Name** | **Type** |
| 1 | id | Int(10) |
| 2 | title | Varchar(52) |
| 3 | subject | Varchar(100) |
| 4 | news | text |

1. This is Hotel room book table

|  |  |  |
| --- | --- | --- |
| **#** | **Name** | **Type** |
| 1 | id | Int(10) |
| 2 | title | Varchar(5) |
| 3 | FName | text |
| 4 | LName | text |
| 5 | Email | Varchar(50) |
| 6 | National | Varchar(30) |
| 7 | Country | Varchar(30) |
| 8 | Phone | text |
| 9 | TRoom | Varchar(20) |
| 10 | Bed | Varchar(10) |
| 11 | NRoom | Varchar(2) |
| 12 | Meal | Varchar(15) |
| 13 | Cin | Date |
| 14 | Cout | Date |
| 15 | Stat | Varchar(15) |
| 16 | nodays | Int(11) |

1. This is Hotel Contact table

|  |  |  |
| --- | --- | --- |
| **#** | **Name** | **Type** |
| 1 | id | Int(10) |
| 2 | fullname | Varchar(100) |
| 3 | phoneno | Int(10) |
| 4 | email | text |
| 5 | cdate | date |
| 6 | approval | Varchar(12) |

1. This is Hotel login table

|  |  |  |
| --- | --- | --- |
| **#** | **Name** | **Type** |
| 1 | id | Int(10) |
| 2 | usname | Varchar(30) |
| 3 | pass | Varchar(30) |

The objective of the system analysis activity is to develop structured system specification for the proposed system. The structured system specification should describe what the proposed system would do; independent of the technology, which will be used to implement these requirements. The structured system specification will be used to implement these requirements.

**Chapter 5**

**IMPLEMENTATION**

Implementation is the stage where the theoretical design is turned into a working system. The most crucial stage in achieving a new successful system and in giving confidence on the new system for the users that it will work efficiently and effectively. The system can be implemented only after thorough testing is done and if is found to work according to the specification.

It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the changeover and an evaluation of change over methods a part from planning. Two major tasks of preparing the implementation are education and training of the users and testing of the system.

The more complex the system being implemented, the more involved will be the system analysis and design effort required just for implementation. This activity includes programming, testing and integration of modules into a progressively more complete system. Implementation is the process of collect all the required parts and assembles them into a major product.

Hotel Management System

* Sequencing information

-Login information should be filled before the user allowed.

* Error Handling

-If user doesn’t filled up validate information then the system display error message for user and request to enter the validate information.

Performance required

* Security

-System should be Protected from unauthorized access Where the validate Username and Password are required so no other can access.

**TABLE CREATION:**

CREATE TABLE IF NOT EXISTS `contact` (

`id` int(10) unsigned NOT NULL,

`fullname` varchar(100) DEFAULT NULL,

`phoneno` int(10) DEFAULT NULL,

`email` text,

`cdate` date DEFAULT NULL,

`approval` varchar(12) DEFAULT NULL

) ;

CREATE TABLE IF NOT EXISTS `login` (

`id` int(10) unsigned NOT NULL,

`usname` varchar(30) DEFAULT NULL,

`pass` varchar(30) DEFAULT NULL

);

CREATE TABLE IF NOT EXISTS `newsletterlog` (

`id` int(10) unsigned NOT NULL,

`title` varchar(52) DEFAULT NULL,

`subject` varchar(100) DEFAULT NULL,

`news` text

);

CREATE TABLE IF NOT EXISTS `payment` (

`id` int(11) DEFAULT NULL,

`title` varchar(5) DEFAULT NULL,

`fname` varchar(30) DEFAULT NULL,

`lname` varchar(30) DEFAULT NULL,

`troom` varchar(30) DEFAULT NULL,

`tbed` varchar(30) DEFAULT NULL,

`nroom` int(11) DEFAULT NULL,

`cin` date DEFAULT NULL,

`cout` date DEFAULT NULL,

`ttot` double(8,2) DEFAULT NULL,

`fintot` double(8,2) DEFAULT NULL,

`mepr` double(8,2) DEFAULT NULL,

`meal` varchar(30) DEFAULT NULL,

`btot` double(8,2) DEFAULT NULL,

`noofdays` int(11) DEFAULT NULL

);

CREATE TABLE IF NOT EXISTS `room` (

`id` int(10) unsigned NOT NULL,

`type` varchar(15) DEFAULT NULL,

`bedding` varchar(10) DEFAULT NULL,

`place` varchar(10) DEFAULT NULL,

`cusid` int(11) DEFAULT NULL

);

CREATE TABLE IF NOT EXISTS `roombook` (

`id` int(10) unsigned NOT NULL,

`Title` varchar(5) DEFAULT NULL,

`FName` text,

`LName` text,

`Email` varchar(50) DEFAULT NULL,

`National` varchar(30) DEFAULT NULL,

`Country` varchar(30) DEFAULT NULL,

`Phone` text,

`TRoom` varchar(20) DEFAULT NULL,

`Bed` varchar(10) DEFAULT NULL,

`NRoom` varchar(2) DEFAULT NULL,

`Meal` varchar(15) DEFAULT NULL,

`cin` date DEFAULT NULL,

`cout` date DEFAULT NULL,

`stat` varchar(15) DEFAULT NULL,

`nodays` int(11) DEFAULT NULL

);

**ALTERATION OF TABLE:**

ALTER TABLE `contact`

ADD PRIMARY KEY (`id`);

ALTER TABLE `login`

ADD PRIMARY KEY (`id`);

ALTER TABLE `newsletterlog`

ADD PRIMARY KEY (`id`);

ALTER TABLE `room`

ADD PRIMARY KEY (`id`);

ALTER TABLE `roombook`

ADD PRIMARY KEY (`id`);

ALTER TABLE `contact`

MODIFY `id` int(10) unsigned NOT NULL AUTO\_INCREMENT;

ALTER TABLE `login`

MODIFY `id` int(10) unsigned NOT NULL AUTO\_INCREMENT,AUTO\_INCREMENT=3;

ALTER TABLE `newsletterlog`

MODIFY `id` int(10) unsigned NOT NULL AUTO\_INCREMENT;

ALTER TABLE `room`

MODIFY `id` int(10) unsigned NULL AUTO\_INCREMENT,AUTO\_INCREMENT=16;

ALTER TABLE `roombook`

MODIFY `id` int(10) unsigned NOT NULL AUTO\_INCREMENT,AUTO\_INCREMENT=2;

**Database Connection:**

User page:

< ?php

$con = mysqli\_connect("localhost","root","","hotel") or die(mysql\_error());

?>

Admin page:

<?php

$con = mysqli\_connect("localhost","root","","hotel") or die(mysql\_error());

?>

Admin Page (Index Page):

<?php

session\_start();

if(isset($\_SESSION["user"]))

{

header("location:home.php");

}

?>

<!DOCTYPE html>

<html >

<head>

<meta charset="UTF-8">

<title>THE WESTONN ADMIN</title>

<link rel="stylesheet" href="css/style.css">

</head>

<body>

<div id="clouds">

<div class="cloud x1"></div>

<!-- Time for multiple clouds to dance around -->

<div class="cloud x2"></div>

<div class="cloud x3"></div>

<div class="cloud x4"></div>

<div class="cloud x5"></div>

</div>

<div class="container">

<div id="login">

<form method="post">

<fieldset class="clearfix">

<p><span class="fontawesome-user"></span><input type="text" name="user" value="Username" onBlur="if(this.value == '') this.value = 'Username'" onFocus="if(this.value == 'Username') this.value = ''" required></p> <!-- JS because of IE support; better: placeholder="Username" -->

<p><span class="fontawesome-lock"></span><input type="password" name="pass" value="Password" onBlur="if(this.value == '') this.value = 'Password'" onFocus="if(this.value == 'Password') this.value = ''" required></p> <!-- JS because of IE support; better: placeholder="Password" -->

</body>

</html>

<?php

include('db.php');

if($\_SERVER["REQUEST\_METHOD"] == "POST") {

// username and password sent from form

$myusername = mysqli\_real\_escape\_string($con,$\_POST['user']);

$mypassword = mysqli\_real\_escape\_string($con,$\_POST['pass']);

$sql = "SELECT id FROM login WHERE usname = '$myusername' and pass = '$mypassword'";

$result = mysqli\_query($con,$sql);

$row = mysqli\_fetch\_array($result,MYSQLI\_ASSOC);

$active = $row['active'];

$count = mysqli\_num\_rows($result);

// If result matched $myusername and $mypassword, table row must be 1 row

if($count == 1) {

$\_SESSION['user'] = $myusername;

header("location: home.php");

}else {

echo '<script>alert("Your Login Name or Password is invalid") </script>' ;

}

}

?>

**QUERIES USED:**

Some of the queries used in our project:

For Selection:

* $query = "SELECT \* FROM payment";
* $sql ="select \* from payment where id = '$pid' ";

For Insertion:

* $newUser="INSERT INTO `roombook`(`Title`, `FName`, `LName`, `Email`, `National`, `Country`, `Phone`, `TRoom`, `Bed`, `NRoom`, `Meal`, `cin`, `cout`,`stat`,`nodays`) VALUES ('$\_POST[title]','$\_POST[fname]','$\_POST[lname]','$\_POST[email]','$\_POST[nation]','$\_POST[country]','$\_POST[phone]','$\_POST[troom]','$\_POST[bed]','$\_POST[nroom]','$\_POST[meal]','$\_POST[cin]','$\_POST[cout]','$new',datediff('$\_POST[cout]','$\_POST[cin]'))";
* $sql = "INSERT INTO `contact`(`fullname`, `phoneno`, `email`,`cdate`,`approval`) VALUES ('$name','$phone','$email',now(),'$approval')" ;

For Deletion:

* $delsql= "DELETE FROM `roombook` WHERE id='$id' ";
* $sql ="DELETE FROM `room` WHERE id = '$did'" ;

For Update:

* $rpsql = "UPDATE `room` SET `place`='$free',`cusid`='$nul' where `cusid`='$id'";
* $rpsql = "UPDATE `room` SET `place`='$notfree',`cusid`='$id' where bedding ='$bed' and type='$troom' ";
* $upsql = "UPDATE `login` SET `usname`='$usname',`pass`='$passwr' WHERE id = '$id'";

**CHAPTER 6**

**TESTING**

Testing is a process of executing a program with the intent of finding an error. Testing is a crucial element of software quality assurance and presents ultimate review of specification, design and coding. System testing is an important phase. Testing represents an interesting anomaly for software. Thus a series of testing are performed for the proposed system before the system is ready for user acceptance testing.

Testing objectives:

* Testing is a process of executing a program with the intent of finding an error.
* A good test case is one that that has a high probability of finding an as yet undiscovered error.
* A successful test is one that uncovers an as undiscovered error.

Testing Principles:

* All test should be traceable to end user requirements.
* Test should be planned long before testing begins.
* Testing should begin on a small scale and progress towards testing is large.
* Exhaustive testing is not possible.
* To be most effective testing should be conducted by a independent third party.

**Test Generation**

This activity generates a set of test data, which can be used to test the new system before accepting it. In the test generation phase all the parts are come which are to be tested to ensure that system does not produce any error.

**Two different techniques to accomplish these objectives:-**

**a) White- box testing**

Focuses on the program structure. Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions.

**b) Black-box testing**

Designed to validate functional requirements without regard to the internal workings of a program. Mainly focuses on the information domain of the software, deriving test cases

by partitioning input and output in a manner that provides through test coverage.

Testing strategies:

A strategy for software testing must accommodate low-level tests that are necessary to

verify that all small source code segment has been correctly implemented as well as high-

level tests that validate major system functions against customer requirements.

Testing fundamentals:

Testing is a process of executing program with the intent of finding error. A good test

case is one that has a high probability of finding an as undiscovered error ..

Testing information flow:

Information flow for testing flows the pattern. Two classes of input provided to test the process. The software configuration includes a software requirement, a design

specification and source code.

**Unit testing**:

Unit testing is essential for the verification of the code produced during the coding phase and hence the goal is to test the internal logic of the modules. Using the detailed design description as a guide, important paths are tested to uncover errors with in the boundary of the modules. All units of Vienna SQL were successfully tested.

Integration testing:

Integration testing focuses on unit tested modules and build the program structure that is dictated by the design phase.

System testing:

System testing tests the integration of each module in the system. It also tests to find discrepancies between the system and its original objective, current specification and system documentation. The primary concern is the compatibility of individual modules.

Acceptance testing:

This testing is done to verify the readiness of the system for the implementation.

Acceptance testing begins when the system is complete. Its purpose is to provide the end user with the confidence that the system is ready for use. It involves planning and execution of functional tests, performance tests and stress tests in order to demonstrate that the implemented system satisfies its requirements.

**Uses of testing:**

* Verify that there is a portal to add new flights in the system.
* Verify that on filling flight details like flight name, code, from and to destinations, capacity, timings, and frequency etc, new flights get successfully added in the system.
* Verify that users can search for flights by name, from-to airports or flight code for checking their status and timings.

**6.1 Test cases:**

Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been execute. Using white- box testing methods, the software engineer can drive test

Cases that.

* Guarantee that logical decisions on their true and false sides.
* Exercise all logical decisions on their true and false sides.
* Execute all loops at their boundaries and within their operational bounds.

The test case specification for system testing has to be submitted for review before system testing commences.

**CHAPTER 7**

**RESULT ANALYSIS**

**USER PANEL:**

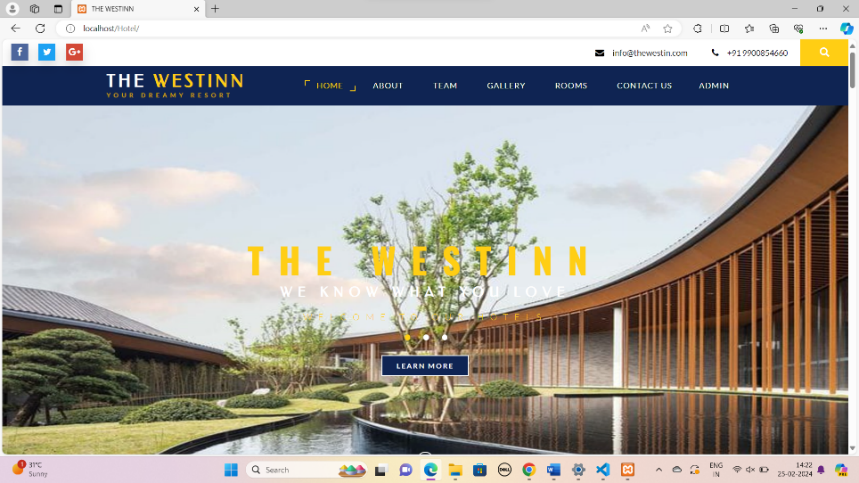
****

Figure 7.1 Home page

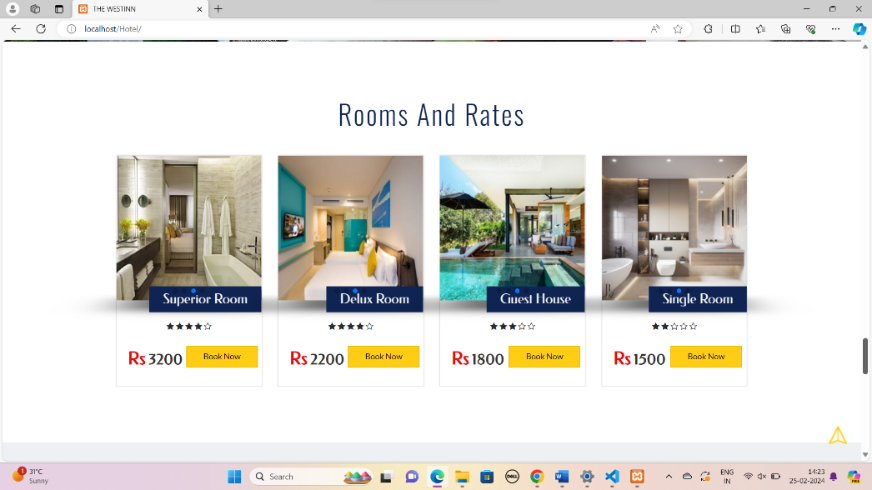
****

Figure 7.2 Room details Page

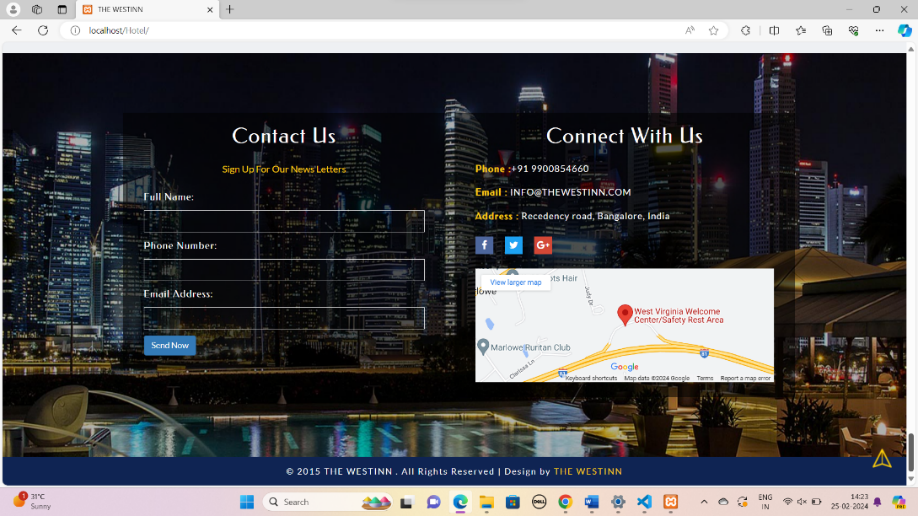
****

Figure 7.3 Contact us Page

**ADMIN PANEL:**

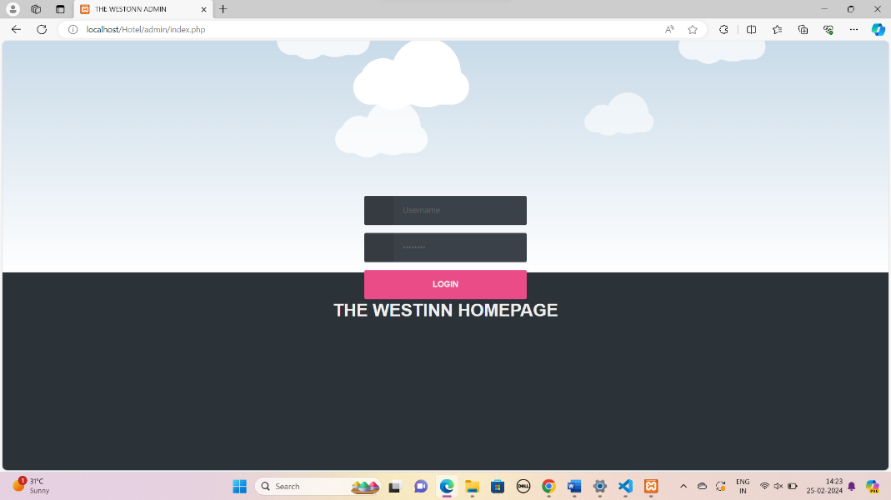
****

Figure 7.4 Admin Login Page

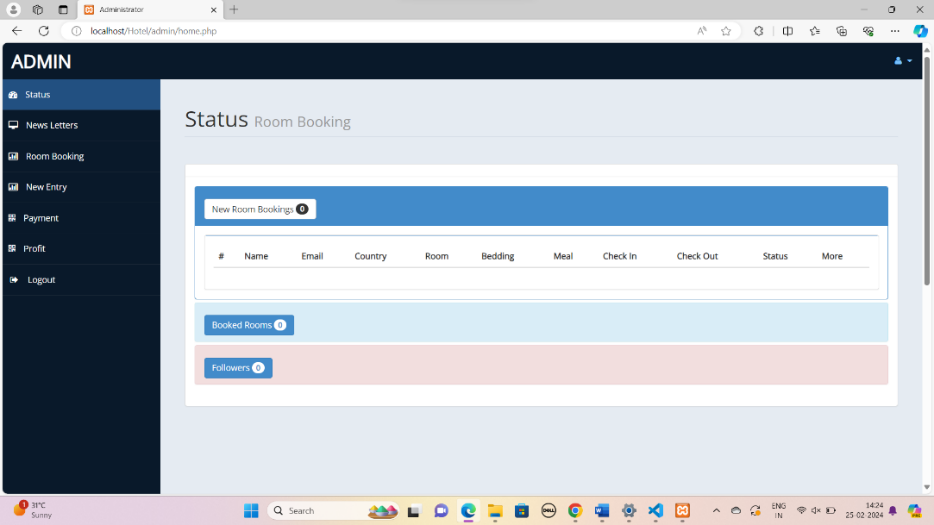
****

Figure 7.5 Room Status Page

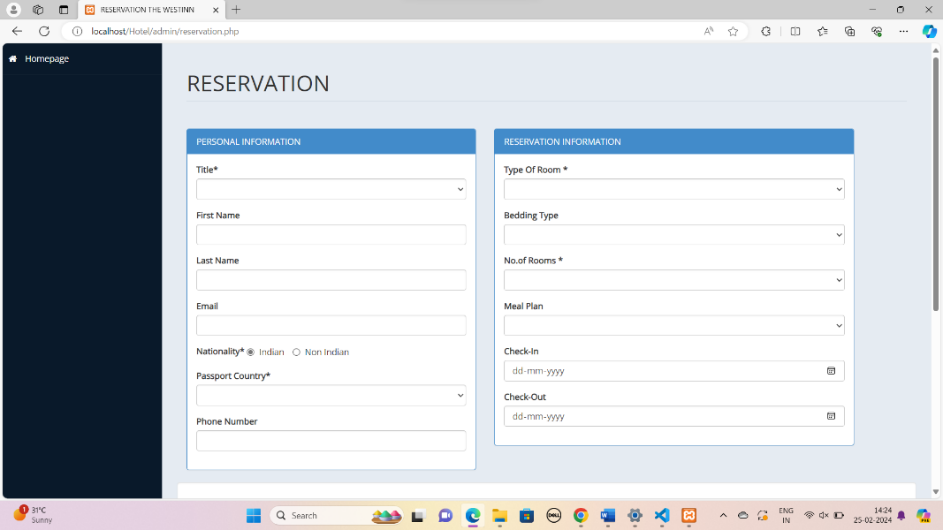
****

Figure 7.6 Reservation Page

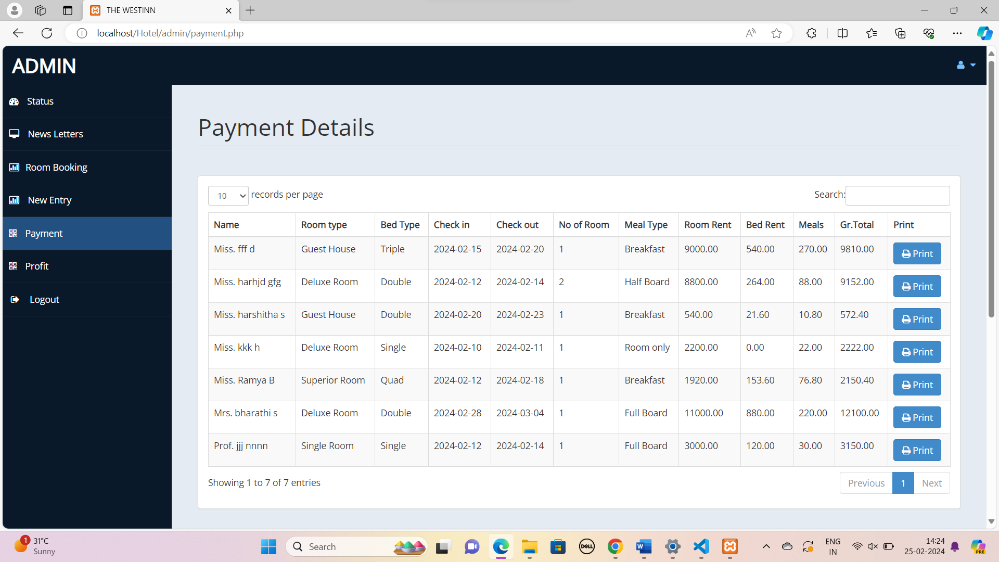
****

Figure 7.7 Payment Details Page

**BACKEND SIDE:**

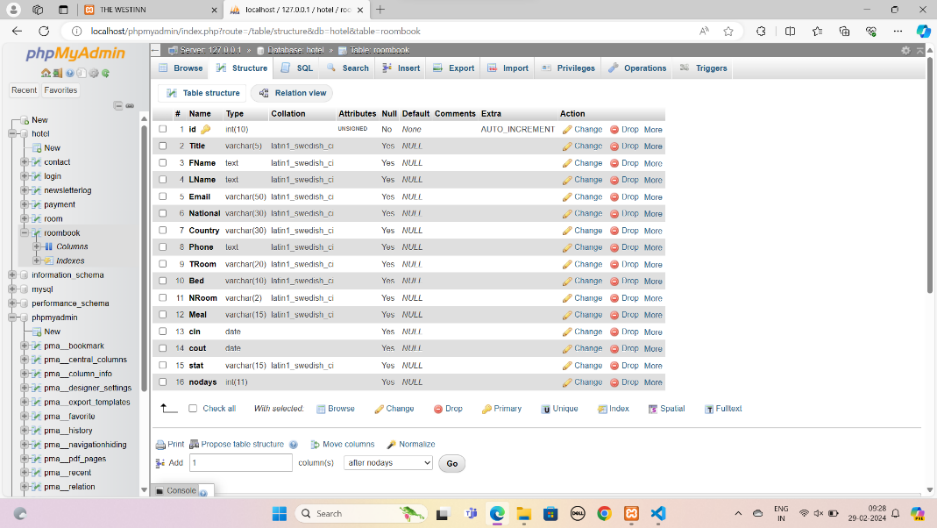
****

Figure 7.8 Reservation Table Backend Side

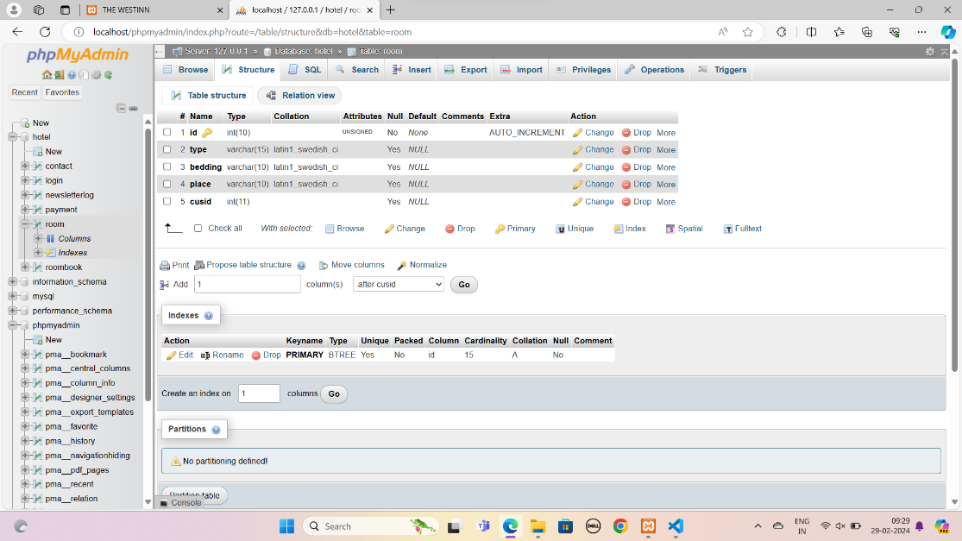
****

Figure 7.9 Newsletter Table Backend Side

**CONCLUSION**

* To conclude the description about this project, the project is developed using PHP and MYSQL and is based on the requirement specification of the analysis of the existing system, with flexibility for future enhancement.
* The expanded functionality of today’s software requires an appropriate approach towards software development. This airline ticket reservation system is designed for people who want to book tickets to fly to different places.
* Identification of the drawbacks of the existing system leads to the designing of computerized system, which is compatible to the existing system. The computerized system is more user friendly and GUI oriented. Thus, automation of the entire system improves the efficiency. the system has adequate scope for modification in future if it is necessary.
* This system will help administration to work easily. Because of its easy access and less time consuming administration can get the information of the Users, Rooms, Payments, etc. They do not have to search in the paper file for the log time. Members an easily handle the system.
* Acceptance testing begins when the system is complete. Its purpose is to provide the end user with the confidence that the system is ready for use
* It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the changeover and an evaluation of change over methods a part from planning.

**FUTURE ENHANCEMENT**

In future, the Hotel management system can extend its features by:

* Adding scope to the online email confirmation features.
* Mobile SMS notification.
* Online payment facility.
* Adding rooms availability information.
* Adding checkout button in admin pannel.
* Adding service specific payment.
* Adding employee details attending the customer.

**BIBLIOGRAPHY**

**WEBSITE:**

[https://www.w3schools.com](https://www.w3schools.com/)

<https://stackoverflow.com>

<https://www.tutorialspoint.com>

<https://www.phptpoint.com>

<https://www.google.co.in/>

<https://www.gist.github.com>

<https://www.datainflow.com>

<https://www.codeproject.co>

<https://www.wikipedia.org/>

<https://www.youtube.com>