

JNAN VIKAS MANDAL’s

JVM’s Mehta Degree College

MOHANLAL RAICHAND MEHTA DEGREE COLLEGE OF COMMERCE

DIWALIMAA DEGREE COLLEGE OF SCIENCE

AMRITLAL RAICHAND MEHTA DEGREE COLLEGE OF ARTS SCIENCE

PADHMSHREE Dr. R.T.DOSHI DEGREE COLLEGE OF COMPUTER SCIENCE

CERTIFICATE

This is to certify that **Mr. Milind Baban Patil** ,Student of Third Year Bachelor of Science (Computer Science) of JVM’s Mehta Degree College has successfully completed the Project work titled “ **Hotel SUNRISE Website**” in partial fulfillment in this educational year of University of Mumbai.

This Project is the record of authentic work carried out during the academic year 2019-20. It is further certified that he has completed all required phases of the Project.

Date of Submission: \_\_\_\_\_\_\_\_\_\_ Exam Seat No: \_\_\_\_\_\_\_\_\_\_\_

Mrs. Sarita Sarang Mrs. Janhavi Kshirsagar

(Internal Guide) (Head of Department)

External Examiner

## INDEX

|  |  |
| --- | --- |
| SR. NO. | TOPIC |
| 1. | Introduction |
| 2. | SPECIFICATION REQUIREMENT |
| 3. | TECHNOLOGIES USED |
| 4. | BEHAVIORAL DESCRIPTION |
| 5. | SYSTEM DESIGN |
| 6. | TESTING AND IMPLEMENTATION |
| 7. | OUT-PUT SCREENS |
| 8. | CODE |
| 9. | Conclusion |

# 1. INTRODUCTION

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

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 – Administrator 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.
* Bookings – All booking and reservations maid on the site are displayed with all booking details: arrival date, departure date, hotel name, room type, number of passengers, price.
* Available rooms – Administrator can also search for room availability from the administrators panel and does not have to go on the site Reports.
* Booking Statistics – Administrator can view statistics of booking on bar charts that show the difference in bookings according to months.
* Mail Templates – Administrators can edit the text on each Mail.

.

## 1.1 Purpose

This proposes that efficiency of hotel organizations could be improved by integrating service-oriented operations service-oriented operations with project management principles. Such integration would instill innovation, proactive attitudes and regulated risk-taking needed to pursue ongoing improvement and proactive response to change. By managing each change as a project, embedded in smoothly running operations, hotels would extend their life span by continuously reinventing themselves.

### 1.1.1 Advantages:

* The advantages of bookinga hotel onlineadd 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 at home. The Internet helps you to browse through the hotels around the world and compare the facilities and rates easily.

## 1.2 Scope

### 1.2.1 Existing System with Limitations:

In this step, we provide a detailed description about the existing system and the problems faced in the existing system. This stage there is no existing system previously; we are developing a new system. Till now no system is available with this type of features and facilities. This system is developed for the all types of users with highly flexible and configurable product is envisaged to ensure global marketing

### 1.2.2 Proposed System Features:

* Some of the service providers won't allow you to choose your hotel, they only allow you to select location and quality of the hotel. Considerable discounts on hotels may be available in off-seasons.
* Utilize the service of online hotel booking service providers when you are planning for a trip.
* Each and every customer looks to book their hotel rooms early and conveniently.
* Users can Post, update and delete the links in all categories.
* Online hotel booking is the best way to book rooms in your favourite hotels.
* Planning a vacation has never been easier and more reasonable than now. Easiness, affordable pricing, and simple comparison shopping make online hotel bookings accepted to all.

# 2. SPECIFICATION REQUIREMENT

Requirement analysis for web applications encompasses three major tasks: formulation, requirements gathering and analysis modeling. During formulation, the basic motivation and goals for the web application are identified, and the categories of users are defined. In the requirements gathering phase, the content and functional requirements are listed and interaction scenarios written from end-user’s point-of-view are developed. This intent is to establish a basic understanding of why the web application is built, who will use it, and what problems it will solve for its users.

## 2.1 Software requirement Specification

A set of programs associated with the operation of a computer is called software. Software is the part of the computer system, which enables the user to interact with several physical hardware devices.

The minimum software requirement specifications for developing this project are as follows:

**Operating System : Windows 7/8//8.1/10/XP/Vista/2000, Linux.**

**Presentation layer : PHP, CSS, HTML, JSP, FLASH**

**Database : My SQL**

**Presentation : Powerpoint 2003**

**Documentation Tool : Ms Office**

## 2.2 Hardware Requirement Specification:

The collection of internal electronic circuits and external physical devices used in building a computer is called the Hardware.

The minimum hardware requirement specifications for developing this project are as follows:

**Processor : Standard processor with a speed of 1.6 GHz**

**RAM :** 256 MB RAM or more

**Hard Disk :** 20 GB or more

**Monitor :** Standard color monitor

**Keyboard :** Standard keyboard

**Mouse :** Standard mouse

## 2.3 Functional Requirements

**Module Description**

The HOTEL MANAGEMENT SYSTEM consists of 2 modules and 5 categories.

The categories are as follows:

* + 1. Advanced Search
    2. Booking
    3. Payment
    4. Mail Integration

The modules are

* Administrator

**Admin Module:**

The Functionalities of the Administrator is:

1. The administrator should login into the system with unique his/her password and username.
2. If the username and password is validated then he can gain access to the system.
3. View the hotels can update and delete the hotels.
4. Post the special offers, add new hotel to the hotels list.
5. Insert /View the Availability and Price.
6. Post the links. Adding the links to the hotels and as well view the links.
7. Blocks the user view the status of the users.
8. Restrict the user to the see the more information without login.

**The Administrator can do the following actions:**

1. Login
2. Change Password
3. Admin Functionalities
   1. View Rooms
   2. Add Rooms
   3. Post Special Offers
   4. Edit / Delete Rooms
   5. Upload Images.
4. Logout

# 3. TECHNOLOGIES USED

## 3.1. PHP: Hypertext Preprocessor

PHP is a widely used, general-purpose scripting language that was originally designed for web development to produce dynamic web pages. For this purpose, PHP code is embedded into the HTML source document and interpreted by a web server with a PHP processor module, which generates the web page document.

PHP Advantages

PHP source code is compiled on-the-fly to an internal format that can be executed by the PHP engine. In order to speed up execution time and not have to compile the PHP source code every time the webpage is accessed, PHP scripts can also be deployed in executable format using a PHP compiler.

PHP is one of the most popular server side scripting languages running today. It is used for creating dynamic web pages that interact with the user offering customized information. PHP offers many advantages; it is fast, stable, secure, easy to use and open source (free).

PHP code is inserted directly into the HTML that makes up a website. When a visitor comes to the website, the code is executed. Because PHP is a server side technology, the user does not need any special browser or plug-ins to see the PHP in action.

The beauty of PHP lies in its simplicity. It is easy to understand and learn, especially for those with backgrounds in programming such as C, javascript and HTML. The language is similar to C and Perl so that anyone with a background in either C or Perl programming will feel comfortable using and understanding PHP. PHP also runs on just about every platform including most UNIX, Macs and Windows versions.

Another key advantage of PHP is its connective abilities. PHP uses a modular system of extensions to interface with a variety of libraries such as graphics, XML, encryption, etc. In addition, programmers can extend PHP by writing their own extensions and compiling them into the executable or they can create their own executable and load it using PHP's dynamic loading mechanism.

A huge advantage that PHP offers is its community. Since PHP is an A huge advantage that PHP offers is its community. Since PHP is an open source project, the PHP community is willing to share. If you're looking for a particular script, chances are another user has already created something similar. Check within the PHP community for availability. Likewise, if you have created a function that others might enjoy, be sure to post the code for others.

A PHP scripting block always starts with <?php and ends with ?>. A PHP scripting block can be placed anywhere in the document.

On servers with shorthand support enabled you can start a scripting block with <? and end with ?>.

For maximum compatibility, we recommend that you use the standard form (<?php) rather than the shorthand form.

|  |
| --- |
| <?php ?> |

A PHP file normally contains HTML tags, just like an HTML file, and some PHP scripting code.

Below, we have an example of a simple PHP script which sends the text "Hello World" to the browser:

|  |
| --- |
| <html> <body>  <?php echo "Hello World"; ?>  </body> </html> |

Each code line in PHP must end with a semicolon. The semicolon is a separator and is used to distinguish one set of instructions from another.

There are two basic statements to output text with PHP: echo and print. In the example above we have used the echo statement to output the text "Hello World".

Note: The file must have a .php extension. If the file has a .html extension, the PHP code will not be executed.

Comments in PHP

In PHP, we use // to make a single-line comment or /\* and \*/ to make a large comment block.

|  |
| --- |
| <html> <body>  <?php //This is a comment  /\* This is a comment block \*/ ?>  </body> </html> |

Variables in PHP

Variables are used for storing values, like text strings, numbers or arrays.

When a variable is declared, it can be used over and over again in your script.

All variables in PHP start with a $ sign symbol.

The correct way of declaring a variable in PHP:

|  |
| --- |
| $var\_name = value; |

PHP is a Loosely Typed Language

In PHP, a variable does not need to be declared before adding a value to it.

In the example above, you see that you do not have to tell PHP which data type the variable is.

PHP automatically converts the variable to the correct data type, depending on its value.

In a strongly typed programming language, you have to declare (define) the type and name of the variable before using it.

In PHP, the variable is declared automatically when you use it.

## 3.2 Sessions

A PHP session variable is used to store information about, or change settings for a user session. Session variables hold information about one single user, and are available to all pages in one application.

PHP Session Variables

When you are working with an application, you open it, do some changes and then you close it. This is much like a Session. The computer knows who you are. It knows when you start the application and when you end. But on the internet there is one problem: the web server does not know who you are and what you do because the HTTP address doesn't maintain state.

A PHP session solves this problem by allowing you to store user information on the server for later use (i.e. username, shopping items, etc). However, session information is temporary and will be deleted after the user has left the website. If you need permanent storage you may want to store the data in a database.

Sessions work by creating a unique id (UID) for each visitor and store variables based on this UID. The UID is either stored in a cookie or is propagated in the URL.

Starting a PHP Session

Before you can store user information in your PHP session, you must first start up the session.

Note: The session\_start() function must appear BEFORE the <html> tag:

|  |
| --- |
| <?php session\_start(); ?>  <html> <body>  </body> </html> |

The code above will register the user's session with the server, allow you to start saving user information, and assign a UID for that user's session.

Storing a Session Variable

The correct way to store and retrieve session variables is to use the PHP $\_SESSION variable:

|  |
| --- |
| <?php session\_start(); // store session data $\_SESSION['views']=1; ?>  <html> <body>  <?php //retrieve session data echo "Pageviews=". $\_SESSION['views']; ?>  </body> </html> |

Output:

|  |
| --- |
| Pageviews=1 |

In the example below, we create a simple page-views counter. The isset() function checks if the "views" variable has already been set. If "views" has been set, we can increment our counter. If "views" doesn't exist, we create a "views" variable, and set it to 1:

|  |
| --- |
| <?php session\_start();  if(isset($\_SESSION['views'])) $\_SESSION['views']=$\_SESSION['views']+1; else $\_SESSION['views']=1; echo "Views=". $\_SESSION['views']; ?> |

Destroying a Session

If you wish to delete some session data, you can use the unset() or the session\_destroy() function.

The unset() function is used to free the specified session variable:

|  |
| --- |
| <?php unset($\_SESSION['views']); ?> |

You can also completely destroy the session by calling the session\_destroy() function:

|  |
| --- |
| <?php session\_destroy(); ?> |

Note: session\_destroy() will reset your session and you will lose all your stored session data.

PHP Form Handling

The most important thing to notice when dealing with HTML forms and PHP is that any form element in an HTML page will automatically be available to your PHP scripts.

Example

The example below contains an HTML form with two input fields and a submit button:

|  |
| --- |
| <html> <body>  <form action="welcome.php" method="post"> Name: <input type="text" name="fname" /> Age: <input type="text" name="age" /> <input type="submit" /> </form>  </body> </html>  The PHP $\_GET and $\_POST variables are used to retrieve information from forms, like user input. |

When a user fills out the form above and click on the submit button, the form data is sent to a PHP file, called "welcome.php":

"welcome.php" looks like this:

|  |
| --- |
| <html> <body>  Welcome <?php echo $\_POST["fname"]; ?>!<br /> You are <?php echo $\_POST["age"]; ?> years old.  </body> </html> |

Output could be something like this:

|  |
| --- |
| Welcome John! You are 28 years old. |

The PHP $\_GET and $\_POST functions will be explained in the next chapters.

Form Validation

User input should be validated on the browser whenever possible (by client scripts). Browser validation is faster and reduces the server load.

You should consider server validation if the user input will be inserted into a database. A good way to validate a form on the server is to post the form to itself, instead of jumping to a different page. The user will then get the error messages on the same page as the form. This makes it easier to discover the error.

## 3.3 History of MySQL

We started out with the intention of using the mSQL database system to connect to our tables using our own fast low-level (ISAM) routines.This resulted in a new SQL interface to our database but with almost the same API interface as mSQL.

The following list describes some of the important characteristics of the MySQL Database Software. See also Section 1.6, “MySQL Development Roadmap”, for more information about current and upcoming features.

**Internals and Portability:**

* Written in C and C++.
* Tested with a broad range of different compilers.
* Works on many different platforms.
* Uses GNU Automake, Autoconf, and Libtool for portability.
* APIs for C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, and Tcl are available.

**Data Types:**

* Many data types: signed/unsigned integers 1, 2, 3, 4, and 8 bytes long, FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET, ENUM, and OpenGIS spatial types.
* Fixed-length and variable-length records.

**Statements and Functions:**

* Full operator and function support in the SELECT and WHERE clauses of queries. For example:
* mysql> SELECT CONCAT(first\_name, ' ', last\_name)
* FROM citizen

**Security:** A privilege and password system that is very flexible and secure, and that allows host-based verification. Passwords are secure because all password traffic is encrypted when you connect to a server.

**Scalability and Limits:**

* Handles large databases. We use MySQL Server with databases that contain 50 million records. We also know of users who use MySQL Server with 60,000 tables and about 5,000,000,000 rows.

**Connectivity:** Clients can connect to the MySQL server using TCP/IP sockets on any platform. On Windows systems in the NT family (NT, 2000, XP, or 2003), clients can connect using named pipes. On Unix systems, clients can connect using Unix domain socket files.

* In MySQL 4.1 and higher, Windows servers also support shared-memory connections if started with the --shared-memory option. Clients can connect through shared memory by using the --protocol=memory option.

**Localization:**

* The server can provide error messages to clients in many languages. See Section 5.11.2, “Setting the Error Message Language”.
* Full support for several different character sets, including latin1 (cp1252), german, big5, ujis, and more. For example, the Scandinavian characters ‘å’, ‘ä’ and ‘ö’ are allowed in table and column names. Unicode support is available as of MySQL 4.1.

**Clients and Tools:**

* MySQL Server has built-in support for SQL statements to check, optimize, and repair tables. These statements are available from the command line through the mysqlcheck client. MySQL also includes myisamchk, a very fast command-line utility for performing these operations on MyISAM tables. See Chapter 5, *Database Administration*.
* All MySQL programs can be invoked with the --help or -? options to obtain online assistance.

### 3.3.1. MySQL Stability

This section addresses the questions, “*How stable is MySQL Server?*” and, “*Can I depend on MySQL Server in this project?*”.The information in this section is based on data gathered from the mailing lists, which are very active in identifying problems as well as reporting types of use.

Each release of the MySQL Server has been usable. Problems have occurred only when users try code from the “gray zones.” The descriptions mostly deal with Versions 3.23 and later of MySQL Server.

The MySQL Server design is multi-layered with independent modules. Some of the newer modules are listed here with an indication of how well-tested each of them is:

* Replication (Stable)

Large groups of servers using replication are in production use, with good results work on enhanced replication feature is continuing

* Inno DB tables (Stable)

The inno DB transactional storage engine has been stable since version 3.23.49. Inno DB is being used in large, heavy-load production systems.

* BDB Tables (Stable)

The Berkeley DB code is very stable, but we are still improving the BDB transactional storage engine interface in My SQL Server.

* Full text searches (Stable)

Full-text searching is widely used. Important feature enhancements were added in MySQL 4.0 and 4.1.

* MyODBC 3.51 (Stable)

MyODBC 3.51 uses ODBC SDK 3.51 and is in wide production use. Some issues brought up appear to be application-related and independent of the ODBC driver or underlying database server.

The InnoDB storage engine maintains InnoDB tables within a tablespace that can be created from several files. This allows a table to exceed the maximum individual file size. The tablespace can include raw disk partitions, which allows extremely large tables. The maximum tablespace size is 64TB.

The following table lists some examples of operating system file-size limits. This is only a rough guide and is not intended to be definitive. For the most up-to-date information, be sure to check the documentation specific to your operating system.

# 4. BEHAVIORAL DESCRIPTION

## 4.1 Data Flow:

**There are three types of DFD’s they are**

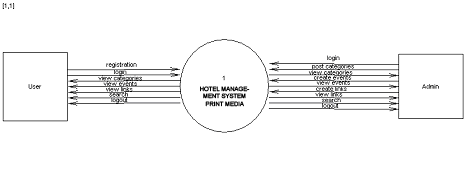
1. **Context Level DFD**
2. **Top Level DFD and**
3. **Detailed Level DFD**

**Context Level DFD:**

In the Context Level the whole system is shown as a single process.

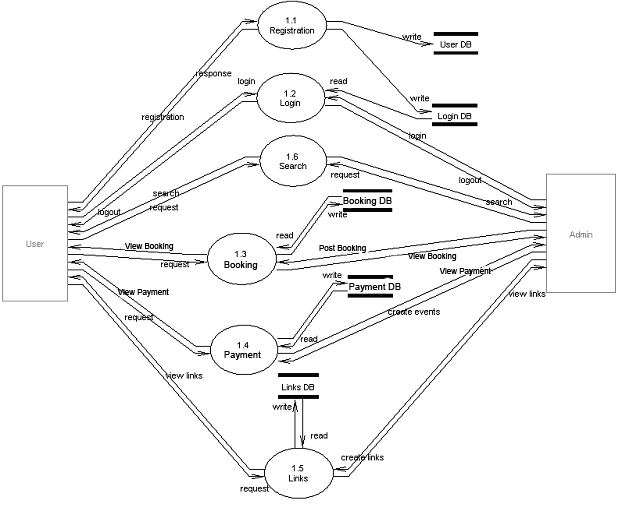
* No data stores are shown.
* Inputs to the overall system are shown together with data sources (as External entities).
* Outputs from the overall system is shown together with their destinations (as External entities).

### 4.1.1 DFD:



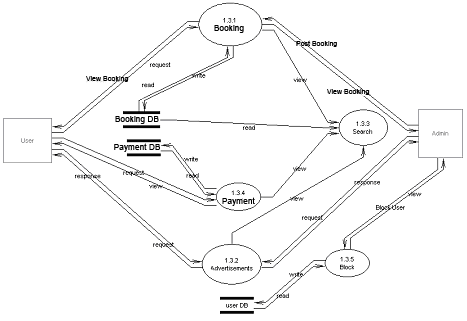
**Top Level DFD:**

The Top Level DFD gives the overview of the whole system identifying the major system processes and data flow. This level focuses on the single process that is drawn in the context diagram by ‘Zooming in’ on its contents and illustrates what it does in more detail.



**Detailed Level DFD:**

In Detailed D.F.Ds the main process is divided into sub processes and we try to find out the flow from one process to another process. We find the interaction among External entities, processes, sub processes and database.



### 4.1.2. Use Case Documentation:

**Use Case Diagram**

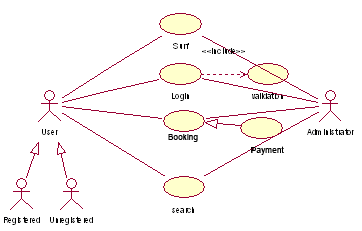
* A use case diagram is a diagram that shows a set of use cases and actors and relationships.

**Contents**

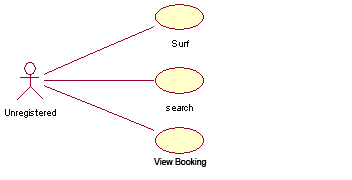
* Use case commonly contain
* Use cases
* Actors

Dependency, generalization and association relationships

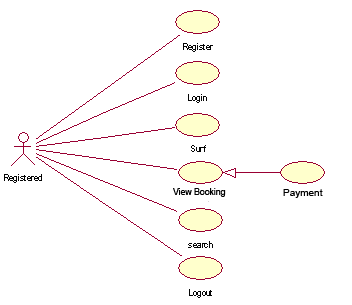
**Over all use case**

****

**Un Registration Use Case**

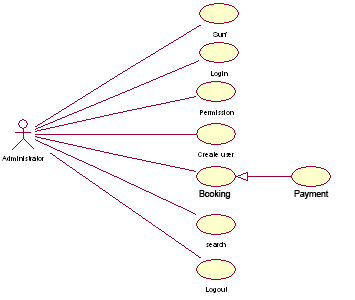
****

**Registration Use Case**

****

**Administrator Use Case**

## 

4.2 Process Flow:

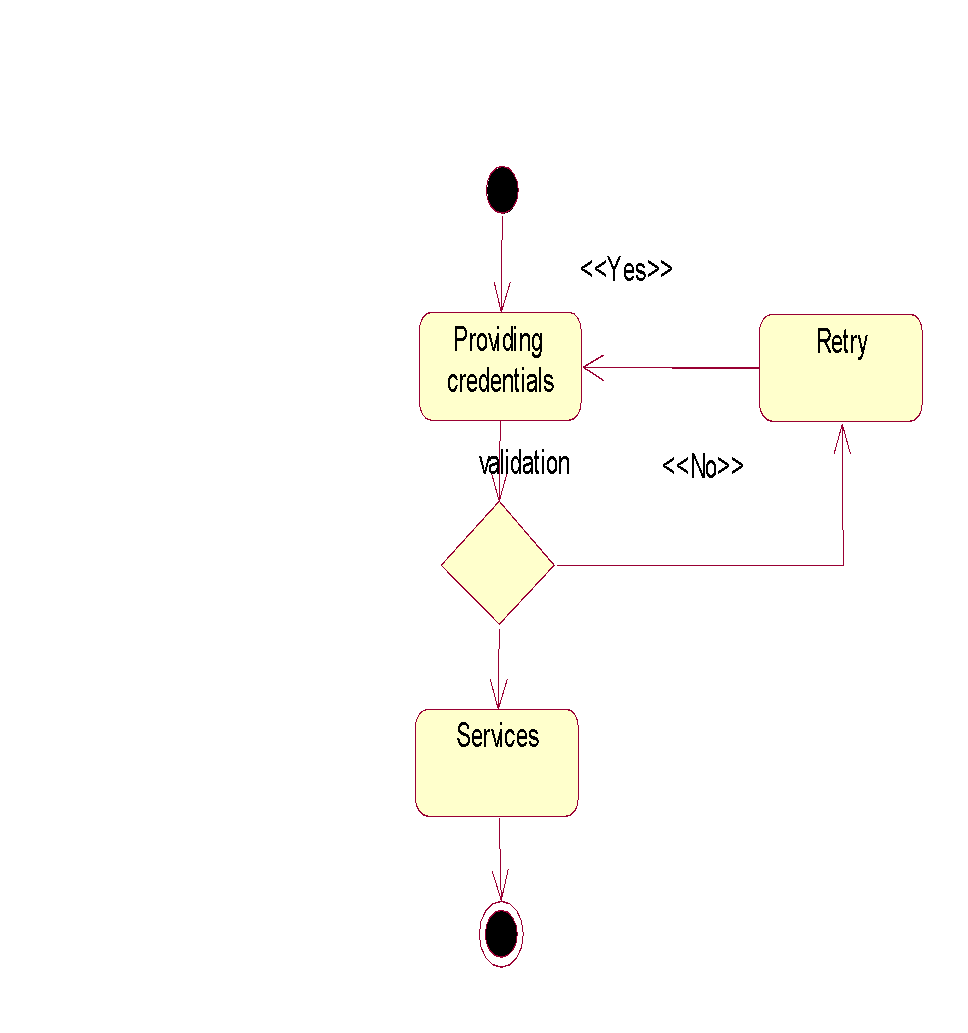
### 4.2.1 Activity Diagrams:

* An activity diagram shows the flow from activity to activity. An activity is an ongoing non- atomic execution within a state machine.
* Activities ultimately result in some action, which is made up of executable atomic computations that result in a change in state of the system or the return of a value.

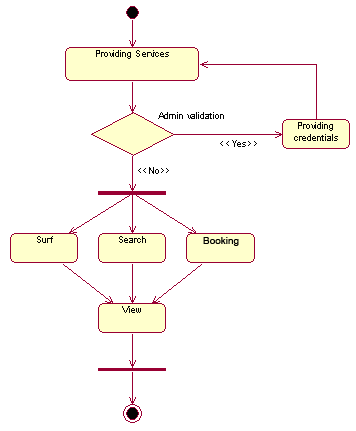
Activity diagrams commonly contain

* Activity states and action states
* Transitions
* Objects
  + Like all other diagrams, activity diagrams may contain notes and constraints.

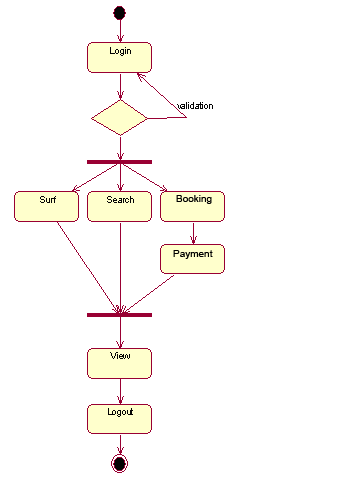
**Login Process**

****

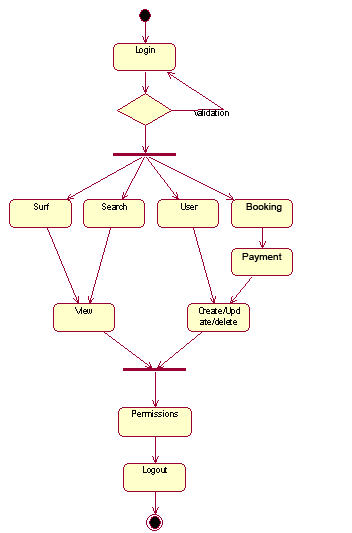
**Un Registration Process**

****

**Registration Process**

****

**Administrator Process**

****

# 5. SYSTEM DESIGN

The main focus of the analysis phase of Software development is on “What needs to be done”. The objects discovered during the analysis can serve as the framework or Design. The class’s attributes, methods and association identified during analysis must be designed for implementation language. New classes must be introduced to store intermediate results during the program execution.

Emphasis shifts from the application domain of implementation and computer such as user interfaces or view layer and access layer. During analysis, we look at the physical entities or business objects in the system, that is, which players and how they cooperate to do the work of the application. These objects represent tangible elements of the business.

During the Design phase, we elevate the model into logical entities, some of which might relate more to the computer domain as people or employees. Here his goal is to design the classes that we need to implement the system. The difference is that, at this level we focus on the view and access classes, such as how to maintain information or the best way to interact with a user or present information.

**Design process:**

During the design phase the classes identified in object-oriented analysis Must be revisited with a shift focus to their implementation. New classes or attributes and Methods must be added for implementation purposes and user interfaces. The object-oriented design process consists of the following activities:

1. Apply design axioms to design classes, their attributes, methods, associations, structure

And protocols Refine and complete the static UML class diagram by adding details to the UML diagram. This step consists of the following activities. \*Refine attributes \*Design methods and protocols by utilizing a UML activity diagram to represent the method’s algorithms.

\*Refine associations between classes

\*Refine class hierarchy and design with inheritance

\*Iterate and refine again

2. Design the access layer

* Create mirror classes: For every business class identified and created. For

      For example, if there are three business classes, create three access layer classes.

* Identify access layer class relationships.
* Simplify classes and their relationships: The main goal here is to eliminate

      redundant classes and structures.

      \*Redundant classes: Do not keep two classes that perform similar translate results

       activities. Simply select one and eliminate the other.

      \*Method classes: Revisit the classes that consist of only one or two methods to       see if they can be eliminated or combined with existing classes.

* Iterate and refine again.

      Define the view layer classes

* + Design the macro level user interface, identifying view layer objects.
  + Design the micro level user interface, which includes these activities:

      \* Design the view layer objects by applying the design axioms and the corollaries.

      \* Built a prototype of the view layer interface.

* Test usability and user satisfaction
* Iterate and refine.

3. Iterate refine the whole design process. From the class diagram, you can begin to extrapolate which classes you will have to build and which existing classes you can reuse. As you do this, also begin this, also begin thinking about the inheritance structure. If you have several classes that seem related but have specific differences.

Design also must be traceable across requirements, analysis, design from the Requirements model.

**DESIGN AXIOMS**

Axioms are a fundamental truth that always is observed to be valid and for which there is no counter example or exception. Such explains that axioms may be hypothesized form a large number of observations by nothing the common phenomena shared by all cases; they cannot be proven or derived, but they can be invalidated by counter examples or exceptions. A theorem is a proposition that may not be self-evident but can be proven from accepted axioms. If therefore, it is equivalent to a law or principle. A corollary is a proposition that follows from an axioms or another proposition that has been proven. Again, corollary is shown to be valid or not valid in the same manner as a theorem. In the two important axioms axiom 1 deals with relationships between system components and axiom 2 deals with the complexity of design.

**The following the two important axioms:**

Axiom 1: The independence axiom, which maintains the independence of the components.

Axiom 2: The information axioms that maintain the information content of the design.

Axioms1 states that, during the design process, as we go from requirement and use case to a system component, each component must satisfy that requirement without affecting other requirements.

An axiom 2 is concerned with simplicity. Scientific theoreticians often rely on a general rule known as Occam’s razor, after William of Occam. He says, “The best theory explains the known facts with a minimum amount of complexity and maximum simplicity and straightforwardness.”

The best designs usually involve the least complex code but not necessarily the fewest number of classes or methods. Minimizing complexity should be the goal, because that produces the most easily maintained and enhanced application. In an object-oriented system, the best way to minimize complexity is to use inheritance and the systems built in classes and to add as little as possible to what already is there.

From the two design axioms, many corollaries may be derived as a direct consequence of the axioms. These corollaries may be more useful in marking specific design decisions, since they can be applied to actual situations.

1. Uncoupled design with less information content: Highly cohesive objects can improve coupling because only a minimal amount of essential information needs to be passed between objects. The degree or strength of coupling between two components is measured by the amount and complexity of information transmitted between them.
2. Single purpose: Each class must have single, clearly defined purposes.
3. Large number of simple classes: Keeping the classes simple allows reusability. Large and complex classes are too specialized to be reused.
4. Strong mapping: There must be a strong association between the physical system and logical design. During the design phase, we need to design this class, design its methods, its association with other objects. So a strong mapping links classes should be identified.
5. Standardization: promote standardization by designing interchangeable and reusing existing classes or components.
6. Design with inheritance: Common behavior must be moved to super classes. The super class-sub class structure must make logical sense.

**Refining attributes and methods:**

Attributes identified in object oriented analyzed must be refined in the design phase. In the analysis phase, the name of the attributes was sufficient. But in the design phase, detailed information must be added to the model. The three basic types of attributes are:

1. Single valued attributes: This has only value or state.
2. Multiplicity or multi valued attributes: This has a collection of many values at any point in time.
3. Instance connection attributes: This is required to provide the mapping needed by an object to fulfill its responsibilities.

**UML attribute presentation:**

Visibility name: type-expression=initial-value

Visibility indicates either public visibility or protected visibility or private visibility. The public visibility indicates that the attribute can be accessible to all classes. The protected visibility indicates that the accessibility is given to the subclasses and operations of the class. The private visibility indicates that the accessibility can be given only to the operations of the class only.

Type expression is a language dependent specification of the implementation type of an attribute. Initial value is a language dependent expression for the initial value is optional.

## 5.1 Sequence and collaboration diagrams

**Sequence Diagram**

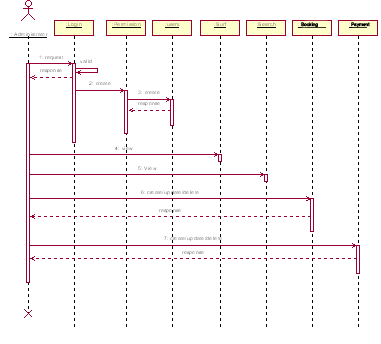
* An interaction diagram shows an interaction, consisting of a set of objects and their relationships, including the messages that may be dispatched among them.
* A sequence diagram is an interaction diagram that emphasizes the time ordering of messages.
* Graphically, a sequence diagram is a table that shows objects arranged along x-axis and messages, ordered in increasing time, along the y-axis.

**Contents**

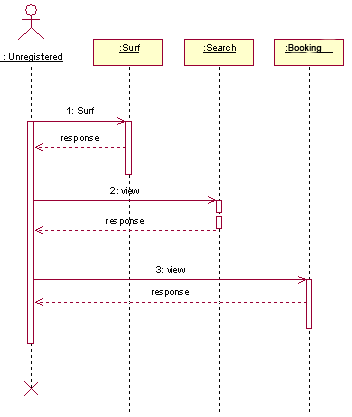
* Sequence diagrams commonly contain the following:
  + - Objects
    - Links
    - Messages

Like all other diagrams, sequence diagrams may contain notes and constraints.

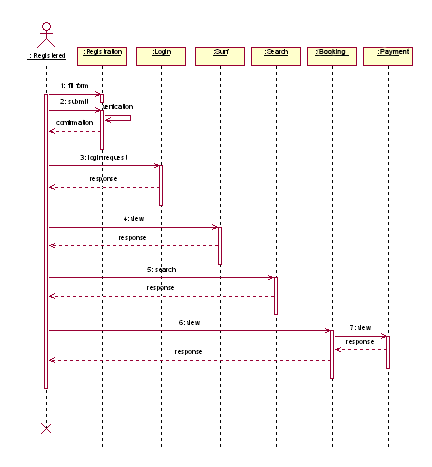
**Administrator Sequence**

****

**Un Registration Sequence**

****

**Registration Sequence**

****

**Collaboration Diagram**

- Collaboration is a society of classes, interfaces, and other elements that work together to provide some cooperative behavior that’s bigger than the sum of all its parts.

- Collaboration is also the specification of how an element, such as a classifier or an operation, is realized by a set of classifiers and associations playing specific roles used in a specific way

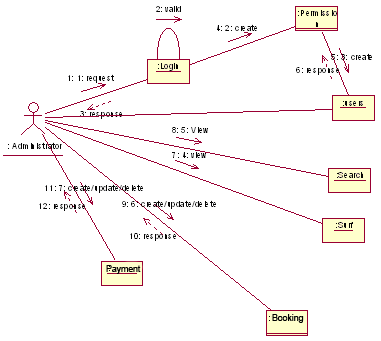
**Contents**

Collaboration diagrams commonly contain the following:

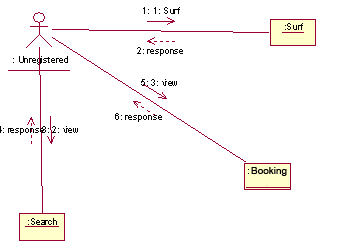
* + - Objects
    - Links
    - Messages

Like all other diagrams, sequence diagrams may contain notes and constraints.

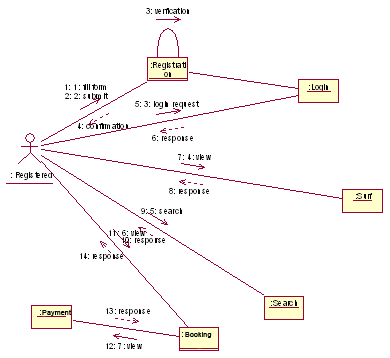
**Administrator Collaboration**

****

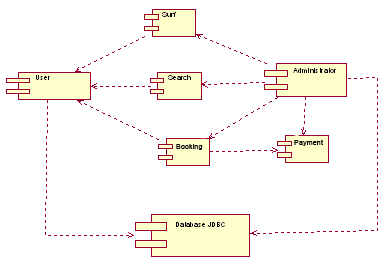
**Un Registration Collaboration**



**Registration Collaboration**



**Component Diagram:**

****

**Deployment Diagram**

- A deployment diagram is a diagram that shows the configuration of run time processing nodes and the components that live on them.

- Graphically, a deployment diagram is a collection of vertices and arcs.

**Contents**

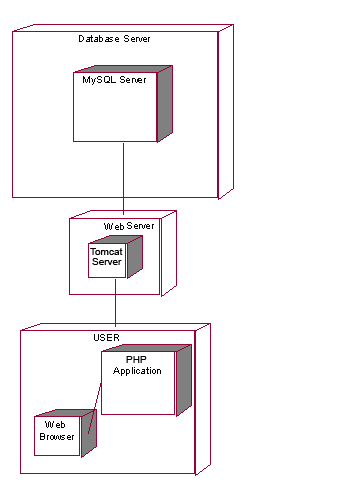
- Deployment diagram commonly contain the following things:

* + - * Nodes
      * Dependency and association relationships

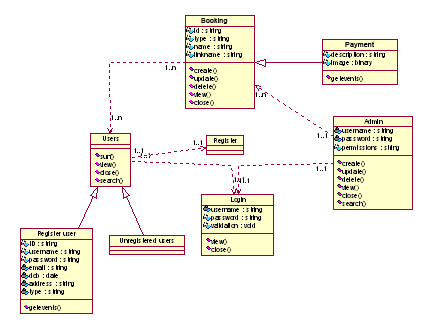
- Like all other diagrams, deployment diagrams may contain notes and constraints.

- Deployment diagrams may also contain components, each of which must live on some node.

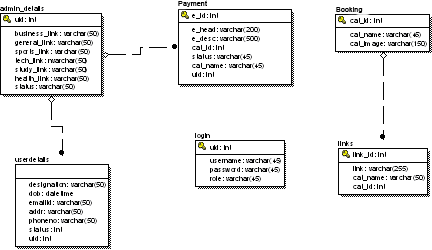
- Deployment diagrams may also contain packages or subsystems, both of which are used to group elements of your model into larger chunks



## 5.2 Class Diagrams



## 5.3 E-R Diagrams



# 6. TESTING AND IMPLEMENTATION

Testing

* **Testing Methodologies**
* Black box Testing:
* White box Testing.
* Gray Box Testing.
* **Levels of Testing**
  + Unit Testing.
  + Module Testing.
  + Integration Testing.
  + System Testing.
  + User Acceptance Testing.
* **Types Of Testing**
  + Smoke Testing.
  + Sanitary Testing.
  + Regression Testing.
  + Re-Testing.
  + Static Testing.
  + Dynamic Testing.
  + Alpha-Testing.
  + Beta-Testing.
  + Monkey Testing.
  + Compatibility Testing.
  + Installation Testing.
  + Adhoc Testing.
  + Ext….
* **TCD (Test Case Documentation)**
* **STLC**
  + Test Planning.
  + Test Development.
  + Test Execution.
  + Result Analysis.
  + Bug-Tracing.
  + Reporting.
* **Microsoft Windows – Standards**
* **Manual Testing**
* **Automation Testing (Tools)**
  + Win Runner.
  + Test Director

**Testing:**

* The process of executing a system with the intent of finding an error.
* Testing is defined as the process in which defects are identified, isolated, subjected for rectification and ensured that product is defect free in order to produce the quality product and hence customer satisfaction.
* Quality is defined as justification of the requirements
* Defect is nothing but deviation from the requirements
* Defect is nothing but a bug.
* Testing --- The presence of bugs
* Testing can demonstrate the presence of bugs, but not their absence
* Debugging and Testing are not the same thing!
* Testing is a systematic attempt to break a program or the AUT
* Debugging is the art or method of uncovering why the script /program did not execute properly.

**Testing Methodologies:**

* **Black box Testing**: is the testing process in which tester can perform testing on an application without having any internal structural knowledge of application.

Usually Test Engineers are involved in the black box testing.

* **White box Testing**: is the testing process in which tester can perform testing on an application with internal structural knowledge.

Usually The Developers are involved in white box testing.

* **Gray Box Testing**: is the process in which the combination of black box and white box techniques are used.

**Levels of Testing:**

|  |  |
| --- | --- |
| |  | | --- | | ***Module1*** ***Module2*** ***Module3***  i/p ***Integration***  o/p i/p ***Integration o/p*** |   ***System Testing: Presentation + business +Databases***  ***UAT: user acceptance testing*** |

##### STLC (SOFTWARE TESTING LIFE CYCLE)

**Test Planning: 1.**Test Plan is defined as a strategic document which                                     describes the procedure how to perform various testing on the total                                      application in the most efficient way.

**2.** Objective of testing,

**3.** Areas that need to be tested,

**4.** Areas that should not be tested,

**5.** Scheduling Resource Planning,

**7.** Areas to be automated, various testing tools used

[

**Test Development**: **1.** Test case Development (check list)

**2.** Test Procedure preparation. (Description of the test cases)

**Test Execution**: **1.** Implementation of test cases. Observing the result.

**Result Analysis**: **1.** Expected value: is nothing but expected behavior

Of application.

**2.** Actual value: is nothing but actual behavior of the

application

**Bug Tracing:**  Collect all the failed cases, prepare documents.

**Reporting:** Prepare document (status of the application)

**Types Of Testing:**

* **Smoke Testing**: is the process of initial testing in which tester looks for the availability of all the functionality of the application in order to perform detailed testing on them. (Main check is for available forms)
* **Sanity Testing:** is a type of testing that is conducted on an application initially to check for the proper behavior of an application that is to check all the functionality are available before the detailed testing is conducted by them.
* **Regression Testing:** is one of the best and most important tests. Regression testing is the process in which the functionality, which is already tested before, is once again tested whenever some new change is added in order to check whether the existing functionality remains the same.
* **Re-Testing:** is the process in which testing is performed on some functionality which is already tested before to make sure that the defects are reproducible and to rule out the environmental issues if at all any defects are there.
* **Static Testing:** is the testing, which is performed on an application when it is not being executed. ex: GUI, Document Testing
* **Dynamic Testing:** is the testing which is performed on an application when it is being executed. ex: Functional testing.
* **Alpha Testing:** it is a type of user acceptance testing, which is conducted on an application when it is just before released to the customer.
* **Beta-Testing:** it is a type of UAT that is conducted on an application when it is released to the customer, when deployed in to the real time environment and being accessed by the real time users.
* **Monkey Testing:** is the process in which abnormal operations, beyond capacity operations are done on the application to check the stability of it in spite of the users abnormal behavior.
* **Compatibility testing:** it is the testing process in which usually the products are tested on the environments with different combinations of databases (application servers, browsers…etc) In order to check how far the product is compatible with all these environments platform combination.
* **Installation Testing:** it is the process of testing in which the tester tries to install or try to deploy the module into the corresponding environment by following the guidelines produced in the deployment document and check whether the installation is successful or not.
* **Adhoc Testing:** Adhoc Testing is the process of testing in which unlike the formal testing where in test case document is used, with out that test case document testing can be done of an application, to cover that testing of the future which are not covered in that test case document. Also it is intended to perform GUI testing which may involve cosmetic issues.

**TCD (Test Case Document):**

**Test Case Document Contains**

* **Test Scope (or) Test objective**
* **Test Scenario**
* **Test Procedure**
* **Test case**

This is the sample test case document for the Academic details of student project:

**Test scope:**

* Test coverage is provided for the screen “ Academic status entry” form of a student module of university management system application
* Areas of the application to be tested

**Test Scenario:**

* When the office personnel use this screen for the marks entry, calculate the status details, saving the information on student’s basis and quit the form.

**Test Procedure:**

* The procedure for testing this screen is planned in such a way that the data entry, status calculation functionality, saving and quitting operations are tested in terms of Gui testing, Positive testing, Negative testing using the corresponding Gui test cases, Positive test cases, Negative test cases respectively

**Test Cases:**

* Template for Test Case

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| T.C.No | Description | Exp | Act | Result |
|  |  |  |  |  |

**Guidelines for Test Cases**:

1. **GUI Test Cases:**

* Total no of features that need to be check
* Look & Feel
* Look for Default values if at all any (date & Time, if at all any require)
* Look for spell check

**Example for GUI Test cases**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| T.C.  No | Description | Expected value | Actual value | Result |
| 1 | Check for all the features in  the screen | The screen must contain  all the features |  |  |
| 2 | Check for the alignment of  the objects as per the validations | The alignment should be  in proper way |  |  |

1. **Positive Test Cases:**

* The positive flow of the functionality must be considered
* Valid inputs must be used for testing
* Must have the positive perception to verify whether the requirements are justified.

**Example for Positive Test cases:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| T.C.  No | Description | Expected value | Actual value | Result |
| 1 | Check for the date Time  Auto Display | The date and time of the system must be displayed |  |  |
| 2 | Enter the valid Roll no into the student roll no field | It should accept |  |  |

1. **Negative Test Cases:**

* Must have a negative perception.
* Invalid inputs must be used for tests.

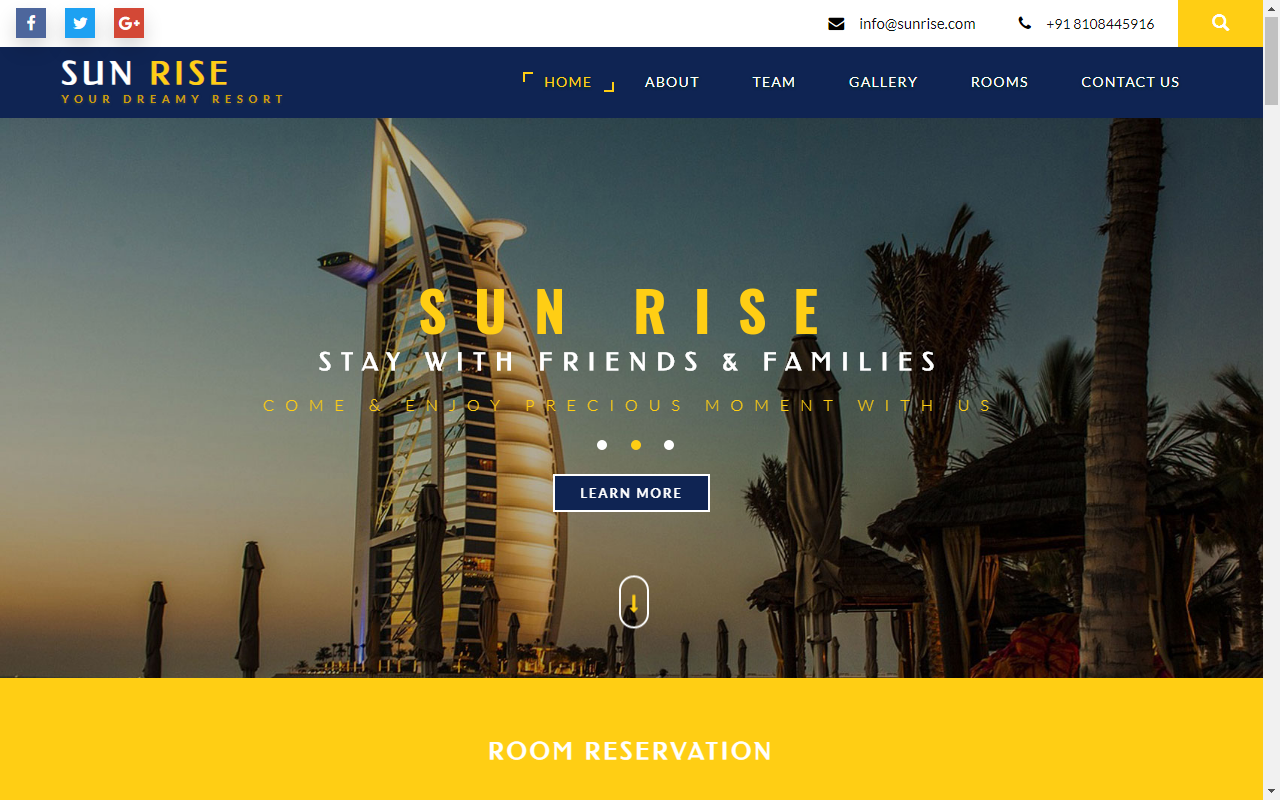
**Example for Negative Test cases**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| T.C.  No | Description | Expected value | Actual  value | Result |
| 1 | Try to modify the information in date and time | Modification should not  be allow |  |  |
| 2 | Enter invalid data in to the student details form, click on  Save | It should not accept invalid data, save should not allow |  |  |

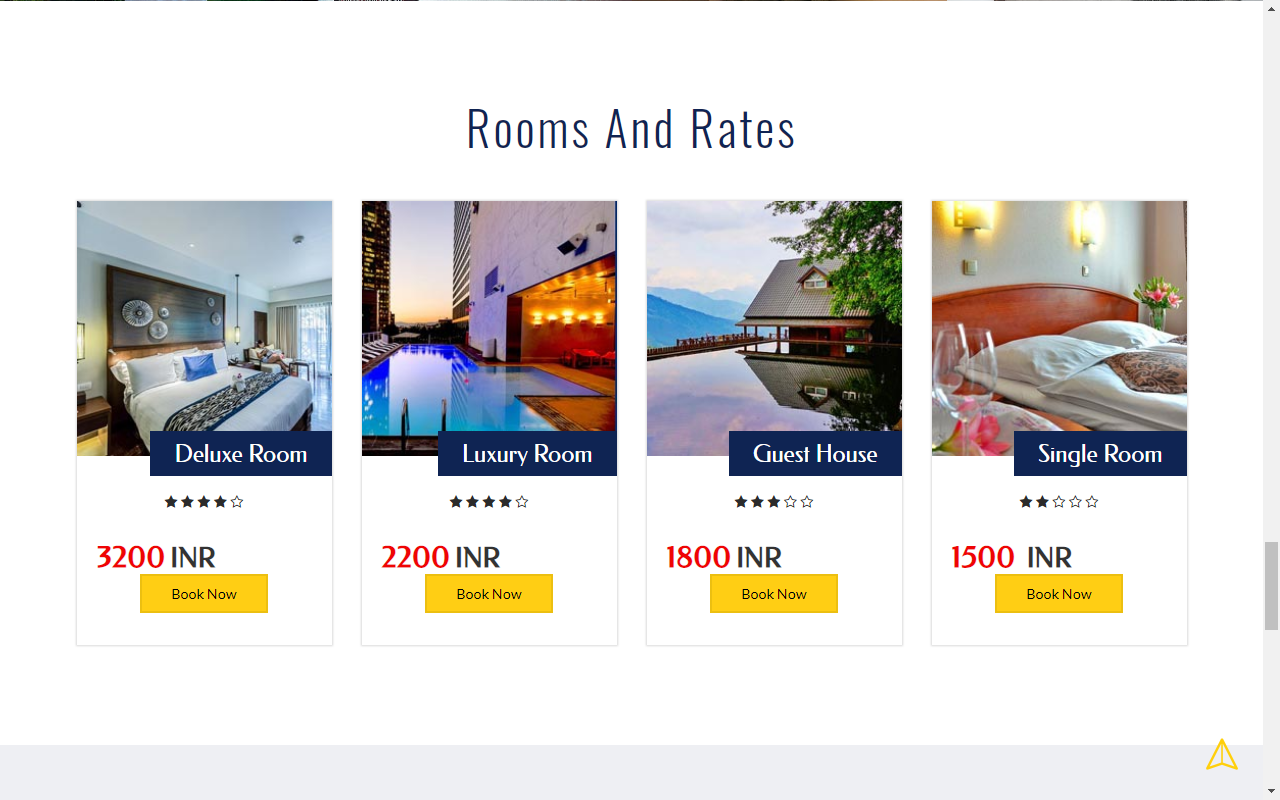
# 7. OUT-PUT SCREENS

## 7.1 Screens

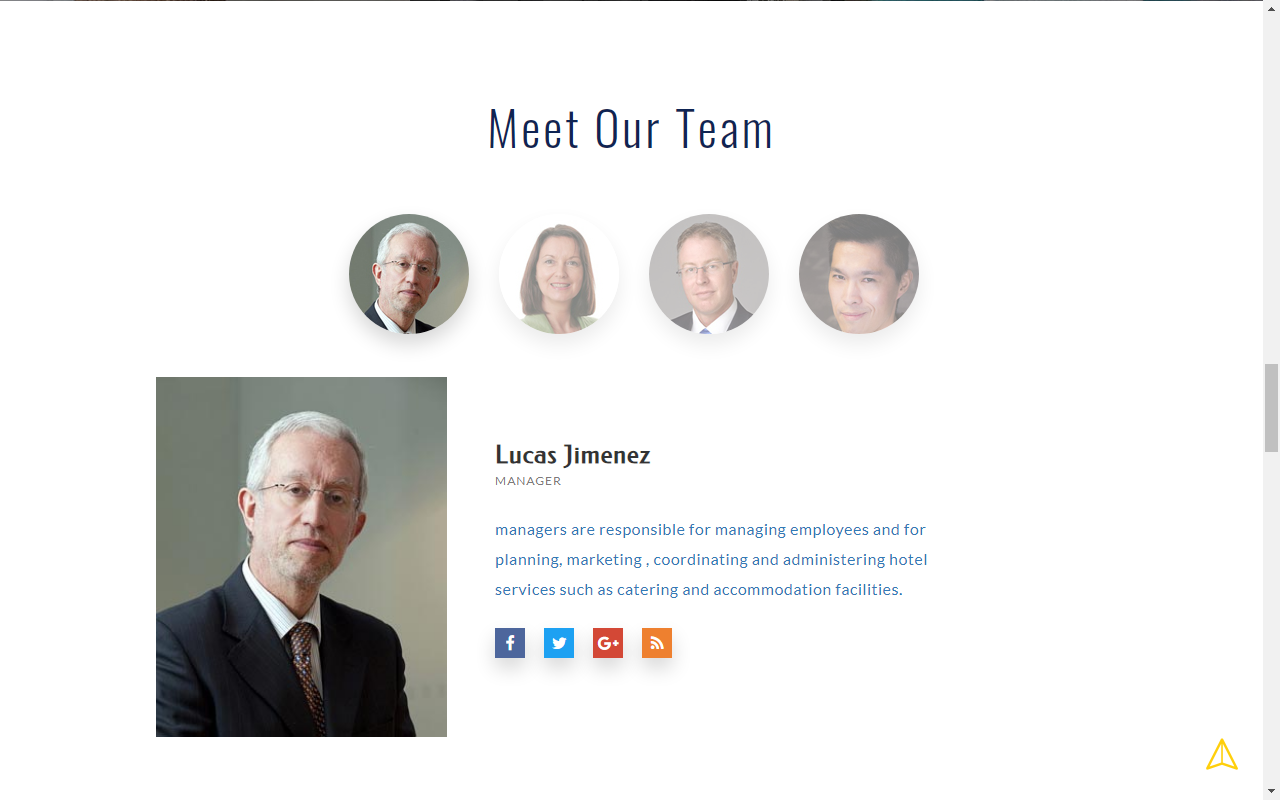
**Home Page:**



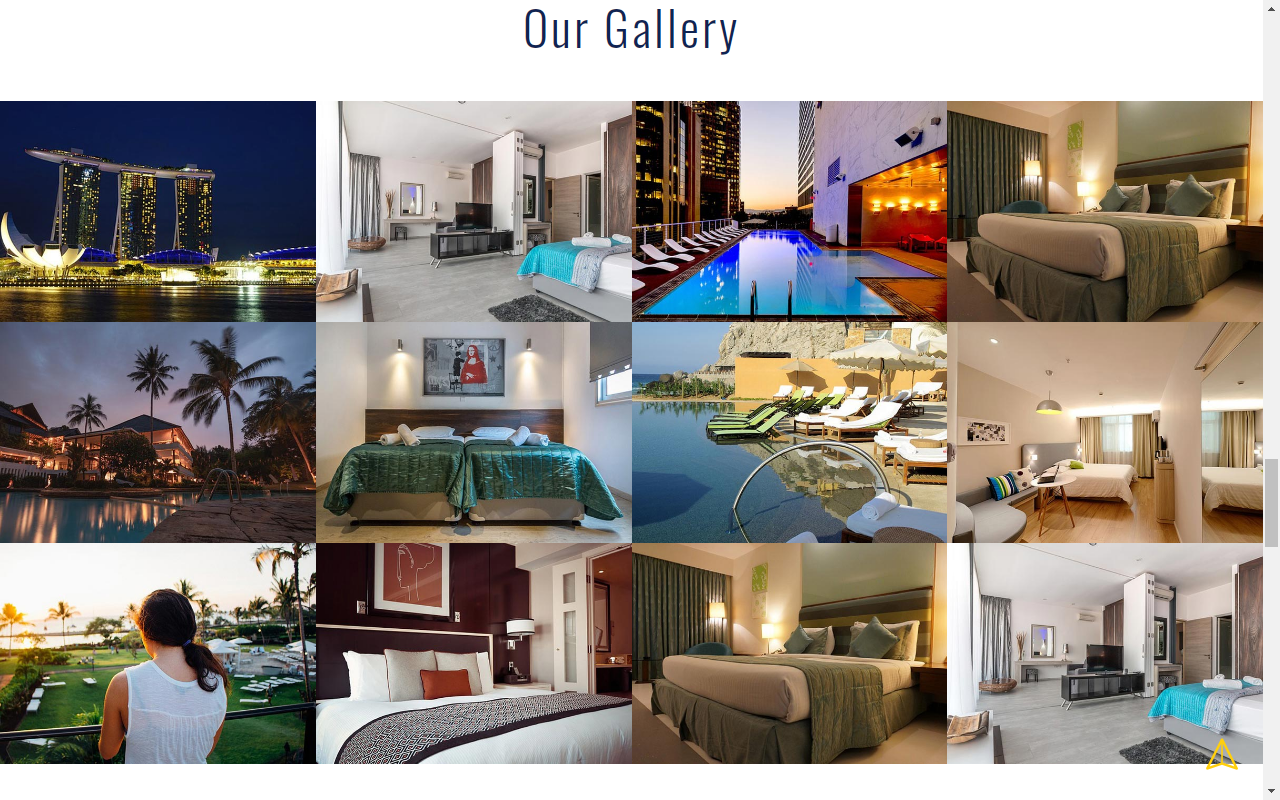
**Rooms:**



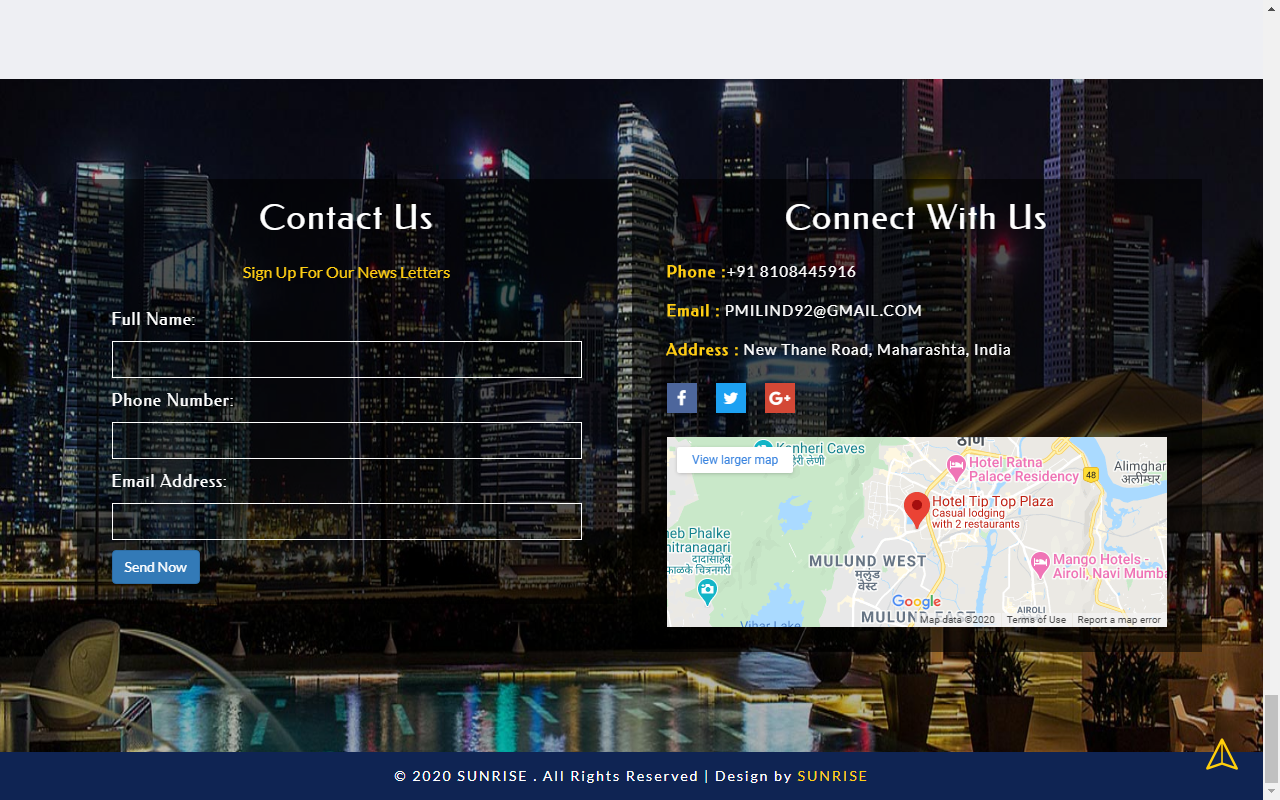
**Team:**



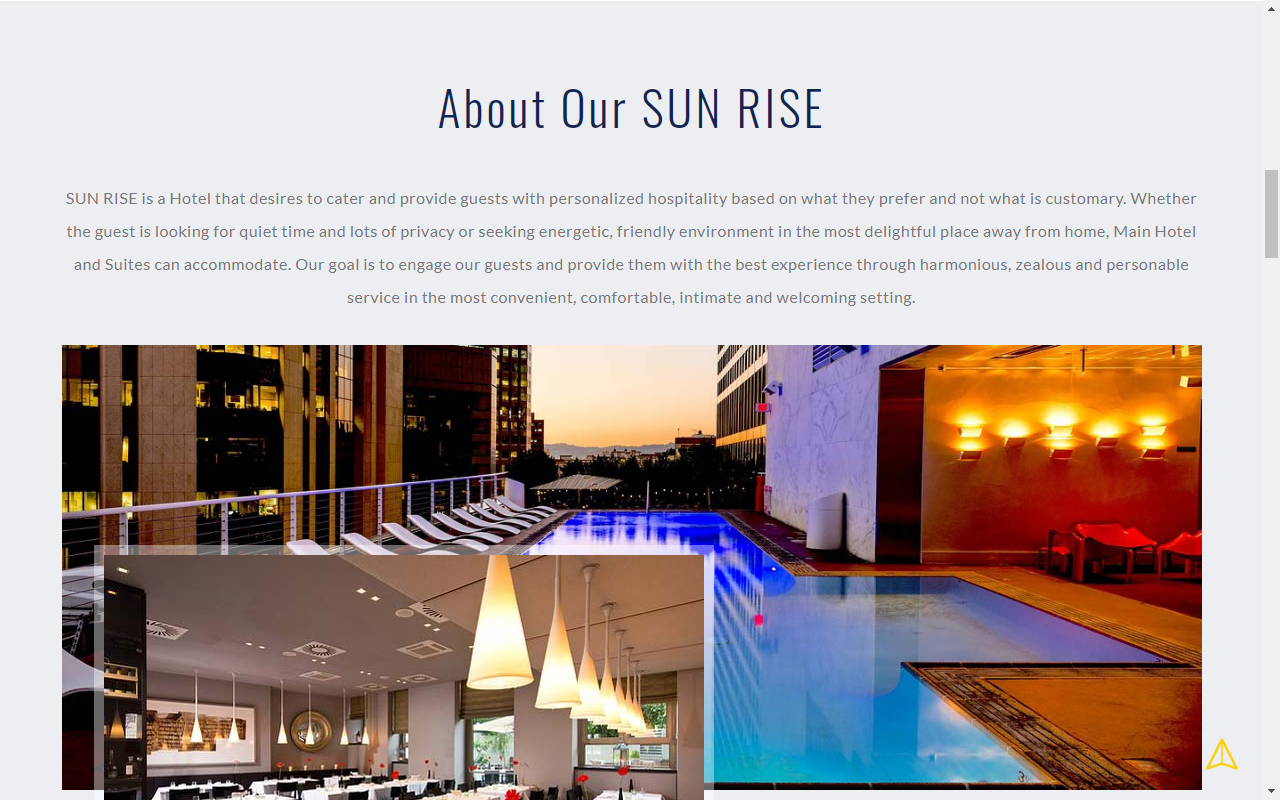
**Gallery:**



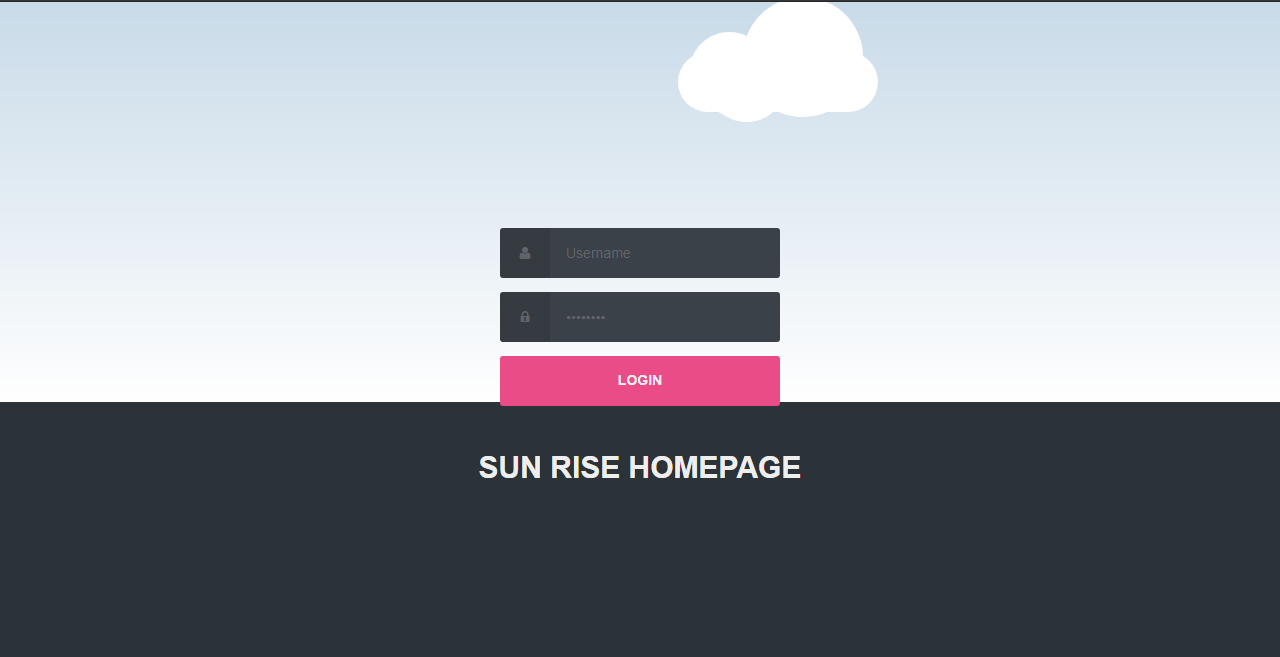
**Contact Us:**

****

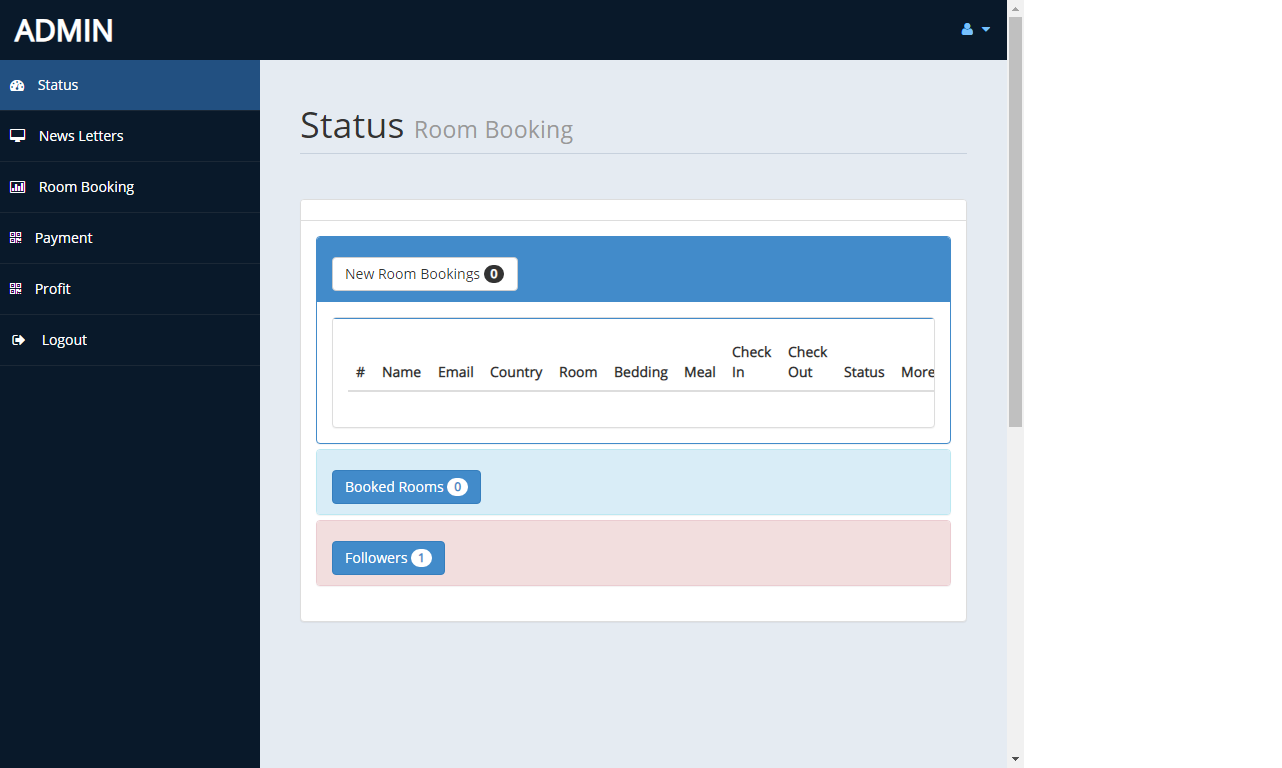
**About:**



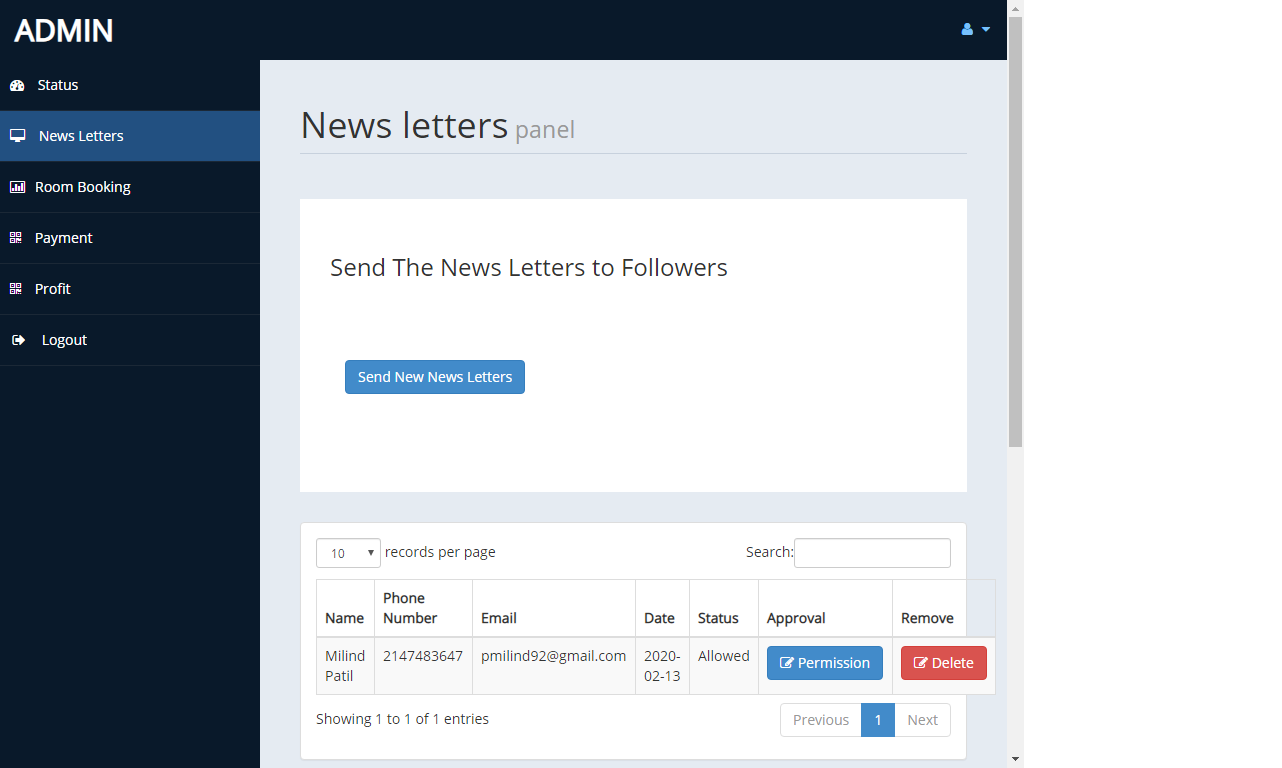
**Admin Login:**

****

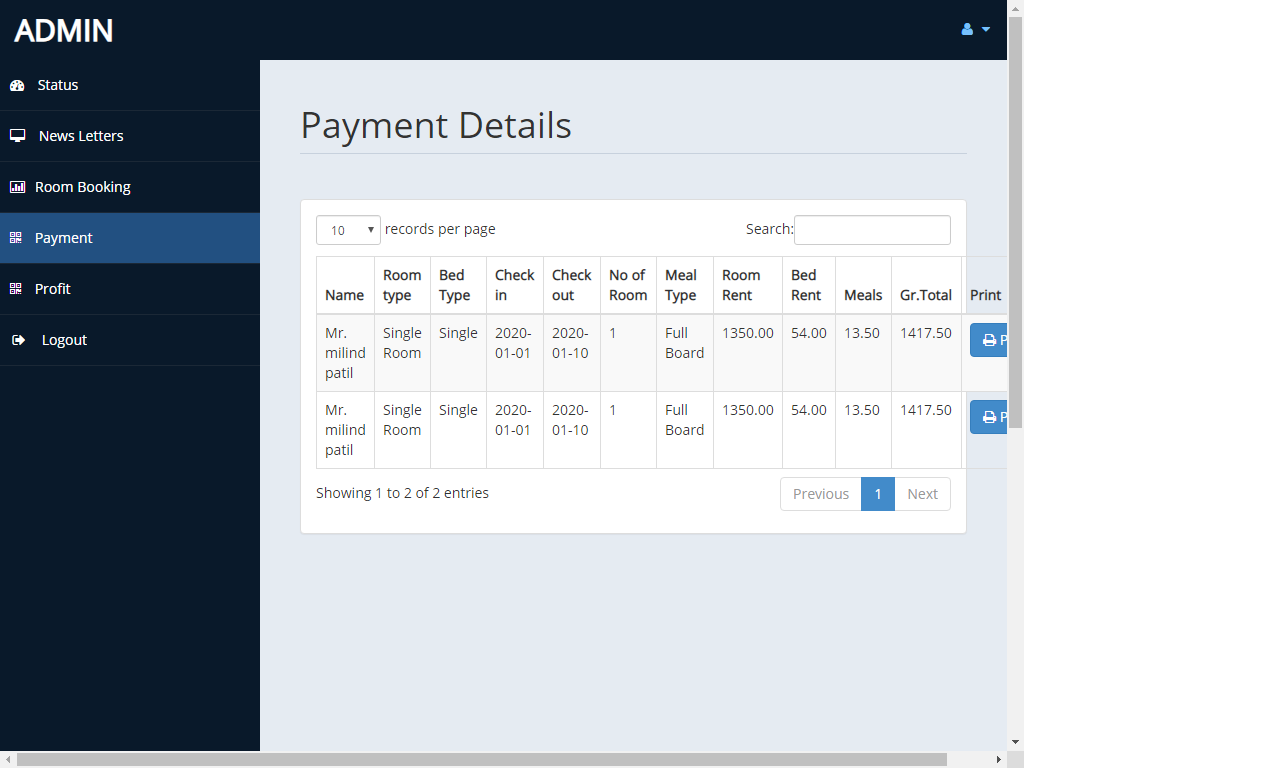
**Status Room Booking:**

****

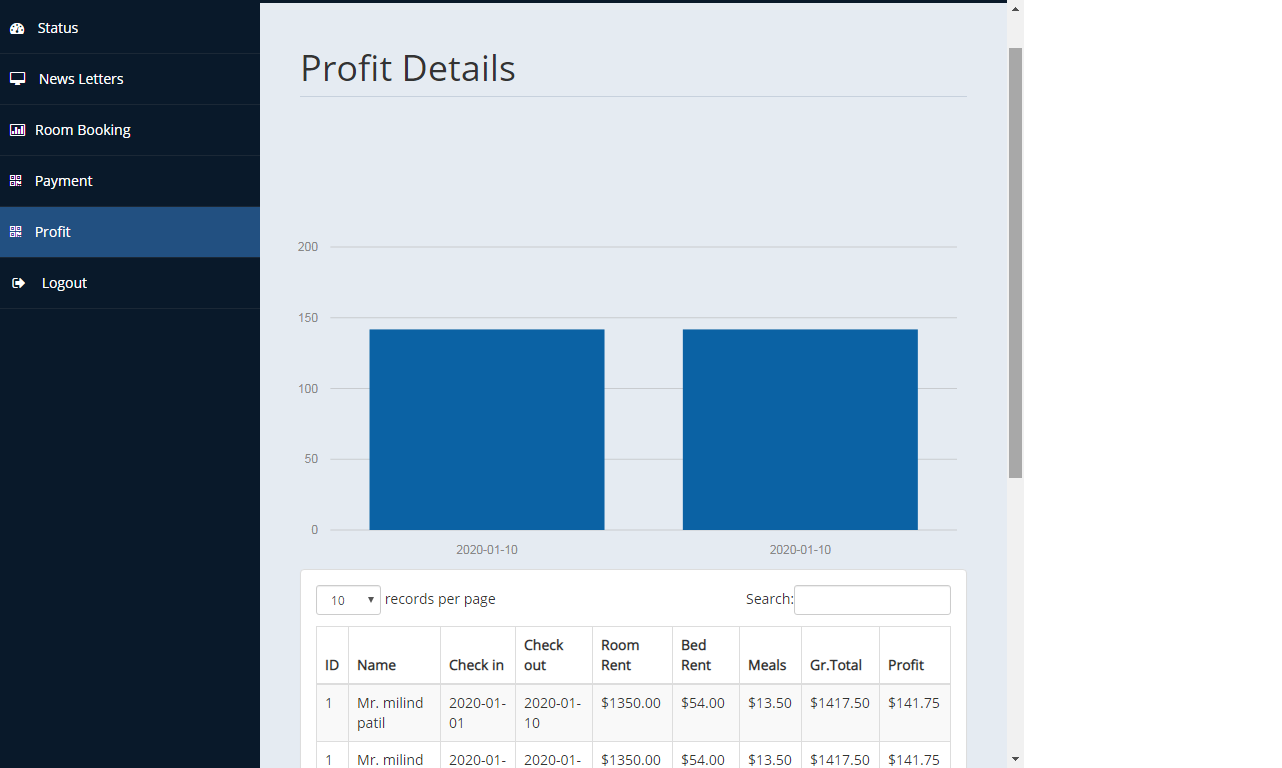
**NewsLetters panel:**

****

**Payment Details :**

****

**Profit Details:**



# 8.CODE

**INDEX.PHP**

**<?php**

**include('db.php');**

**?>**

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<title>SUNRISE HOTEL</title>**

**<!-- for-mobile-apps -->**

**<meta name="viewport" content="width=device-width, initial-scale=1">**

**<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />**

**<meta name="keywords" content="Resort Inn Responsive , Smartphone Compatible web template , Samsung, LG, Sony Ericsson, Motorola web design" />**

**<script type="application/x-javascript"> addEventListener("load", function() { setTimeout(hideURLbar, 0); }, false);**

**function hideURLbar(){ window.scrollTo(0,1); } </script>**

**<!-- //for-mobile-apps -->**

**<link href="css/bootstrap.css" rel="stylesheet" type="text/css" media="all" />**

**<link href="css/font-awesome.css" rel="stylesheet">**

**<link rel="stylesheet" href="css/chocolat.css" type="text/css" media="screen">**

**<link href="css/easy-responsive-tabs.css" rel='stylesheet' type='text/css'/>**

**<link rel="stylesheet" href="css/flexslider.css" type="text/css" media="screen" property="" />**

**<link rel="stylesheet" href="css/jquery-ui.css" />**

**<link href="css/style.css" rel="stylesheet" type="text/css" media="all" />**

**<script type="text/javascript" src="js/modernizr-2.6.2.min.js"></script>**

**<!--fonts-->**

**<link href="//fonts.googleapis.com/css?family=Oswald:300,400,700" rel="stylesheet">**

**<link href="//fonts.googleapis.com/css?family=Federo" rel="stylesheet">**

**<link href="//fonts.googleapis.com/css?family=Lato:300,400,700,900" rel="stylesheet">**

**<!--//fonts-->**

**</head>**

**<body>**

**<!-- header -->**

**<div class="banner-top">**

**<div class="social-bnr-agileits">**

**<ul class="social-icons3">**

**<li><a href="https://www.facebook.com/pmilind92" class="fa fa-facebook icon-border facebook"> </a></li>**

**<li><a href="https://twitter.com/pmilind92" class="fa fa-twitter icon-border twitter"> </a></li>**

**<li><a href="https://plus.google.com/u/0/" class="fa fa-google-plus icon-border googleplus"> </a></li>**

**</ul>**

**</div>**

**<div class="contact-bnr-w3-agile">**

**<ul>**

**<li><i class="fa fa-envelope" aria-hidden="true"></i><a href="mailto:info@example.com">INFO@SUNRISE.COM</a></li>**

**<li><i class="fa fa-phone" aria-hidden="true"></i>+91 8108445916</li>**

**<li class="s-bar">**

**<div class="search">**

**<input class="search\_box" type="checkbox" id="search\_box">**

**<label class="icon-search" for="search\_box"><span class="glyphicon glyphicon-search" aria-hidden="true"></span></label>**

**<div class="search\_form">**

**<form action="#" method="post">**

**<input type="search" name="Search" placeholder=" " required=" " />**

**<input type="submit" value="Search">**

**</form>**

**</div>**

**</div>**

**</li>**

**</ul>**

**</div>**

**<div class="clearfix"></div>**

**</div>**

**<div class="w3\_navigation">**

**<div class="container">**

**<nav class="navbar navbar-default">**

**<div class="navbar-header navbar-left">**

**<button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1">**

**<span class="sr-only">Toggle navigation</span>**

**<span class="icon-bar"></span>**

**<span class="icon-bar"></span>**

**<span class="icon-bar"></span>**

**</button>**

**<h1><a class="navbar-brand" href="index.php">SUN <span>RISE</span><p class="logo\_w3l\_agile\_caption">Your Dreamy Resort</p></a></h1>**

**</div>**

**<!-- Collect the nav links, forms, and other content for toggling -->**

**<div class="collapse navbar-collapse navbar-right" id="bs-example-navbar-collapse-1">**

**<nav class="menu menu--iris">**

**<ul class="nav navbar-nav menu\_\_list">**

**<li class="menu\_\_item menu\_\_item--current"><a href="" class="menu\_\_link">Home</a></li>**

**<li class="menu\_\_item"><a href="#about" class="menu\_\_link scroll">About</a></li>**

**<li class="menu\_\_item"><a href="#team" class="menu\_\_link scroll">Team</a></li>**

**<li class="menu\_\_item"><a href="#gallery" class="menu\_\_link scroll">Gallery</a></li>**

**<li class="menu\_\_item"><a href="#rooms" class="menu\_\_link scroll">Rooms</a></li>**

**<li class="menu\_\_item"><a href="#contact" class="menu\_\_link scroll">Contact Us</a></li> <li class="menu\_\_item"><a href="admin/index.php">Admin panel</a>**

**</ul>**

**</nav>**

**</div>**

**</nav>**

**</div>**

**</div>**

**<!-- //header -->**

**<!-- banner -->**

**<div id="home" class="w3ls-banner">**

**<!-- banner-text -->**

**<div class="slider">**

**<div class="callbacks\_container">**

**<ul class="rslides callbacks callbacks1" id="slider4">**

**<li>**

**<div class="w3layouts-banner-top">**

**<div class="container">**

**<div class="agileits-banner-info">**

**<h4>SUN RISE</h4>**

**<h3>We know what you love</h3>**

**<p>Welcome to our hotels</p>**

**<div class="agileits\_w3layouts\_more menu\_\_item">**

**<a href="#" class="menu\_\_link" data-toggle="modal" data-target="#myModal">Learn More</a>**

**</div>**

**</div>**

**</div>**

**</div>**

**</li>**

**<li>**

**<div class="w3layouts-banner-top w3layouts-banner-top1">**

**<div class="container">**

**<div class="agileits-banner-info">**

**<h4>SUN RISE</h4>**

**<h3>Stay with friends & families</h3>**

**<p>Come & enjoy precious moment with us</p>**

**<div class="agileits\_w3layouts\_more menu\_\_item">**

**<a href="#" class="menu\_\_link" data-toggle="modal" data-target="#myModal">Learn More</a>**

**</div>**

**</div>**

**</div>**

**</div>**

**</li>**

**<li>**

**<div class="w3layouts-banner-top w3layouts-banner-top2">**

**<div class="container">**

**<div class="agileits-banner-info">**

**<h4>SUN RISE</h4>**

**<h3>want luxurious vacation?</h3>**

**<p>Get accommodation today</p>**

**<div class="agileits\_w3layouts\_more menu\_\_item">**

**<a href="#" class="menu\_\_link" data-toggle="modal" data-target="#myModal">Learn More</a>**

**</div>**

**</div>**

**</div>**

**</div>**

**</li>**

**</ul>**

**</div>**

**<div class="clearfix"> </div>**

**<!--banner Slider starts Here-->**

**</div>**

**<div class="thim-click-to-bottom">**

**<a href="#about" class="scroll">**

**<i class="fa fa-long-arrow-down" aria-hidden="true"></i>**

**</a>**

**</div>**

**</div>**

**<!-- //banner -->**

**<!--//Header-->**

**<!-- //Modal1 -->**

**<div class="modal fade" id="myModal" tabindex="-1" role="dialog">**

**<!-- Modal1 -->**

**<div class="modal-dialog">**

**<!-- Modal content-->**

**<div class="modal-content">**

**<div class="modal-header">**

**<button type="button" class="close" data-dismiss="modal">&times;</button>**

**<h4>SUN <span>RISE</span></h4>**

**<img src="images/1.jpg" alt=" " class="img-responsive">**

**<h5>We know what you love</h5>**

**<p>Providing guests unique and enchanting views from their rooms with its exceptional amenities, makes Star Hotel one of bests in its kind.Try our food menu, awesome services and friendly staff while you are here.</p>**

**</div>**

**</div>**

**</div>**

**</div>**

**<!-- //Modal1 -->**

**<div id="availability-agileits">**

**<div class="col-md-12 book-form-left-w3layouts">**

**<a href="admin/reservation.php"><h2>ROOM RESERVATION</h2></a>**

**</div>**

**<div class="clearfix"> </div>**

**</div>**

**<!-- banner-bottom -->**

**<div class="banner-bottom">**

**<div class="container">**

**<div class="agileits\_banner\_bottom">**

**<h3><span>Experience a good stay, enjoy fantastic offers</span> Find our friendly welcoming reception</h3>**

**</div>**

**<div class="w3ls\_banner\_bottom\_grids">**

**<ul class="cbp-ig-grid">**

**<li>**

**<div class="w3\_grid\_effect">**

**<span class="cbp-ig-icon w3\_road"></span>**

**<h4 class="cbp-ig-title">MASTER BEDROOMS</h4>**

**<span class="cbp-ig-category">SUN RISE</span>**

**</div>**

**</li>**

**<li>**

**<div class="w3\_grid\_effect">**

**<span class="cbp-ig-icon w3\_cube"></span>**

**<h4 class="cbp-ig-title">SEA VIEW BALCONY</h4>**

**<span class="cbp-ig-category">SUN RISE</span>**

**</div>**

**</li>**

**<li>**

**<div class="w3\_grid\_effect">**

**<span class="cbp-ig-icon w3\_users"></span>**

**<h4 class="cbp-ig-title">LARGE CAFE</h4>**

**<span class="cbp-ig-category">SUN RISE</span>**

**</div>**

**</li>**

**<li>**

**<div class="w3\_grid\_effect">**

**<span class="cbp-ig-icon w3\_ticket"></span>**

**<h4 class="cbp-ig-title">WIFI COVERAGE</h4>**

**<span class="cbp-ig-category">SUN RISE</span>**

**</div>**

**</li>**

**</ul>**

**</div>**

**</div>**

**</div>**

**<!-- //banner-bottom -->**

**<!-- /about -->**

**<div class="about-wthree" id="about">**

**<div class="container">**

**<div class="ab-w3l-spa">**

**<h3 class="title-w3-agileits title-black-wthree">About Our SUN RISE</h3>**

**<p class="about-para-w3ls"> SUN RISE is a Hotel that desires to cater and provide guests with personalized hospitality based on what they prefer and not what is customary. Whether the guest is looking for quiet time and lots of privacy or seeking energetic, friendly environment in the most delightful place away from home, Main Hotel and Suites can accommodate. Our goal is to engage our guests and provide them with the best experience through harmonious, zealous and personable service in the most convenient, comfortable, intimate and welcoming setting.</p>**

**<img src="images/about.jpg" class="img-responsive" alt="Hair Salon">**

**<div class="w3l-slider-img">**

**<img src="images/a1.jpg" class="img-responsive" alt="Hair Salon">**

**</div>**

**<div class="w3ls-info-about">**

**<h4>You'll love all the amenities we offer!</h4>**

**<p>Lorem ipsum dolor sit amet, ut magna aliqua. </p>**

**</div>**

**</div>**

**<div class="clearfix"> </div>**

**</div>**

**</div>**

**<!-- //about -->**

**<!--sevices-->**

**<div class="advantages">**

**<div class="container">**

**<div class="advantages-main">**

**<h3 class="title-w3-agileits">Our Services</h3>**

**<div class="advantage-bottom">**

**<div class="col-md-6 advantage-grid left-w3ls wow bounceInLeft" data-wow-delay="0.3s">**

**<div class="advantage-block ">**

**<i class="fa fa-credit-card" aria-hidden="true"></i>**

**<h4>Stay First, Pay After! </h4>**

**<p>The needs of the times and in certain circumstances and will frequently occur that pleasures have to of things, or avoids pleasure.</p>**

**<p><i class="fa fa-check" aria-hidden="true"></i>Decorated room, proper air conditioned</p>**

**<p><i class="fa fa-check" aria-hidden="true"></i>Private balcony</p>**

**</div>**

**</div>**

**<div class="col-md-6 advantage-grid right-w3ls wow zoomIn" data-wow-delay="0.3s">**

**<div class="advantage-block">**

**<i class="fa fa-clock-o" aria-hidden="true"></i>**

**<h4>24 Hour Restaurant</h4>**

**<p>The needs of the times and in certain circumstances and will frequently occur that pleasures have to of things, or avoids pleasure.</p>**

**<p><i class="fa fa-check" aria-hidden="true"></i>24 hours room service</p>**

**<p><i class="fa fa-check" aria-hidden="true"></i>24-hour Concierge service</p>**

**</div>**

**</div>**

**<div class="clearfix"> </div>**

**</div>**

**</div>**

**</div>**

**</div>**

**<!--//sevices-->**

**<!-- team -->**

**<div class="team" id="team">**

**<div class="container">**

**<h3 class="title-w3-agileits title-black-wthree">Meet Our Team</h3>**

**<div id="horizontalTab">**

**<ul class="resp-tabs-list">**

**<li>**

**<img src="images/teams1.jpg" alt=" " class="img-responsive" />**

**</li>**

**<li>**

**<img src="images/teams2.jpg" alt=" " class="img-responsive" />**

**</li>**

**<li>**

**<img src="images/teams3.jpg" alt=" " class="img-responsive" />**

**</li>**

**<li>**

**<img src="images/teams4.jpg" alt=" " class="img-responsive" />**

**</li>**

**</ul>**

**<div class="resp-tabs-container">**

**<div class="tab1">**

**<div class="col-md-6 team-img-w3-agile">**

**</div>**

**<div class="col-md-6 team-Info-agileits">**

**<h4>Lucas Jimenez</h4>**

**<span>Manager</span>**

**<p>managers are responsible for managing employees and for planning, marketing , coordinating and administering hotel services such as catering and accommodation facilities.</p>**

**<div class="social-bnr-agileits footer-icons-agileinfo">**

**<ul class="social-icons3">**

**<li><a href="#" class="fa fa-facebook icon-border facebook"> </a></li>**

**<li><a href="#" class="fa fa-twitter icon-border twitter"> </a></li>**

**<li><a href="#" class="fa fa-google-plus icon-border googleplus"> </a></li>**

**<li><a href="#" class="fa fa-rss icon-border rss"> </a></li>**

**</ul>**

**</div>**

**</div>**

**<div class="clearfix"> </div>**

**</div>**

**<div class="tab2">**

**<div class="col-md-6 team-img-w3-agile">**

**</div>**

**<div class="col-md-6 team-Info-agileits">**

**<h4>Sarah Connor</h4>**

**<span>Receptionist</span>**

**<p>Hotel receptionists greet incoming guests and welcome them to the hotel. They confirm reservations, explain hotel amenities, give directions to rooms, and produce room keys. They also arrange for luggage to be brought to rooms upon request.</p>**

**<div class="social-bnr-agileits footer-icons-agileinfo">**

**<ul class="social-icons3">**

**<li><a href="#" class="fa fa-facebook icon-border facebook"> </a></li>**

**<li><a href="#" class="fa fa-twitter icon-border twitter"> </a></li>**

**<li><a href="#" class="fa fa-google-plus icon-border googleplus"> </a></li>**

**<li><a href="#" class="fa fa-rss icon-border rss"> </a></li>**

**</ul>**

**</div>**

**</div>**

**<div class="clearfix"> </div>**

**</div>**

**<div class="tab3">**

**<div class="col-md-6 team-img-w3-agile">**

**</div>**

**<div class="col-md-6 team-Info-agileits">**

**<h4>Ivan Simpson</h4>**

**<span>Manager</span>**

**<p>managers are responsible for managing employees and for planning, marketing , coordinating and administering hotel services such as catering and accommodation facilities.</p>**

**<div class="social-bnr-agileits footer-icons-agileinfo">**

**<ul class="social-icons3">**

**<li><a href="#" class="fa fa-facebook icon-border facebook"> </a></li>**

**<li><a href="#" class="fa fa-twitter icon-border twitter"> </a></li>**

**<li><a href="#" class="fa fa-google-plus icon-border googleplus"> </a></li>**

**<li><a href="#" class="fa fa-rss icon-border rss"> </a></li>**

**</ul>**

**</div>**

**</div>**

**<div class="clearfix"> </div>**

**</div>**

**<div class="tab4">**

**<div class="col-md-6 team-img-w3-agile">**

**</div>**

**<div class="col-md-6 team-Info-agileits">**

**<h4>Marc Gutierrez</h4>**

**<span>Receptionist</span>**

**<p>Hotel receptionists greet incoming guests and welcome them to the hotel. They confirm reservations, explain hotel amenities, give directions to rooms, and produce room keys. They also arrange for luggage to be brought to rooms upon request.</p>**

**<div class="social-bnr-agileits footer-icons-agileinfo">**

**<ul class="social-icons3">**

**<li><a href="#" class="fa fa-facebook icon-border facebook"> </a></li>**

**<li><a href="#" class="fa fa-twitter icon-border twitter"> </a></li>**

**<li><a href="#" class="fa fa-google-plus icon-border googleplus"> </a></li>**

**<li><a href="#" class="fa fa-rss icon-border rss"> </a></li>**

**</ul>**

**</div>**

**</div>**

**<div class="clearfix"> </div>**

**</div>**

**</div>**

**</div>**

**</div>**

**</div>**

**<!-- //team -->**

**<!-- Gallery -->**

**<section class="portfolio-w3ls" id="gallery">**

**<h3 class="title-w3-agileits title-black-wthree">Our Gallery</h3>**

**<div class="col-md-3 gallery-grid gallery1">**

**<a href="images/g1.jpg" class="swipebox"><img src="images/g1.jpg" class="img-responsive" alt="/">**

**<div class="textbox">**

**<h4>SUN RISE</h4>**

**<p><i class="fa fa-picture-o" aria-hidden="true"></i></p>**

**</div>**

**</a>**

**</div>**

**<div class="col-md-3 gallery-grid gallery1">**

**<a href="images/g2.jpg" class="swipebox"><img src="images/g2.jpg" class="img-responsive" alt="/">**

**<div class="textbox">**

**<h4>SUN RISE</h4>**

**<p><i class="fa fa-picture-o" aria-hidden="true"></i></p>**

**</div>**

**</a>**

**</div>**

**<div class="col-md-3 gallery-grid gallery1">**

**<a href="images/g3.jpg" class="swipebox"><img src="images/g3.jpg" class="img-responsive" alt="/">**

**<div class="textbox">**

**<h4>SUN RISE</h4>**

**<p><i class="fa fa-picture-o" aria-hidden="true"></i></p>**

**</div>**

**</a>**

**</div>**

**<div class="col-md-3 gallery-grid gallery1">**

**<a href="images/g4.jpg" class="swipebox"><img src="images/g4.jpg" class="img-responsive" alt="/">**

**<div class="textbox">**

**<h4>SUN RISE</h4>**

**<p><i class="fa fa-picture-o" aria-hidden="true"></i></p>**

**</div>**

**</a>**

**</div>**

**<div class="col-md-3 gallery-grid gallery1">**

**<a href="images/g5.jpg" class="swipebox"><img src="images/g5.jpg" class="img-responsive" alt="/">**

**<div class="textbox">**

**<h4>SUN RISE</h4>**

**<p><i class="fa fa-picture-o" aria-hidden="true"></i></p>**

**</div>**

**</a>**

**</div>**

**<div class="col-md-3 gallery-grid gallery1">**

**<a href="images/g6.jpg" class="swipebox"><img src="images/g6.jpg" class="img-responsive" alt="/">**

**<div class="textbox">**

**<h4>SUN RISE</h4>**

**<p><i class="fa fa-picture-o" aria-hidden="true"></i></p>**

**</div>**

**</a>**

**</div>**

**<div class="col-md-3 gallery-grid gallery1">**

**<a href="images/g6.jpg" class="swipebox"><img src="images/g7.jpg" class="img-responsive" alt="/">**

**<div class="textbox">**

**<h4>SUN RISE</h4>**

**<p><i class="fa fa-picture-o" aria-hidden="true"></i></p>**

**</div>**

**</a>**

**</div>**

**<div class="col-md-3 gallery-grid gallery1">**

**<a href="images/g6.jpg" class="swipebox"><img src="images/g8.jpg" class="img-responsive" alt="/">**

**<div class="textbox">**

**<h4>SUN RISE</h4>**

**<p><i class="fa fa-picture-o" aria-hidden="true"></i></p>**

**</div>**

**</a>**

**</div>**

**<div class="col-md-3 gallery-grid gallery1">**

**<a href="images/g9.jpg" class="swipebox"><img src="images/g9.jpg" class="img-responsive" alt="/">**

**<div class="textbox">**

**<h4>SUN RISE</h4>**

**<p><i class="fa fa-picture-o" aria-hidden="true"></i></p>**

**</div>**

**</a>**

**</div>**

**<div class="col-md-3 gallery-grid gallery1">**

**<a href="images/g10.jpg" class="swipebox"><img src="images/g10.jpg" class="img-responsive" alt="/">**

**<div class="textbox">**

**<h4>SUN RISE</h4>**

**<p><i class="fa fa-picture-o" aria-hidden="true"></i></p>**

**</div>**

**</a>**

**</div>**

**<div class="col-md-3 gallery-grid gallery1">**

**<a href="images/g4.jpg" class="swipebox"><img src="images/g4.jpg" class="img-responsive" alt="/">**

**<div class="textbox">**

**<h4>SUN RISE</h4>**

**<p><i class="fa fa-picture-o" aria-hidden="true"></i></p>**

**</div>**

**</a>**

**</div>**

**<div class="col-md-3 gallery-grid gallery1">**

**<a href="images/g2.jpg" class="swipebox"><img src="images/g2.jpg" class="img-responsive" alt="/">**

**<div class="textbox">**

**<h4>SUN RISE</h4>**

**<p><i class="fa fa-picture-o" aria-hidden="true"></i></p>**

**</div>**

**</a>**

**</div>**

**<div class="clearfix"> </div>**

**</section>**

**<!-- //gallery -->**

**<!-- rooms & rates -->**

**<div class="plans-section" id="rooms">**

**<div class="container">**

**<h3 class="title-w3-agileits title-black-wthree">Rooms And Rates</h3>**

**<div class="priceing-table-main">**

**<div class="col-md-3 price-grid">**

**<div class="price-block agile">**

**<div class="price-gd-top">**

**<img src="images/r1.jpg" alt=" " class="img-responsive" />**

**<h4>Deluxe Room</h4>**

**</div>**

**<div class="price-gd-bottom">**

**<div class="price-list">**

**<ul>**

**<li><i class="fa fa-star" aria-hidden="true"></i></li>**

**<li><i class="fa fa-star" aria-hidden="true"></i></li>**

**<li><i class="fa fa-star" aria-hidden="true"></i></li>**

**<li><i class="fa fa-star" aria-hidden="true"></i></li>**

**<li><i class="fa fa-star-o" aria-hidden="true"></i></li>**

**</ul>**

**</div>**

**<div class="price-selet">**

**<h3><span>3200</span>INR</h3>**

**<a href="admin/reservation.php" >Book Now</a>**

**</div>**

**</div>**

**</div>**

**</div>**

**<div class="col-md-3 price-grid ">**

**<div class="price-block agile">**

**<div class="price-gd-top">**

**<img src="images/r2.jpg" alt=" " class="img-responsive" />**

**<h4>Luxury Room</h4>**

**</div>**

**<div class="price-gd-bottom">**

**<div class="price-list">**

**<ul>**

**<li><i class="fa fa-star" aria-hidden="true"></i></li>**

**<li><i class="fa fa-star" aria-hidden="true"></i></li>**

**<li><i class="fa fa-star" aria-hidden="true"></i></li>**

**<li><i class="fa fa-star" aria-hidden="true"></i></li>**

**<li><i class="fa fa-star-o" aria-hidden="true"></i></li>**

**</ul>**

**</div>**

**<div class="price-selet">**

**<h3><span>2200</span>INR</h3>**

**<a href="admin/reservation.php" >Book Now</a>**

**</div>**

**</div>**

**</div>**

**</div>**

**<div class="col-md-3 price-grid lost">**

**<div class="price-block agile">**

**<div class="price-gd-top">**

**<img src="images/r3.jpg" alt=" " class="img-responsive" />**

**<h4>Guest House</h4>**

**</div>**

**<div class="price-gd-bottom">**

**<div class="price-list">**

**<ul>**

**<li><i class="fa fa-star" aria-hidden="true"></i></li>**

**<li><i class="fa fa-star" aria-hidden="true"></i></li>**

**<li><i class="fa fa-star" aria-hidden="true"></i></li>**

**<li><i class="fa fa-star-o" aria-hidden="true"></i></li>**

**<li><i class="fa fa-star-o" aria-hidden="true"></i></li>**

**</ul>**

**</div>**

**<div class="price-selet">**

**<h3><span>1800</span>INR</h3>**

**<a href="admin/reservation.php" >Book Now</a>**

**</div>**

**</div>**

**</div>**

**</div>**

**<div class="col-md-3 price-grid wthree lost">**

**<div class="price-block agile">**

**<div class="price-gd-top ">**

**<img src="images/r4.jpg" alt=" " class="img-responsive" />**

**<h4>Single Room</h4>**

**</div>**

**<div class="price-gd-bottom">**

**<div class="price-list">**

**<ul>**

**<li><i class="fa fa-star" aria-hidden="true"></i></li>**

**<li><i class="fa fa-star" aria-hidden="true"></i></li>**

**<li><i class="fa fa-star-o" aria-hidden="true"></i></li>**

**<li><i class="fa fa-star-o" aria-hidden="true"></i></li>**

**<li><i class="fa fa-star-o" aria-hidden="true"></i></li>**

**</ul>**

**</div>**

**<div class="price-selet">**

**<h3><span>1500</span> INR</h3>**

**<a href="admin/reservation.php" >Book Now</a>**

**</div>**

**</div>**

**</div>**

**</div>**

**<div class="clearfix"> </div>**

**</div>**

**</div>**

**</div>**

**<!--// rooms & rates -->**

**<!-- visitors -->**

**<div class="w3l-visitors-agile" >**

**<div class="container">**

**<h3 class="title-w3-agileits title-black-wthree">What other visitors experienced</h3>**

**</div>**

**<div class="w3layouts\_work\_grids">**

**<section class="slider">**

**<div class="flexslider">**

**<ul class="slides">**

**<li>**

**<div class="w3layouts\_work\_grid\_left">**

**<img src="images/5.jpg" alt=" " class="img-responsive" />**

**<div class="w3layouts\_work\_grid\_left\_pos">**

**<img src="images/c1.jpg" alt=" " class="img-responsive" />**

**</div>**

**</div>**

**<div class="w3layouts\_work\_grid\_right">**

**<h4>**

**<i class="fa fa-star" aria-hidden="true"></i>**

**<i class="fa fa-star" aria-hidden="true"></i>**

**<i class="fa fa-star" aria-hidden="true"></i>**

**<i class="fa fa-star" aria-hidden="true"></i>**

**<i class="fa fa-star" aria-hidden="true"></i>**

**Worth to come again**

**</h4>**

**<h5>Julia Rose</h5>**

**<p>Germany</p>**

**</div>**

**<div class="clearfix"> </div>**

**</li>**

**<li>**

**<div class="w3layouts\_work\_grid\_left">**

**<img src="images/5.jpg" alt=" " class="img-responsive" />**

**<div class="w3layouts\_work\_grid\_left\_pos">**

**<img src="images/c2.jpg" alt=" " class="img-responsive" />**

**</div>**

**</div>**

**<div class="w3layouts\_work\_grid\_right">**

**<h4>**

**<i class="fa fa-star" aria-hidden="true"></i>**

**<i class="fa fa-star" aria-hidden="true"></i>**

**<i class="fa fa-star" aria-hidden="true"></i>**

**<i class="fa fa-star" aria-hidden="true"></i>**

**<i class="fa fa-star-o" aria-hidden="true"></i>**

**Worth to come again**

**</h4>**

**<h5>Jahnatan Smith</h5>**

**<p>United States</p>**

**</div>**

**<div class="clearfix"> </div>**

**</li>**

**<li>**

**<div class="w3layouts\_work\_grid\_left">**

**<img src="images/5.jpg" alt=" " class="img-responsive" />**

**<div class="w3layouts\_work\_grid\_left\_pos">**

**<img src="images/c3.jpg" alt=" " class="img-responsive" />**

**</div>**

**</div>**

**<div class="w3layouts\_work\_grid\_right">**

**<h4>**

**<i class="fa fa-star" aria-hidden="true"></i>**

**<i class="fa fa-star" aria-hidden="true"></i>**

**<i class="fa fa-star" aria-hidden="true"></i>**

**<i class="fa fa-star" aria-hidden="true"></i>**

**<i class="fa fa-star-o" aria-hidden="true"></i>**

**Worth to come again**

**</h4>**

**<h5>Rosalind Cloer</h5>**

**<p>Italy</p>**

**</div>**

**<div class="clearfix"> </div>**

**</li>**

**<li>**

**<div class="w3layouts\_work\_grid\_left">**

**<img src="images/5.jpg" alt=" " class="img-responsive" />**

**<div class="w3layouts\_work\_grid\_left\_pos">**

**<img src="images/c4.jpg" alt=" " class="img-responsive" />**

**</div>**

**</div>**

**<div class="w3layouts\_work\_grid\_right">**

**<h4>**

**<i class="fa fa-star" aria-hidden="true"></i>**

**<i class="fa fa-star" aria-hidden="true"></i>**

**<i class="fa fa-star" aria-hidden="true"></i>**

**<i class="fa fa-star-o" aria-hidden="true"></i>**

**<i class="fa fa-star-o" aria-hidden="true"></i>**

**Worth to come again**

**</h4>**

**<h5>Amie Bublitz</h5>**

**<p>Switzerland</p>**

**</div>**

**<div class="clearfix"> </div>**

**</li>**

**</ul>**

**</div>**

**</section>**

**</div>**

**</div>**

**<!-- visitors -->**

**<!-- contact -->**

**<section class="contact-w3ls" id="contact">**

**<div class="container">**

**<div class="col-lg-6 col-md-6 col-sm-6 contact-w3-agile2" data-aos="flip-left">**

**<div class="contact-agileits">**

**<h4>Contact Us</h4>**

**<p class="contact-agile2">Sign Up For Our News Letters</p>**

**<form method="post" name="sentMessage" id="contactForm" >**

**<div class="control-group form-group">**

**<label class="contact-p1">Full Name:</label>**

**<input type="text" class="form-control" name="name" id="name" required >**

**<p class="help-block"></p>**

**</div>**

**<div class="control-group form-group">**

**<label class="contact-p1">Phone Number:</label>**

**<input type="tel" class="form-control" name="phone" id="phone" required >**

**<p class="help-block"></p>**

**</div>**

**<div class="control-group form-group">**

**<label class="contact-p1">Email Address:</label>**

**<input type="email" class="form-control" name="email" id="email" required >**

**<p class="help-block"></p>**

**</div>**

**<input type="submit" name="sub" value="Send Now" class="btn btn-primary">**

**</form>**

**<?php**

**if(isset($\_POST['sub']))**

**{**

**$name =$\_POST['name'];**

**$phone = $\_POST['phone'];**

**$email = $\_POST['email'];**

**$approval = "Not Allowed";**

**$sql = "INSERT INTO `contact`(`fullname`, `phoneno`, `email`,`cdate`,`approval`) VALUES ('$name','$phone','$email',now(),'$approval')" ;**

**if(mysqli\_query($con,$sql))**

**echo"OK";**

**}**

**?>**

**</div>**

**</div>**

**<div class="col-lg-6 col-md-6 col-sm-6 contact-w3-agile1" data-aos="flip-right">**

**<h4>Connect With Us</h4>**

**<p class="contact-agile1"><strong>Phone :</strong>+91 8108445916</p>**

**<p class="contact-agile1"><strong>Email :</strong> <a href="mailto:name@example.com">PMILIND92@GMAIL.COM</a></p>**

**<p class="contact-agile1"><strong>Address :</strong> New Thane Road, Maharashta, India</p>**

**<div class="social-bnr-agileits footer-icons-agileinfo">**

**<ul class="social-icons3">**

**<li><a href="#" class="fa fa-facebook icon-border facebook"> </a></li>**

**<li><a href="#" class="fa fa-twitter icon-border twitter"> </a></li>**

**<li><a href="#" class="fa fa-google-plus icon-border googleplus"> </a></li>**

**</ul>**

**</div>**

**<iframe src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d25347.005190213647!2d72.94931256213815!3d19.201540444505987!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x3be7b91d68959563%3A0x6f253b9168129506!2sHotel%20Tip%20Top%20Plaza!5e0!3m2!1sen!2sin!4v1581447698840!5m2!1sen!2sin" ></iframe>**

**</div>**

**<div class="clearfix"></div>**

**</div>**

**</section>**

**<!-- /contact -->**

**<div class="copy">**

**<p>© 2020 SUNRISE . All Rights Reserved | Design by <a href="index.php">SUNRISE</a></p>**

**</div>**

**<!--/footer -->**

**<!-- js -->**

**<script type="text/javascript" src="js/jquery-2.1.4.min.js"></script>**

**<!-- contact form -->**

**<script src="js/jqBootstrapValidation.js"></script>**

**<!-- /contact form -->**

**<!-- Calendar -->**

**<script src="js/jquery-ui.js"></script>**

**<script>**

**$(function() {**

**$( "#datepicker,#datepicker1,#datepicker2,#datepicker3" ).datepicker();**

**});**

**</script>**

**<!-- //Calendar -->**

**<!-- gallery popup -->**

**<link rel="stylesheet" href="css/swipebox.css">**

**<script src="js/jquery.swipebox.min.js"></script>**

**<script type="text/javascript">**

**jQuery(function($) {**

**$(".swipebox").swipebox();**

**});**

**</script>**

**<!-- //gallery popup -->**

**<!-- start-smoth-scrolling -->**

**<script type="text/javascript" src="js/move-top.js"></script>**

**<script type="text/javascript" src="js/easing.js"></script>**

**<script type="text/javascript">**

**jQuery(document).ready(function($) {**

**$(".scroll").click(function(event){**

**event.preventDefault();**

**$('html,body').animate({scrollTop:$(this.hash).offset().top},1000);**

**});**

**});**

**</script>**

**<!-- start-smoth-scrolling -->**

**<!-- flexSlider -->**

**<script defer src="js/jquery.flexslider.js"></script>**

**<script type="text/javascript">**

**$(window).load(function(){**

**$('.flexslider').flexslider({**

**animation: "slide",**

**start: function(slider){**

**$('body').removeClass('loading');**

**}**

**});**

**});**

**</script>**

**<!-- //flexSlider -->**

**<script src="js/responsiveslides.min.js"></script>**

**<script>**

**// You can also use "$(window).load(function() {"**

**$(function () {**

**// Slideshow 4**

**$("#slider4").responsiveSlides({**

**auto: true,**

**pager:true,**

**nav:false,**

**speed: 500,**

**namespace: "callbacks",**

**before: function () {**

**$('.events').append("<li>before event fired.</li>");**

**},**

**after: function () {**

**$('.events').append("<li>after event fired.</li>");**

**}**

**});**

**});**

**</script>**

**<!--search-bar-->**

**<script src="js/main.js"></script>**

**<!--//search-bar-->**

**<!--tabs-->**

**<script src="js/easy-responsive-tabs.js"></script>**

**<script>**

**$(document).ready(function () {**

**$('#horizontalTab').easyResponsiveTabs({**

**type: 'default', //Types: default, vertical, accordion**

**width: 'auto', //auto or any width like 600px**

**fit: true, // 100% fit in a container**

**closed: 'accordion', // Start closed if in accordion view**

**activate: function(event) { // Callback function if tab is switched**

**var $tab = $(this);**

**var $info = $('#tabInfo');**

**var $name = $('span', $info);**

**$name.text($tab.text());**

**$info.show();**

**}**

**});**

**$('#verticalTab').easyResponsiveTabs({**

**type: 'vertical',**

**width: 'auto',**

**fit: true**

**});**

**});**

**</script>**

**<!--//tabs-->**

**<!-- smooth scrolling -->**

**<script type="text/javascript">**

**$(document).ready(function() {**

**/\***

**var defaults = {**

**containerID: 'toTop', // fading element id**

**containerHoverID: 'toTopHover', // fading element hover id**

**scrollSpeed: 1200,**

**easingType: 'linear'**

**};**

**\*/**

**$().UItoTop({ easingType: 'easeOutQuart' });**

**});**

**</script>**

**<div class="arr-w3ls">**

**<a href="#home" id="toTop" style="display: block;"> <span id="toTopHover" style="opacity: 1;"> </span></a>**

**</div>**

**<!-- //smooth scrolling -->**

**<script type="text/javascript" src="js/bootstrap-3.1.1.min.js"></script>**

**</body>**

**</html>**

**Admin/index.php**

**<?php**

**session\_start();**

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

**{**

**header("location:home.php");**

**}**

**?>**

**<!DOCTYPE html>**

**<html >**

**<head>**

**<meta charset="UTF-8">**

**<title>SUN RISE 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" -->**

**<p><input type="submit" name="sub" value="Login"></p>**

**</fieldset>**

**</form>**

**</div> <!-- end login -->**

**<div class="bottom"> <h3><a href="../index.php">SUN RISE HOMEPAGE</a></h3></div>**

**</div>**

**</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>' ;**

**}**

**header("location:/index.php?");**

**}**

**?>**

# 9. CONCLUSION

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 in technology.

**This system is developed exclusively for the people of Thane. It provides facilities to the user with user friendly modules with sub modules. This system is developed in an understandable approach which can be easier to the layman of the computers. This system is developed totally GUI based and with smart links.**

# 9. BIBLIOGRAPHY

**System Analysis And Design**

- Kenneth E. Kendall, Julie E. Kendall

**An Analysis and Design of Information Systems**

- Grayce M. Booth

**Software Engineering**

- Roger S. Pressman

**Database Management System**

- James A. Larson

**PHP: A Beginner’s Guide**

- Kuwento Megosinarso

**Java Server Pages Programming: A visual professional guide for Design**

- Scott Mitchell