**VRINDAVAN SEVA SADAN**



**(THAKUR SHREE RADHA BIHARI CHARITABLE TRUST) RADHA MOHANVAN, CHHATTIKARA MARG, RAMAN RETI, SHRI DHAM VRINDAVAN(UP)**

***A Project Report submitted in partial fulfillment of the requirements for the award of the degree of***

# Bachelor of Computer Applications

## By

|  |  |
| --- | --- |
| **PRASHANT DEV UPADHYAY** | **(174200109)** |
| **PRASHANT SHARMA** | **(174200111)** |
| **PRATISHTHA DWIVEDI** | **(174200113)** |
| **PRERNA SHARMA** | **(174200114)** |
| **PUSPENDRA SINGH** | **(174200119)** |

Under the Guidance of

**Dr. Vinod Jain (Assistant Professor)** Department of Computer Engineering & Applications **Institute of Engineering & Technology**

**GLA University Mathura- 281406, INDIA**

**May, 2020**

**Department of Computer Engineering and Applications GLA University, Mathura**

**17 km. Stone NH#2, Mathura-Delhi Road,**

**P.O. – Chaumuhan, Mathura – 281406**

**U.P (India)**

**Declaration**

We hereby declare that the work which is being presented in the B.C.A. Project **“ VRINDAVAN SEVA SADAN (THAKUR SHREE RADHA BIHARI CHARITABLE TRUST) RADHA MOHANVAN, CHHATTIKARA MARG, RAMAN RETI, SHRI DHAM VRINDAVAN(UP) ”**, Impartial fulfillment of the requirements for the award of the ***Bachelor of Computer Applications*** and submitted to the Department of Computer Engineering and Applications of GLA University, Mathura, is an authentic record of our own work carried under the supervision of **Dr. Vinod Jain, Assistant Professor of Computer Engineering Department.**

The contents of this project report, in full or in parts, have not been submitted to any other institute or university for the award of any degree.

|  |  |  |
| --- | --- | --- |
| Prashant Dev  Upadhyay  (174200109) |  | Prashant Sharma (174200111) |
| Pratishtha Dwivedi (174200113) |  | Prerna Sharma  (174200114) |
| Pushpendra Singh(174200119) |  |  |

The current manual system uses paperwork and direct human language communication by mouth to manage the hotel. This delays information transmission in the hotel. Booking is done through phone calls or through visit to the hotel booking office. The guest’s personal details such as Name, Age, Nationality, and Duration of stay, are input during booking in. The booking office orders for preparation of the guest’s room before his/ her check in date. The documents are transferred manually to the filling department for compilation of the guest’s file. On the reporting date the file is transferred to the reception. On checking in the guest is given the key to his allocated room, he also specify if he needs room service. The receptionist hands over the guest’s file to the accountant on the next table. Here the guest pays accommodation and meals fee. The guest’s file is updated on daily basis of his expenditure costs. The accounts department generates the bills on daily basis and delivered to the guests in their rooms at dusk by the service maids. The guest pays at the accounts desk, where the receipts are generated. For a one meal customer the bill is generated immediately after ordering and he pays at the accountant desk before leaving. During checking out of guests, their expenditure outlines are generated a day before check out date. The guests receive their outlines at the accounts desk as they check out, where they pay for bills balances if any.

|  |  |
| --- | --- |
| **Topic** | **Pg. no.** |
| List of Tables | Vii |
| List of Figures | Viii |
| Chapter1 Introduction | 1 |
| 1.1 Overview &Motivation | 1 |
| 1.2 Objectives | 2 |
| 1.3 Organization of Project Report | 2 |
| Chapter2 Software Requirement Analysis | 3 |
| 2.1 Requirement Analysis | 3 |
| 2.1.1 Hardware Requirement | 4 |
| 2.1.2 Software Requirement | 4 |
| 2.1.3 Tools and Technologies | 4 |
| 2.2 Feasibility Study | 5 |
| 2.2.1 Technical Feasibility | 5 |
| 2.2.2 Operational Feasibility | 6 |
| 2.2.3 Economical Feasibility | 7 |
| 2.3 Analysis | 7 |
| 2.4 Summary of Modules | 8 |
| Chapter3 Software Design | 9 |
| 3.1 DFD | 10 |
| DFD Level 0 | 11 |
| DFD Level 1 | 11 |
|  |  |
| 3.2 ER | 12 |
| ER Diagram | 13 |
| 3.3 Database Design | 14 |
| Chapter 4 Implementation & User Interface | 17 |
|  |  |
|  |  |

|  |  |
| --- | --- |
| Chapter 5 Software Testing | 22 |
| 5.1 Testing | 22 |
| 5.2 Objectives of Software Testing | 22 |
| 5.3 Principles of Software Testing | 23 |
| 5.3.1 White Box Testing | 23 |
| 5.3.2 Black Box Testing | 23 |
| 5.4 Testing Fundamentals | 24 |
| 5.5 Testing Information | 24 |
| Conclusion | 25 |
| References | 26 |

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Table Name** | **Table No.** | **Page** |
| **1** | Admin Login Table | Table 3.1 | 15 |
| **2** | User Login Table | Table 3.2 | 15 |
| **3** | Employee Login Table | Table 3.3 | 16 |
| **4** | Contact Us Table | Table 3.4 | 16 |
| **5** | Reply Message table | Table 3.5 | 17 |
| **6** | Room Table | Table 3.6 | 17 |
| **7** | Food Table   1. Breakfast Table 2. Lunch Table 3. Dinner Table 4. Full Meal | Table 3.7  Table 3.8  Table 3.9  Table 3.10 | 20  21  22  23 |
| **8** | Employee Leave | Table 3.11 | 24 |
| **9** | Room Booking Table | Table 3.12 | 25 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

/

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Figure Name** | **Fig.No.** | **Page** |
| **1** | Data Flow Diagram Symbols | Fig 3.1 | 10 |
| **2** | 0 level DFD | Fig 3.1.1 | 11 |
| **3** | 1 level DFD- Administrator | Fig 3.1.2 | 12 |
| **4** | Entity Relationship Diagram | Fig 3.2 | 13 |
| **5** | Database tables | Fig 3.3 | 15 |
| **6** | Screenshots | Fig 4.1 | 19 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# INTRODUCTION

**Present Problem Statement:**

The current manual system uses paperwork and direct human language communication by mouth to manage the hotel. This delays information transmission in the hotel. Booking is done through phone calls or through visit to the hotel booking office. The guest’s personal details such as Name, Age, Nationality, and Duration of stay, are input during booking in. The booking office orders for preparation of the guest’s room before his/ her check in date. The documents are transferred manually to the filling department for compilation of the guest’s file. On the reporting date the file is transferred to the reception. On checking in the guest is given the key to his allocated room, he also specify if he needs room service. The receptionist hands over the guest’s file to the accountant on the next table. Here the guest pays accommodation and meals fee. The guest’s file is updated on daily basis of his expenditure costs. The accounts department generates the bills on daily basis and delivered to the guests in their rooms at dusk by the service maids. The guest pays at the accounts desk, where the receipts are generated. For a one meal customer the bill is generated immediately after ordering and he pays at the accountant desk before leaving. During checking out of guests, their expenditure outlines are generated a day before check out date. The guests receive their outlines at the accounts desk as they check out, where they pay for bills balances if any.

**Proposed in current System:**

* Difficulty in location of guest files
* Large storage space
* Human and computational errors
* Poorly generated records
* Complains from guests
* Poor communication
* Difficulty in data analysis

## Disadvantage of current system:

* Guest files can easily get lost or mix up with other guest file documents.
* Files occupy a large storage space.
* Unnecessary duplication of data.
* Files are prone to theft unauthorized modification due to low data security levels and standards.
* Due to easy access to guest data by unauthorized users, guest data is extremely unconfident.
* Retrieval of guest records is extremely difficult.
* Data entry procedure is prone to errors.
* Guest records are extremely difficult to modify since modification generates dirty and unpresentable reports.

#### Chapter1 Introduction

## Proposed System:

To avoid the problems with Manual System hotel decide to implement an automatic computerized

and advanced system.

* To enable online booking via the internet.
* To enable automated data entry methods.
* Ensure efficient and reliable communication within the hotel.
* Avoid data entry errors by use of input masks.
* Enable easy authorized modification of data.
* Enforce security measures to avoid unauthorized access to guest records.
* Enable fast and easy retrieval of guest records and data for fast reference activities.

## Organization of Project Report:

|  |  |
| --- | --- |
| **PHASES** | **TIME DURATION** |
| Software requirement specification | 2 weeks |
| System design | 3 weeks |
| Coding | 5 weeks |
| Testing | 2 weeks |
| Documentation | 2 weeks |
| Implementation | 1 weeks |

**SOFTWARE REQUIREMENT ANALYSIS**

System Analysis is a detailed study of the various operations performed by a system and their relationship within and outside the system. It is a systematic technique that defines goals and objectives the goal of the development is to deliver the system in the line with the user’s requirements, and analysis is this process.

System study has been conducted with the following objectives in mind: -

* + - Identify the client’s need.
    - Evaluate the system concept for feasibility.
    - Perform economical and technical analysis.
    - Allocate functional to hardware, software, people, database and other system elements
    - Establish cost and schedule constraints.
    - Both hardware and software expertise is required to successfully attain the objectives.

## Requirement Analysis

Information gathering is usually the first phase of the software development project. The purpose of this phase is to identify and document the exact requirements for the system. The user’s request identifies the need for a new information system and on investigation re-defined the new problem to be based on MIS, which supports management. The objective is to determine whether the request is valid and feasible before a recommendation is made to build a new or existing manual system continue

The major steps are –

* + - Defining the user requirements.
    - Studying the present system to verify the problem.
    - Defining the performance expected by the candidate to user equirement

#### Hardware Requirements

Processor : Intel Dual Core or More Processor Speed : 1.5 GHZ

RAM : 2 GB

Hard Disk : 2 GB of free space

**Internet connection : Required**

#### Software Requirements

Operating System: Window 7 and higher

Front End: HTML, CSS, Java Script, jQuery, Bootstrap

Back End: SQL Server, PHP

Text Editor: Sublime text, Notepad++, Visual Studio Code.

Server: XAMPP Server (Cross-platform, Apache, MariaDB (MySQL), PHP and Perl)

#### Tools and Technology Tools:

* + - * Windows 7 & and higher
      * Notepad++
      * Visual Studio Code
      * SQL Server
      * XAMPP Server (Cross-platform, Apache, MariaDB (MySQL), PHP and Perl)

#### Technology:

* **HTML5:** Hypertext Markup Language is the standard markup language for creating web pages and web application. HTML element are the building blocks of HTML pages. With HTML constructs, image and other objects, such as interactive form.
* **CSS3: C**ascading **S**tyle **S**heets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable. CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.
* **JavaScript:** JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.
* **jQuery**: jQuery is a small, light-weight and fast JavaScript library. It is cross-platform and supports different types of browsers. It is also referred as? write less do more? because it takes a lot of common tasks that requires many lines of JavaScript code to accomplish, and binds them into methods that can be called with a single line of code whenever needed. It is also very useful to simplify a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.
* **Bootstrap 4**: Bootstrap is the most popular HTML, CSS and JavaScript framework for developing a responsive and mobile friendly website. It is absolutely free to download and use. It is a front-end framework used for easier and faster web development. It includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many others. It can also use JavaScript plug-ins. It facilitates you to create responsive designs.
* **SQL:** SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in a relational database. SQL is the standard language for Relational Database System. All the Relational Database Management Systems (RDMS) like MySQL, MS Access, Oracle, Sybase, Informix, Postgres and SQL Server use SQL as their standard database language.
* **PHP:** PHP is a server-side scripting language. that is used to develop Static websites or Dynamic websites or Web applications. PHP stands for Hypertext Pre-processor, that earlier stood for Personal Home Pages. PHP scripts can only be interpreted on a server that has PHP installed. The client computers accessing the PHP scripts require a web browser only. A PHP file contains PHP tags and ends with the extension ".php".

## Feasibility Study

Feasibility study is the process of determination of whether or not a project is worth doing. Feasibility studies are undertaken within tight time constraints and normally culminateina written and oral feasibility report .I have taken a fixed time in feasibility study with my co-developer. The contents and recommendations of this feasibility study helped us as a sound basis for deciding how to precede the project. It helped in taking decisions such as which software to use, hardware combinations,etc.

#### Technical feasibility:

This is concerned with specifying equipment of software and hardware that will successfully satisfy the user requirements. The technical needs of the system may vary considerably, but might include:

* The facility to produce output in a given time.
* Response time under certain condition.
* Ability to produce a certain volume of transaction at a particular speed.
* In examining technical feasibility, configuration of the system is given more importance than the actual make of hardware. The configuration should give the complete picture about the system requirements. What speeds of input and output should be achieved at particular quality of printing.

According to the definition of technical feasibility the compatibility between front-end and back-end is very important. In our project the compatibility of both is very good. The degree of compatibility of PHP and SQL Server 2014 is very good. The speed of output is very good when we enter the data and click button then the response time is very fast and give result very quick. In ever find difficulty when we use complex query or heavy transaction. The speed of transaction is always smooth and constant. This software provides facility to communicate data to distant location.

We use Active Server Pages and JavaScript. The designing of front-end of any project is very important so we selected Active Server Pages, HTML & CSS as front-end due to following reason:

* + Easy implementation of code.
  + Well define interface and database.
  + Well define hand shaking of SQL Server2014

At present scenario the no of backend are available but I have selected SQL Server 2008 because of the following number of reasons.

* + - Able to handle large data.
    - Security.
    - Robust RDBMS
    - Backup &Recovery

With the help of above support we remove defect of existing software. In future we can easily switch over any platform. To ensure that system does not halt in case of undesired situation or events. Problem effected of any module does not affect any module of the system. A change of hardware does not produce problem.

#### Operational Feasibility:

It is mainly related to human organizational and political aspects. The points to be considered are:

* + - * What changes will be brought with the system?
      * What organization structures are distributed structures are distributed.
      * What new skills will be required? Do the existing staff members have these skills? If not, can they be trained in due course of time?

At present stage all the work is done manually. So, throughput and response time is too much. Major problem is lack of security check that should have been applied.

Finding out the detail regarding users request was very difficult, because data store was in different registers and different places. In case of any problem, no one can solve the problem until the person responsible is not present.

Current communication is entirely on telephonic conversation or personal meetings. Post computerization staff can interact using internet.

Now, we will explain the last point of operational feasibility i.e. handling and keeping of software , at every point of designing I will take care that menu options are not too complex and can be easily learned and required least amount of technical skills as operators are going to be from non-computers background.

#### Economic feasibility:

Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system. More commonly known as cost/benefit analysis: the procedure is to determine the benefits and saving that are expected from a proposed system and compare them with cost. If benefits out weights cost, a decision is taken to design and implement the system. Otherwise, further justification or alternative in the proposed system will have to be made if it is to have a chance of being approved. This is an ongoing effort that improves in accuracy at each phase of the system life cycle.

At present Company has ten systems with following configuration:

* + - * Ram 4 GB or above for fast execution and reliability
      * MOTHER Board x64 based PC
      * Color Monitor 14” and17”
      * Hard Disk 100GB
      * Hence the economic feasibility is very good.

## Analysis

System analysis is the first step towards the software building process. The purpose of system analysis is to understand the system requirements, identify the data, functional and behavioral requirements and building the models of the system for better understanding of the system.

In the process of system analysis one should first understand that, what the present system is, is how it works (i.e. processes).After analyzing these points we become able to identify the problems in the present system. Upon evaluating current problems and desired information (input and output to the system), the analyst looks towards one or more solutions. To begin with, the data objects, processing functions, and behavior of the system are defined in detail. After this models, from three different aspects of the system-data, function and behavior. The models created during the system analysis process helps in better understanding of data and control flow, functional processing, operational behavioral and information content.

## Summary of Modules

#### Administrator

The administrator is responsible for maintaining Database of web portal. This module will update information of all database related to the hotel. Admin can access the data and can modify it.

#### Guest

Guest have first log in the website where he or she has enter the name, contact email id etc. After that it can access the database /of booking, room, transport, meal, sporting.

#### Sporting

Sporting the different games that are available in the hotel.

#### Meal

It will include all different item of foods that are available in the hotel.

**Room**

It will include all different item of foods that are available in the hotel.

**Employee**

It will include all employee that are working in the hotel.

**Transport**

It will include all transport medium that are available in the hotel.

# SOFTWARE DESIGN

A software design document (SDD) is a written description of a [software](http://en.wikipedia.org/wiki/Software) product, that a software designer writes in order to give a [software development](http://en.wikipedia.org/wiki/Software_development) team overall guidance to the architecture of the software project. An SDD usually accompanies an architecture diagram with pointers to detailed feature specifications of smaller pieces of the design. Practically, a design document is required to coordinate a large team under a single vision. A design document needs to be a stable reference, outlining all parts of the software and how they will work. The document is commanded to give a fairly complete description, while maintaining a high-level view of the software.

There are two kinds of design documents called HLDD (high-level design document) and LLDD (low-level design document).

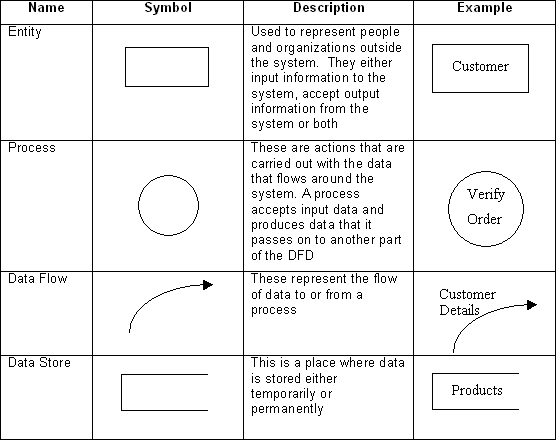
The SDD contains the following documents:

* 1. The [**data design**](http://en.wikipedia.org/wiki/Data-driven_design) describes structures that reside within the software. Attributes and relationships between [data objects](http://en.wikipedia.org/wiki/Data_object) dictate the choice of [data structures](http://en.wikipedia.org/wiki/Data_structures).
  2. The [**architecture design**](http://en.wikipedia.org/wiki/Software_architecture) uses information flowing characteristics, and maps them into the program structure. The transformation mapping method is applied to exhibit distinct boundaries between incoming and outgoing data. The data flow diagrams allocate control input, processing and output along three separate modules.
  3. The [**interface design**](http://en.wikipedia.org/wiki/Interface_design) describes internal and external program interfaces, as well as the design of human interface. Internal and external interface designs are based on the information obtained from the analys is model.
  4. The [**procedural design**](http://en.wikipedia.org/wiki/Procedural_design) describes structured programming concepts using graphical, tabular and textual notations. These design mediums enable the designer to represent procedural detail that facilitates translation to code. This blueprint for implementation forms the basis for all subsequent software engineering worked.

## Data Flow Diagram (DFD)

The Data Flow Diagram (DFD) is a graphical representation of the flow of data through an information system. It enables you to represent the processes in your information system from the viewpoint of data. The DFD lets you visualize how the system operates, what the system accomplishes and how it will be implemented, when it is refined with further specification.

* Data flow diagrams are used by systems analysts to design information-processing systems but also as a way to model whole organizations. You build a DFD at the very beginning of your business process modeling in order to model the functions your system has to carry out and the interaction between those functions together with focusing on data exchanges between processes. You can associate data with conceptual, logical, and physical data models and object-oriented models.



#### Fig 3.1: Data Flow Diagram Symbols

* + 1. **DFD LEVEL0**

**Fig 3.2: 0 Level DFD**

* + 1. **DFD LEVEL 1**

**Fig 3.3: 1 Level DFD**

In 0 level DFD, all type of user can fill their login details to get in MUSKAAN (project) and then all of them can access the info about the missing and founded children.

In 1.1 level DFD, the admin can login by providing the valid username and password. After that admin can create id and password for the policemen and also creates reports.

## Entity Relationship Diagram(ER-Diagram)

An entity-relationship diagram (ERD) is a graphical representation of an information system that shows the relationship between people, objects, places, concepts or events withinthatsystem.AnERDisa[datamodeling](http://searchdatamanagement.techtarget.com/definition/data-modeling)techniquethatcanhelpdefinebusiness processes and can be used as the foundation for a [relational database](http://searchsqlserver.techtarget.com/definition/relational-database).

While useful for organizing [data](http://searchdatamanagement.techtarget.com/definition/data) that can be represented by a relational structure, an entity-relationship diagram can't sufficiently represent semi-structured or [unstructured](http://searchbusinessanalytics.techtarget.com/definition/unstructured-data) [data,](http://searchbusinessanalytics.techtarget.com/definition/unstructured-data) and an ERD is unlikely to be helpful on its own in integrating data into a pre existing information system.

Three main components of an ERD are the [entities](http://whatis.techtarget.com/definition/entity), which are objects or concepts that can have data stored about them, the relationship between those entities, and the [cardinality,](http://whatis.techtarget.com/definition/cardinality) which defines that relationship in terms of numbers.

#### Components of the ER Model

The three main components of the ER Model are **entities**, **attributes** and

#### relationships.

* In ERM terms, an entity is a "thing" within the organization that we want to keep information about, such as a customer, employee or course. In other words, an entity in an ERM actually refers to a table, and rows within the table are referred to as entity occurrences. Entities are represented by rectangles containing the name of the entity. Entity names must be singular and incapital letters.
* Each entity has attributes which are the properties of each entity. Attributes will be implemented as columns in the tables. Each attribute has a domain which specifies the set of possible values an attribute can have. For instance, the range of values for a telephone extension may be specified as a set of integer numbers between 4000 and 4999. An attributes domain is not displayed in ER diagrams, but is recorded in the data dictionary.
* Attributes can be of various types. Acomposite attribute can be sub divided into smaller parts. For example, an attribute Name can be subdivided into First Name and Last Name. Attributes that cannot be subdivided are called simple attributes. First Name and Last Name are now simple attributes. Most attributes have only a single value and as such are called single valued attributes. For example, a Teacher can have only one Last Name or a Subject can have only one Subject Code. Multivalued attributes can have more than one value. For example, a Student could have more than one Certificate or a Department may have several Extensions.
* A key attribute is an attribute that has a unique value for each entity occurrence. In other words, a key attribute is used to identify each row uniquely. For example, a Subject Code will uniquely identify each subject as no two subjects can have the same Subject Code. Key attributes are represented by underlining its name.
* A relationship is the association between entities or entity occurrences

#### 3.2.1 ER Diagram of VRINDAVAN SEVA SADAN (Thakur Shree Radha Bihari Charitable Trust) Radha Mohanvan, Chhattikara Marg, Raman Reti, Shri Dham Vrindavan(UP):

**Fig 3.2: ER Daigram**

**3.3 Database Design**

A good database design is crucial for a high-performance application, just as an Aerodynamic body is important to a race car. If the car doesn’t have smooth lines, it will produce drag and go slower. Without optimized relationships, your database won’t perform as efficiently as possible. Thinking about relationships and database efficiency is part of normalization.

Beyond the issue of performance is the issue of maintenance your database should be easy to maintain. This includes storing only a limited amount (if any) of repetitive data. If you have a lot of repetitive data and one instance of that data undergoes a change (such as a name change), that change has to be made for all occurrences of the data. To eliminate duplication and enhance your ability to maintain the data, you might create a table of possible values and use a key to refer to the value. That way, if the value changes names, the change occurs only on master table. The reference remains the same throughout other tables.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Description** | **Constraints** |
| ID | Int (255) | ID of admin | Primary Key |
| Username | varchar (255) | Username of admin | Not Null |
| Password | Varchar (255) | Password of admin | Not Null |

#### Table 3.1: Admin Login

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Description** | **Constraints** |
| U\_id | Int (255) | ID of user | Primary Key |
| Name | Varchar (255) | Name of user | Not Null |
| Username | Varchar (255) | Username of user | Not Null |
| Phone | Varchar (255) | Phone No. of user | Not Null |
| Aadhar | Varchar (255) | Unique Aadhar No. | Not Null |
| Email | Varchar (255) | User email address | Not Null |
| Password | Varchar (255) | Password of user | Not Null |
| Role | Varchar (255) | Always user | Not Null |
| U\_img | Varchar (255) | User profile img | Not Null |

**Table 3.2: User Login**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Description** | **Constraints** |
| Emp\_id | Int (255) | Id of emp | Primary Key |
| Emp\_name | Varchar (255) | Name of emp | Not Null |
| Emp\_username | Varchar (255) | Username of emp | Not Null |
| Emp\_phone | Varchar (255) | Phone no. of emp | Not Null |
| Emp\_email | Varchar (255) | Email of emp | Not Null |
| Emp\_address | Varchar (255) | Address of emp | Not Null |
| Emp\_city | Varchar (255) | City of emp | Not Null |
| Emp\_state | Varchar (255) | State of emp | Not Null |
| Emp\_dept | Varchar (255) | Department of emp | Not Null |
| Emp\_doj | Varchar (255) | Date of joining emp | Not Null |
| Emp\_dob | Varchar (255) | Date of birth of emp | Not Null |
| Emp\_password | Varchar (255) | Password of emp | Not Null |

**Table 3.3: Employee Login**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Description** | **Constraints** |
| Con\_id | Int (255) | id of contact msg | Primary Key |
| Con\_user\_name | Varchar (255) | Name of user | Not Null |
| Con\_user\_email | Varchar (255) | Email of user | Not null |
| Con\_user\_phone | Varchar (255) | Phone of user | Not null |
| Con\_user\_sbj | Varchar (255) | Subject of question | Not null |
| Con\_user\_msg | Text | Complete message | Not null |
| Con\_msg\_date | Varchar (255) | Message date | Not Null |
| U\_id | Int (255) | Id of user | Not Null |

**Table 3.4: Contact US**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Description** | **Constraints** |
| mess\_id | Int (255) | id of message | Primary Key |
| U\_id | Int (255) | Id of user | Not Null |
| Con\_id | Int (255) | Id of contact | Not null |
| Message | Text | Complete message | Not null |

**Table 3.5: Reply Message**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Description** | **Constraints** |
| Room\_id | Int(255) | id of leave | Primary Key |
| Room\_number | Varchar (255) | Room number | Not Null |
| Room\_type | Varchar (255) | Type of Room | Not null |
| Room\_bed | Varchar (255) | Bed in the Room | Not null |
| Room\_size | Varchar (255) | Size of Room | Not null |
| Room\_extra | Varchar (255) | Extra facility of Room | Not null |
| Room\_abour | Varchar (255) | Something about Room | Not Null |
| Room\_price | Varchar (255) | Price of Room | Not Null |
| Room\_img | Varchar (255) | Image of Room | Not Null |

**Table 3.6: Rooms**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Description** | **Constraints** |
| Breakst\_id | Int(255) | id of breakfast | Primary Key |
| Breakst\_name | Varchar (255) | Name of breakfast | Not Null |
| Breakst\_size | Varchar (255) | Size of breakfast | Not null |
| Breakst\_price | Varchar (255) | Price of breakfast | Not null |

**Table 3.7: Breakfast**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Description** | **Constraints** |
| Lunch\_id | Int(255) | id of lunch | Primary Key |
| Lunch\_name | Varchar (255) | Name of lunch | Not Null |
| Lunch\_size | Varchar (255) | Size of lunch | Not null |
| Lunch\_price | Varchar (255) | Price of lunch | Not null |

**Table 3.8: Lunck**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Description** | **Constraints** |
| Dinner\_id | Int(255) | id of dinner | Primary Key |
| Dinner\_name | Varchar (255) | Name of dinner | Not Null |
| Dinner\_size | Varchar (255) | Size of dinner | Not null |
| Dinner\_price | Varchar (255) | Price of dinner | Not null |

**Table 3.9: Dinner**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Description** | **Constraints** |
| Fullmeal\_id | Int(255) | id of fullmeal | Primary Key |
| Fullmeal\_name | Varchar (255) | Name of fullmeal | Not Null |
| Fullmeal\_size | Varchar (255) | Size of fullmeal | Not null |
| Fullmeal\_price | Varchar (255) | Price of fullmeal | Not null |

**Table 3.10: Fullmeal**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Description** | **Constraints** |
| Id | Int(255) | id of leave | Primary Key |
| Emp\_id | Int (255) | Id of emp | Not Null |
| Emp\_name | Varchar (255) | Name of emp | Not null |
| Emp\_phone | Varchar (255) | Phone of emp | Not null |
| Emp\_email | Varchar (255) | Email of emp | Not null |
| Start\_date | Varchar (255) | Leave start date | Not null |
| End\_date | Varchar (255) | Leave end date | Not Null |
| Emp\_reason | Varchar (255) | Reason of leave | Not Null |
| Emp\_explain | Text | - |  |

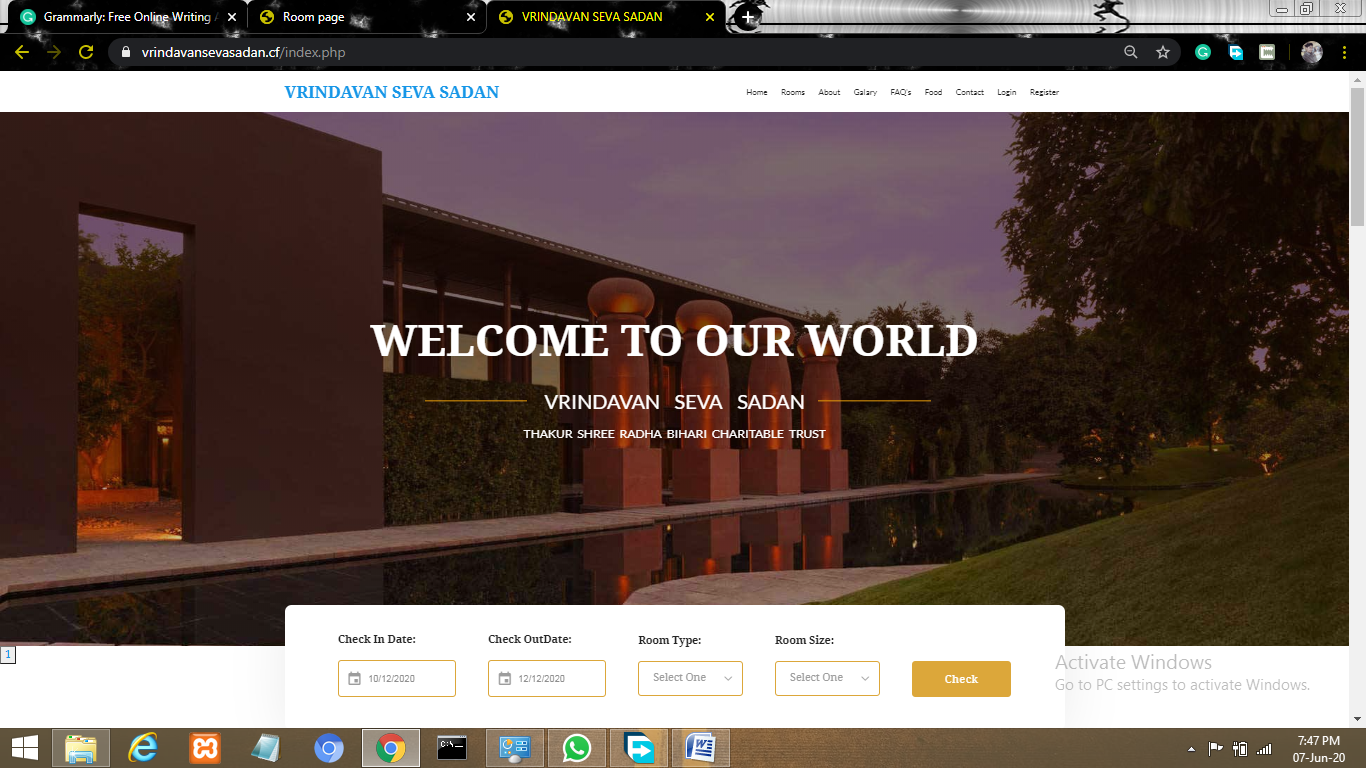
**Table 3.11: Employee Leave Application**

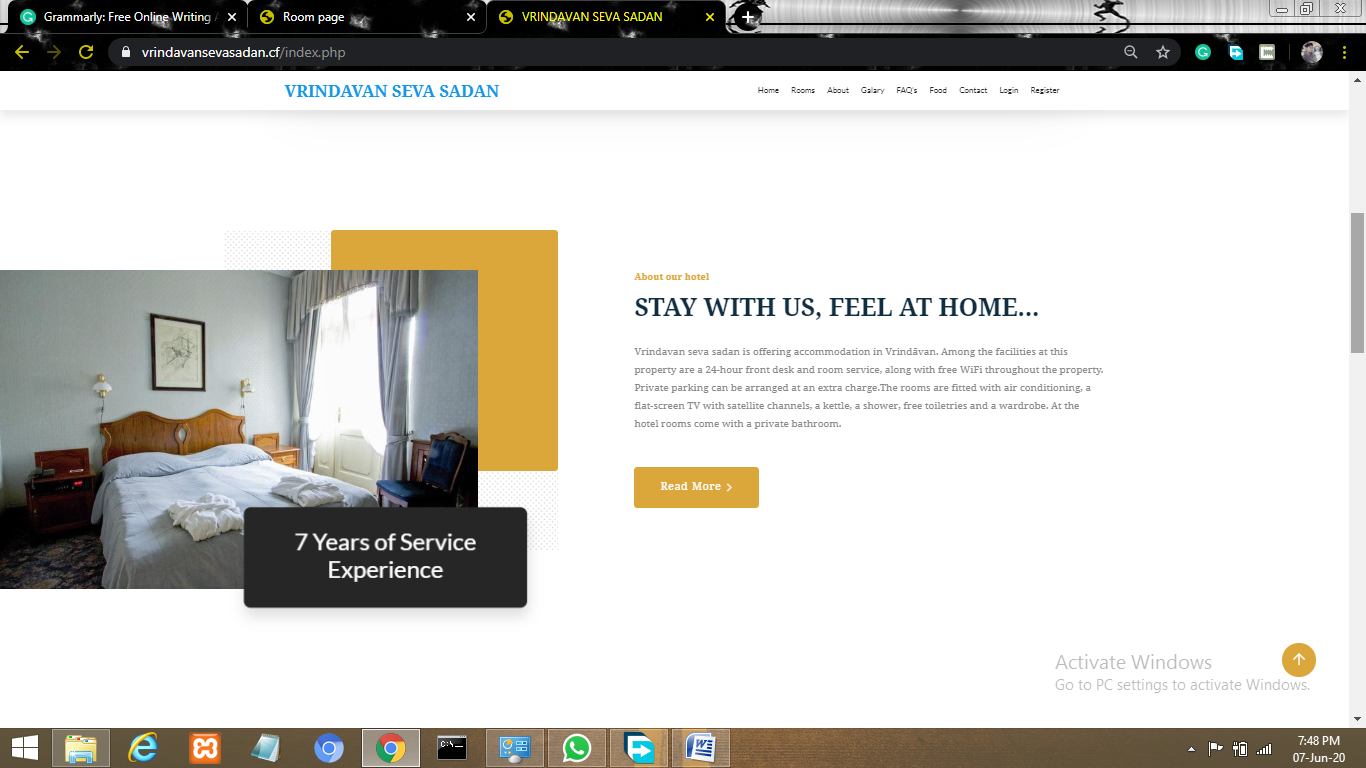
|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Description** | **Constraints** |
| Room\_id | Int(255) | Id of Room | Primary Key |
| Title | Varchar (255) | Title of room | Not Null |
| User\_name | Varchar (255) | Username of user | Not Null |
| Email | Varchar (255) | Email of user | Not Null |
| Phone | Varchar (255) | Phone of user | Not Null |
| State | Varchar (255) | State of user | Not Null |
| City | Varchar (255) | City of user | Not Null |
| Cin | Varchar (255) | Check in date | Not Null |
| Cout | Varchar (255) | Check out date | Not Null |
| Meal | Varchar (255) | Food meal | Not Null |
| No. of days | Int (255) | Number of days | Not Null |

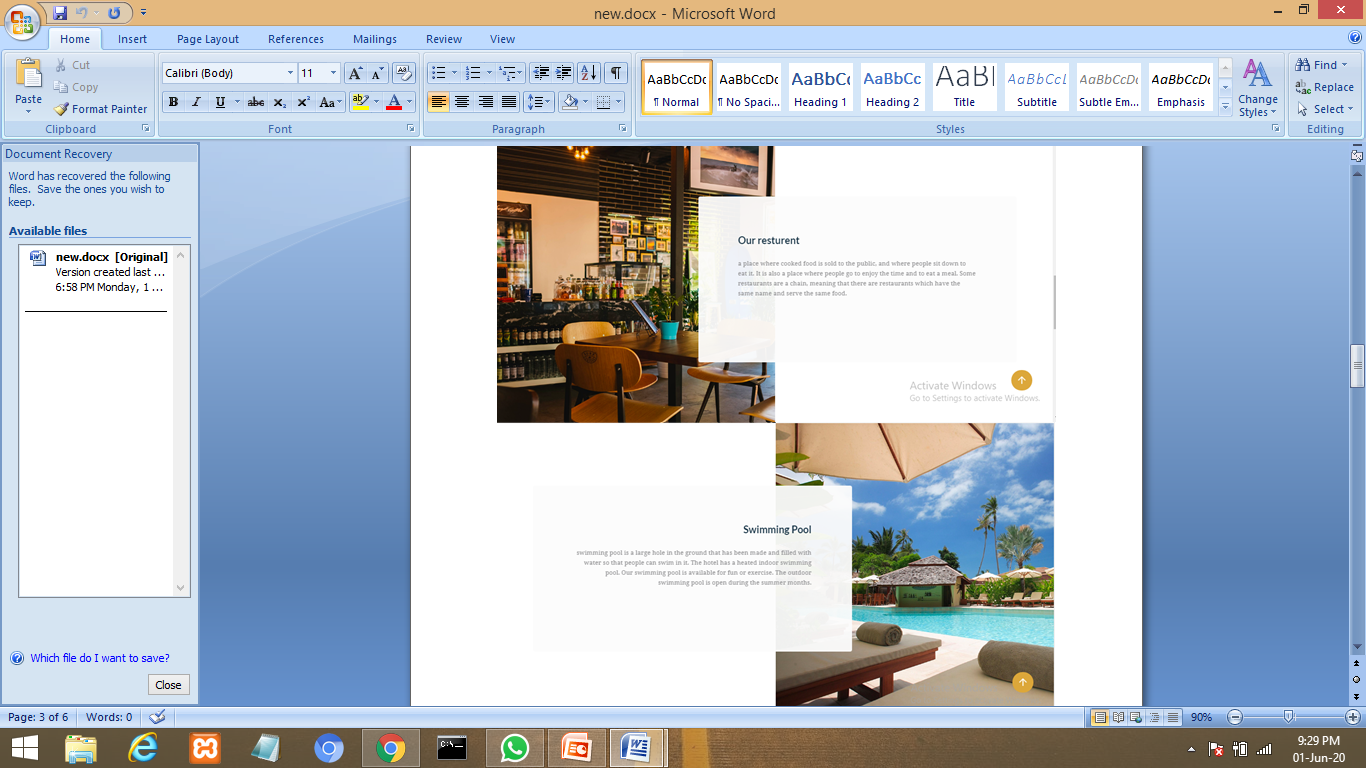
**Table 3.12: Room Bookings**

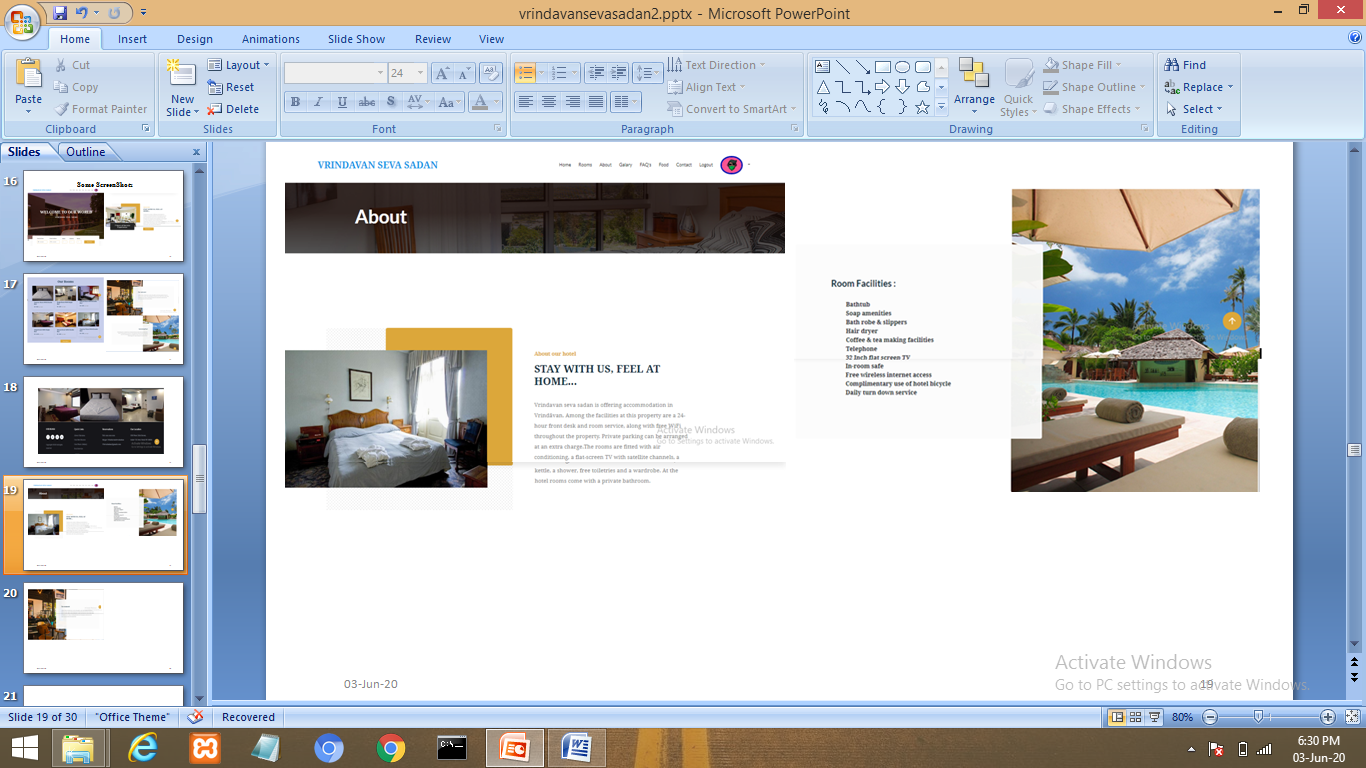
17

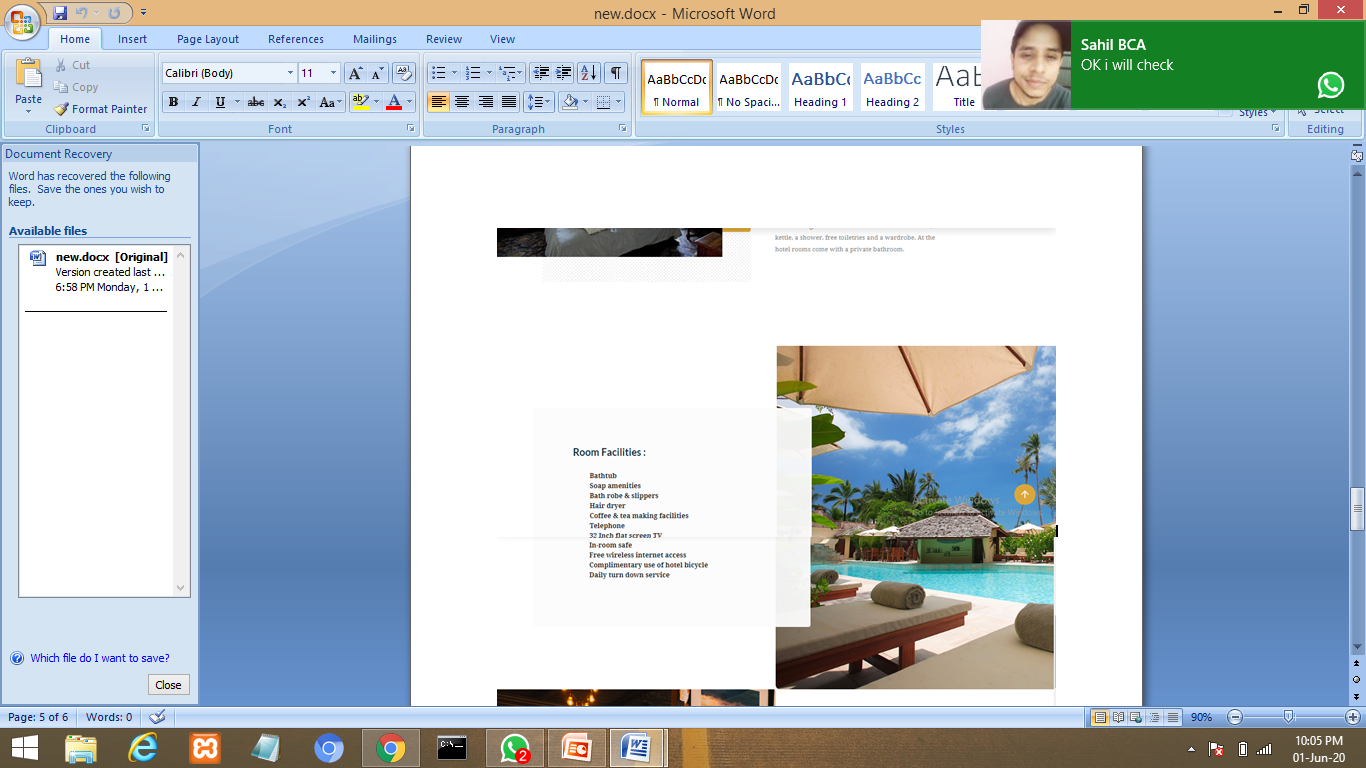
**IMPLEMENTATION &USER INTERFACE**

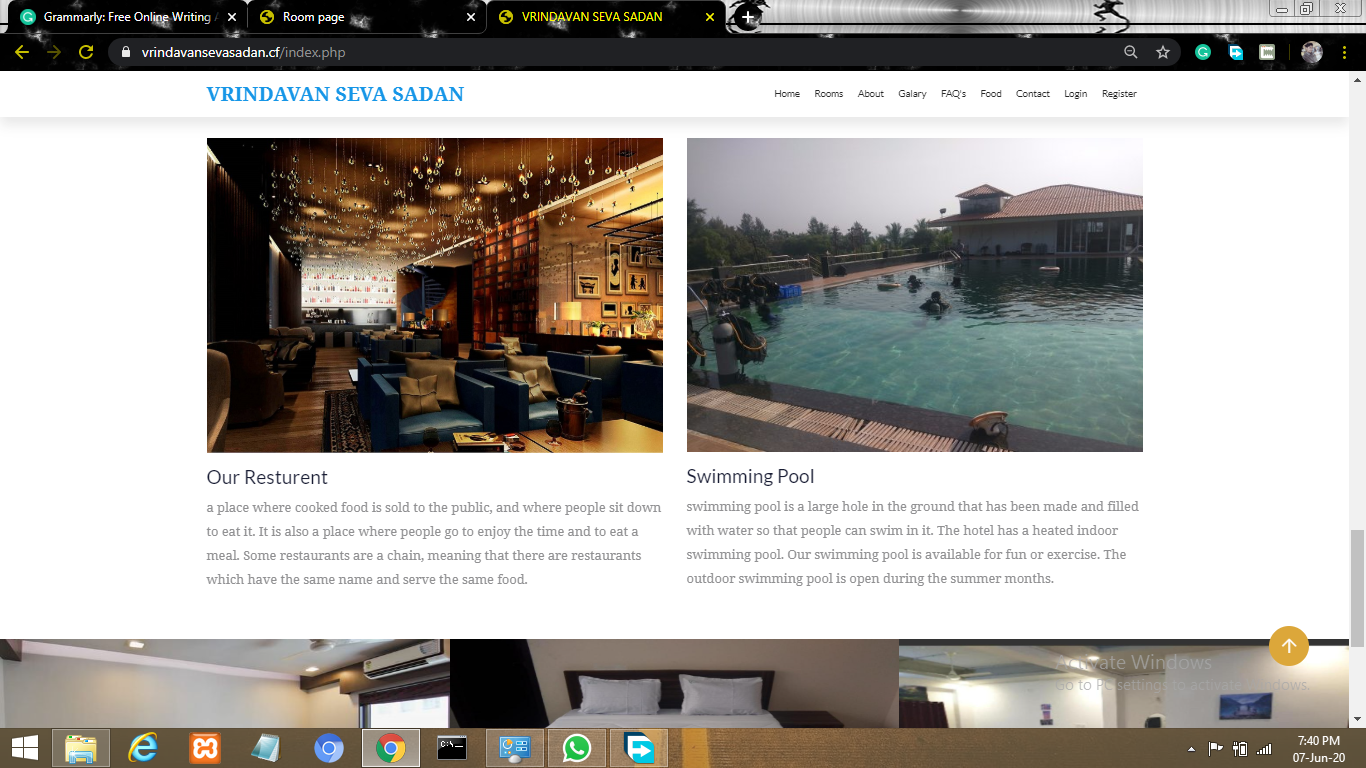
****

****

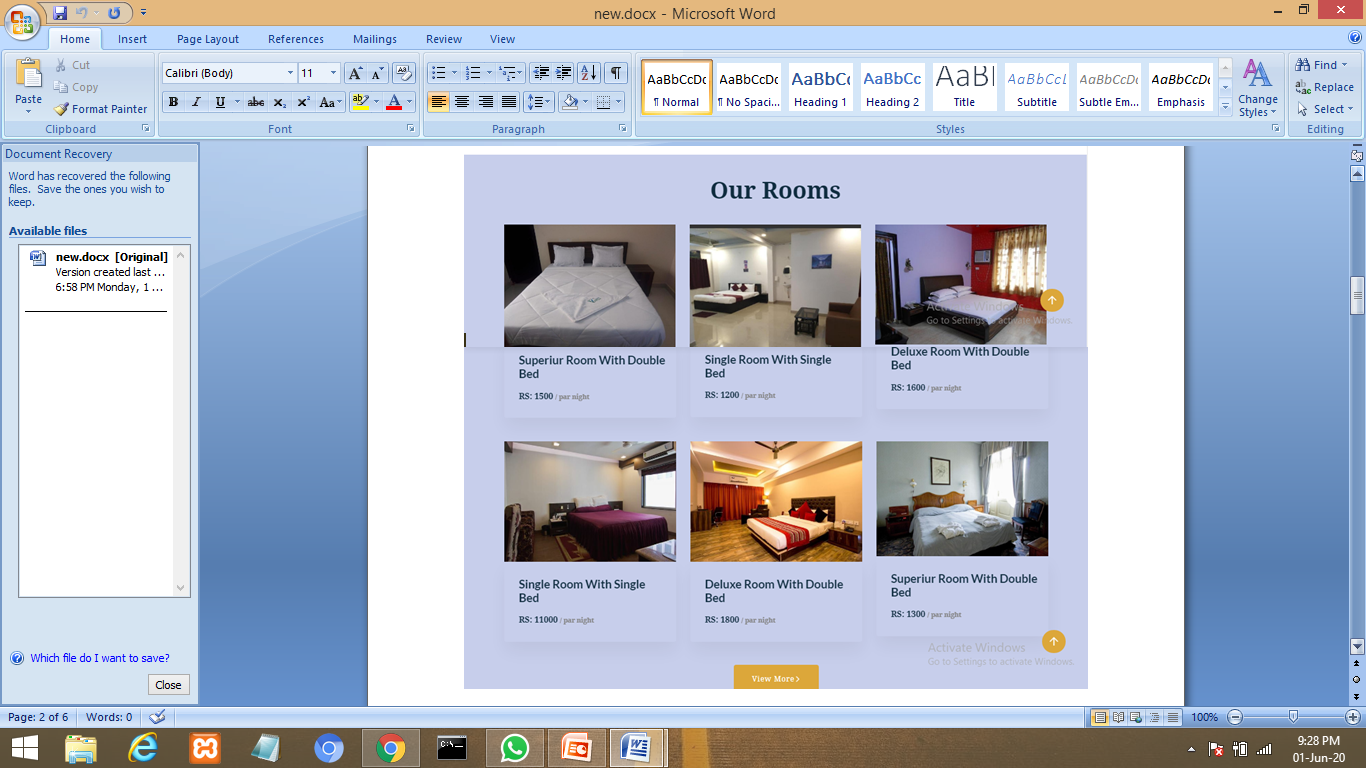
****

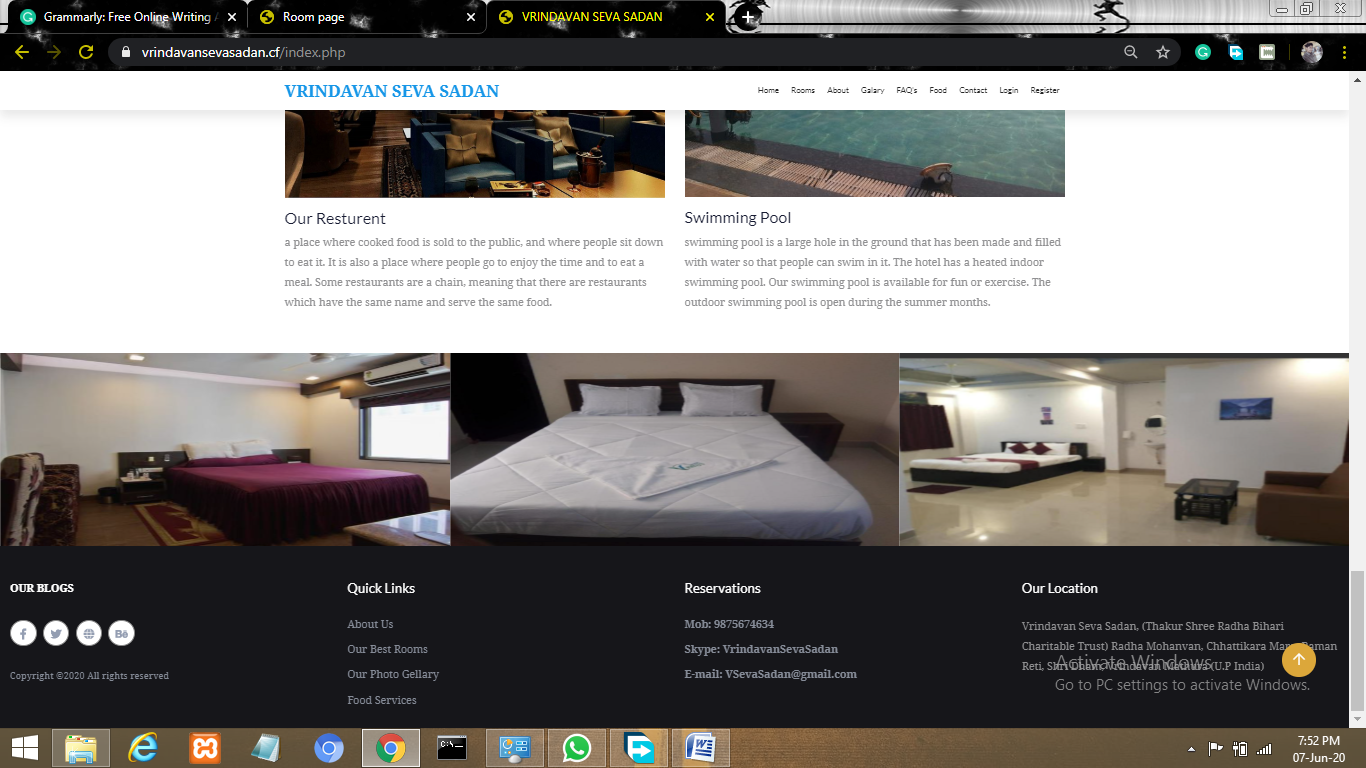
****

****

****

**Fig 4.1: Home Page**

****

****

**Fig 4.2: Room Page**

**Chapter 4**

**Implementation & User Interface**

**SOFTWARE TESTING**

21

## Testing

* + - Software testing is the process of executing a program with intension of finding errors in the code. It is a process of evolution of system or its parts by manual or automatic means to verify that it is satisfying specified or requirements or not.
    - Generally, no system is perfect due to communication problems between user and developer, time constraints, or conceptual mistakes by developer.
    - To purpose of system testing is to check and find out these errors or faults as early as possible so losses due to it can be saved.
    - Testing is the fundamental process of software success.
    - Testing is not a distinct phase in system development life cycle but should be applicable throughout all phases i.e. design development and maintenance phase.
    - Testing is used to show incorrectness and considered to success when an error is detected.

## Objectives of Software Testing

* + - **Software Quality Improvement:** The computer and the software are mainly used for complex and critical applications and a bug or fault in software causes severe losses. So a great consideration is required for checking for quality of software.

#### Verification And Validation:

* + - * Verification means to test that we are building the product in right way .i.e. are we using the correct procedure for the development of software so that it can meet the user requirements.
      * Validation means to check whether we are building the right product or not.

22

**Chapter 5**

**Software Testing**

* + - **Software Reliability Estimation:** The objective is to discover the residual designing errors before delivery to the customer. The failure data during process are taken down in order to estimate the software reliability

## Principles of Software Testing

* + - All tests should be traceable to end user requirements.
    - Tests should be planned long before testing begins
    - Testing should begin on a small scale and progress towards testing in large
    - To be most effective testing should be conducted by an independent third party

The primary objective for test case design is to derive a set of tests that has the highest livelihood for uncovering defects in software. To accomplish this objective two different categories of test case design techniques are used. They are

* + - White box testing.
    - Black box testing.

#### White-box testing:

White box testing focus on the program control 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 have been executed.

#### Block-box testing:

Black box testing is designed to validate functional requirements without regard to the internal workings of a program. Black box testing 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. Incorrect and missing functions, interface errors, errors in data structures, error in function all logic are the errors falling in this category.

23

**Chapter 5**

**SoftwareTesting**

## Testing fundamentals

Testing is a process of executing program with the intent of finding error. A good test case is one that has high probability of finding an undiscovered error. If testing is conducted successfully it uncovers the errors in the software. Testing cannot show the absence of defects, it can only show that software defects present.

## Testing Information flow:

Information flow for testing flows the pattern. Two class of input provided to test the process. The software configuration includes a software requirements specification, a design specification and source code.

Test configuration includes test plan and test cases and test tools. Tests are conducted and all the results are evaluated. That is test results are compared with expected results. When erroneous data are uncovered, an error is implied and debugging commences.

24

**CONCLUSION**

This was the first considerably large and important project undertaken by me during my BCA course. It was an experience that changed the way I perceived project development. The coding could not best arted before the whole system was completely finalized. Even then there were so many changes required and the coding needed to be changed. I attribute this to inadequate information gathering from the user. Though there were many meetings with the user and most of the requirements were gathered, a few misinterpretations of the requirements still crept in. It made me realize how important the systems analysis phase is. The project is a classic example, that learning of concepts needs to be supplemented with application of that knowledge.

On the whole it was a wonderful experience developing  **VRINDAVAN SEVA SADAN (THAKUR SHREE RADHA BIHARI CHARITABLE TRUST) RADHA MOHANVAN, CHHATTIKARA MARG, RAMAN RETI, SHRI DHAM VRINDAVAN(UP)** and I would have considered my education in complete with out undertaking such a project which allowed me to apply all that I have learnt and tried to develop a project that can be useful for people who are looking for accommodation very easily and efficiently. It is developed using PHP so that it can be accessed very easily and at any time. The system is developed with an aim of usability so that it is an easy to use system that requires the least amount of user input possible. For using this system general computer knowledge is enough. An easy well-structured module will show the correct path to reach the destination .Users will be authenticated to ensure that no unauthorized users gain access to private information.

25

**BIBLIOGRAPHY &REFERENCES**

To develop this website of Hotel VRINDAVAN SEVA SADAN (Thakur Shree Radha Bihari Charitable Trust) Radha Mohanvan, Chhattikara Marg, Raman Reti, Shri Dham Vrindavan(UP), we used PHP for Front End and SQL Server 2014 for Back End (Database). We take some knowledge towards automation system from some books that are given below:

## *The Joy of PHP Programming: A Beginner’s Guide – by Alan Forbes*

## *PHP & My SQL Novice to Ninja – by Kevin Yank*

### References:

### [1]. www.w3schools.com/php.net

### [2]. www.stackoverflow.com

### [3]. www.php.net-tutorial.com

### [4]. www.w3schools.com/css

26