

ONLINE AYURVEDIC AND WELLNESS RESORT

Project Report Submitted By

VAIDEHI M NAIR

Reg.No:AJC17MCA-I058

In Partial fulfillment for the Award of the Degree Of

**INTEGRATED MASTER OF COMPUTER APPLICATIONS
(INMCA)
APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY**



**AMAL JYOTHI COLLEGE OF ENGINEERING
KANJIRAPPALLY**

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE,
Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2017-2022

DEPARTMENT OF COMPUTER APPLICATIONS
AMAL JYOTHI COLLEGE OF ENGINEERING
KANJIRAPPALLY



CERTIFICATE

This is to certify that the Project report, “**ONLINE AYURVEDIC AND WELLNESS RESORT**” is the bonafide work of **VAIDEHI M NAIR (Reg.No:AJC17MCA-I058)** in partial fulfillment of the requirements for the award of the Degree of Integrated Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2017-22.

Ms. Merin Chacko
Internal Guide

Ms. Meera Rose Mathew
Coordinator

Rev. Fr. Dr. Rubin Thottupurathu Jose
Head of the Department

External Examiner

DECLARATION

I hereby declare that the project report “**ONLINE AYURVEDIC AND WELLNESS RESORT**” is a bonafided work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Degree of Integrated Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2017-2022.

Date:

KANJIRAPPALLY

VAIDEHI M NAIR

Reg. No: AJC17MCA-I058

ACKNOWLEDGEMENT

First and foremost, I thank God Almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev. Fr. Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Ms. Meera Rose Mathew** for her valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also like to express sincere gratitude to my guide **Ms. Merin Chacko** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

VAIDEHI M NAIR

ABSTRACT

Online Ayurvedic and Wellness Resort is an organized computerized system designed and programmed to deal with day to day operations and management of the ayurvedic resort. The program can look after check-ins, checkouts, records, database treatments, rooms, billings in the organisation. It also maintains resort information. This is an online ayurvedic and wellness resort that has various ayurvedic treatments and rooms available. This application will help people to choose their ayurvedic treatments and to choose their desired rooms to stay. People can choose their treatment packages after consulting the doctor. Users can purchase products from online store. Doctor can view assigned users and prescribe medicines. In current system it takes doctors appointment, room availability, online booking and online payment. The basic aim to provide best treatment at low cost. The major problem for the guests nowadays are to get report after treatment, many ayurvedic resorts has reports in their system but it's not available for the guest when they are outside. In this project we are going to provide the extra facility to store the report in the database and make available from anywhere in the world.

CONTENT

Sl. No	Topic	Page No
1	INTRODUCTION	2
1.1	PROJECT OVERVIEW	3
1.2	PROJECT SPECIFICATION	3
2	SYSTEM STUDY	5
2.1	INTRODUCTION	6
2.2	EXISTING SYSTEM	7
2.3	DRAWBACKS OF EXISTING SYSTEM	8
2.4	PROPOSED SYSTEM	8
2.5	ADVANTAGES OF PROPOSED SYSTEM	8
3	REQUIREMENT ANALYSIS	10
3.1	FEASIBILITY STUDY	11
3.1.1	ECONOMICAL FEASIBILITY	11
3.1.2	TECHNICAL FEASIBILITY	12
3.1.3	BEHAVIORAL FEASIBILITY	12
3.2	SYSTEM SPECIFICATION	13
3.2.1	HARDWARE SPECIFICATION	13
3.2.2	SOFTWARE SPECIFICATION	13
3.3	SOFTWARE DESCRIPTION	13
3.3.1	PHP	13
3.3.2	MYSQL	13
4	SYSTEM DESIGN	16
4.1	INTRODUCTION	17
4.2	UML DIAGRAM	17
4.2.1	USE CASE DIAGRAM	18
4.2.2	SEQUENCE DIAGRAM	19
4.2.3	CLASS DIAGRAM	21
4.2.4	ACTIVITY DIAGRAM	22
4.2.5	COMPONENT DIAGRAM	24

4.2.6	DEPLOYMENT DIAGRAM	25
4.2.7	STATE DIAGRAM	26
4.3	USER INTERFACE DESIGN	27
4.4	DATA BASE DESIGN	30
5	SYSTEM TESTING	42
5.1	INTRODUCTION	43
5.2	LEVELS OF TEST	44
5.2.1	UNIT TESTING	44
5.2.2	INTEGRATION TESTING	45
5.2.3	VALIDATION TESTING	45
5.2.4	USER ACCEPTANCE TESTING	45
5.2.5	AUTOMATION TESTING	46
5.2.6	SELENIUM TESTING	46
5.2.1.1	TEST CASE FOR LOGIN PAGE	47
5.2.1.2	TEST CASE FOR UPDATE ROOM	50
6	IMPLEMENTATION	53
6.1	INTRODUCTION	54
6.2	IMPLEMENTATION PROCEDURE	54
6.2.1	USER TRAINING	55
6.2.2	TRAINING APPLICATION SOFTWARE	55
6.2.3	SYSTEM MAINTENANCE	55
7	CONCLUSION & FUTURE SCOPE	57
7.1	CONCLUSION	58
7.2	FUTURE SCOPE	58
8	BIBLIOGRAPHY	59
9	APPENDIX	61
9.1	SAMPLE CODE	62
9.2	SCREEN SHOTS	68

List of Abbreviation

IDE	-	Integrated Development Environment
HTML	-	Hyper Text Markup Language.
CSS	-	Cascading Style Sheet
SQL	-	Structured Query Language
UML	-	Unified Modeling Language

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

“Online Ayurvedic Wellness Resort” is a web based project. The program can look after checkins, checkouts, records, database treatments, rooms, billings in the organisation. It also maintains resort information. This is an online ayurvedic and wellness resort that has various ayurvedic treatments and rooms available. This application will help people to choose their ayurvedic treatments and to choose their desired rooms to stay. People can choose their treatment packages after consulting the doctor, In current system it takes doctors appointment, room availability, online booking and online payment. The basic aim to provide best treatment at low cost. The major problem for the guests nowadays are to get report after treatment, many ayurvedic resorts has reports in their system but it's not available for the guest when they are outside. In this project we are going to provide the extra facility to store the report in the database and make available from anywhere in the world.

Hospital management encompasses all facets of a hospital, as well as the coordination of all of its components. This could include everything from patient care to record keeping to drug inventory and cleanliness.

To ensure that their company runs successfully, all hospital healthcare managers collaborate with clinicians, make policy choices, oversee patient care, budgeting and accounting, and lead marketing activities.

1.2 PROJECT SPECIFICATION

The proposed system is a website in which user can choose ayurvedic treatments and rooms. The system includes 4 Actors. They are:

1. Admin Module

Admin must have a login into this system. He has the overall control of the system. Admin can add or update ayurvedic packages and rooms and products for the online store. Admin can manage doctor and staff. Admin can approve and reject the leave requests from staff and doctor. Admin can View all the registered users and also manage all his data. Admin can view feedback and maintenance details. Admin can add doctor and staff by giving them a username and password. In this project admin assigns each registered user to a particular doctor.

2. User Module

User can register and login to the system. They can book treatments and rooms and manage their data. Users can purchase products from online store. Users can view their assigned doctors for treatment. The project provides a secure online payment for the users ensuring high security. They can view their prescriptions that is given by the doctor. Razorpay is the payment gateway used in the system. The razorpay checkout offers multiple payment methods, allowing your customer the flexibility to complete the payment using the payment method of their choice.

3. Doctor Module

Doctor is the next main module of the system. Doctor can manage treatments. Doctor can view assigned users and their details. In this project doctor can prescribe medicines to the patients and generate a medical report. Doctor can view the previous prescriptions. Doctor can apply leave and is able to view the status that is accepted, rejected or pending. The leave request is approved or rejected by the admin. Doctor can delete the leave request at any time. Doctor can update their profile and password.

4. Staff Module

Staff can login to the system. They can manage their details. Staff can apply leave and is able to view the status that is accepted, rejected or pending. The leave request is approved or rejected by the admin. Staff can delete the leave request at any time. Staff can enter maintenance details. Also staff can enter expenses of the organization.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minute's detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies, a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

2.2 EXISTING SYSTEM

Hospitals currently use a manual system for the management and maintenance of critical information. The current system requires numerous paper forms, with data stores spread through out the hospital management infrastructure. Often information is incomplete or does not follow management standards. Forms are often lost in transit between departments requiring a comprehensive auditing process to ensure that no vital information is lost. Multiple copies of the same information exist in the hospital and may lead to inconsistencies in data in various data stores.

In the existing system, for a patient to contact a doctor and take an appointment is possible only if patient goes to that particular hospital. Although the people can't get the correct information about doctors, their details and different services provided by the hospital. The only way is to contact directly. Those who want to have some information in the medical reports or files or want to get appointment to particular doctor from his own place are not possible. In cases, a person suffering from some diseases may not have the medical document with them. This may result in severe problems. Nowadays in order to get correct information and right treatment for a patient has to go by him wherever needed. This is a lengthy process, which takes a lot of time, and also costs are high. It's not possible to get all the information or details as well as we can't satisfy the user through this process. The existing system has drawbacks in the following areas:

- Time consuming
- Reminds the user about prescribed medicines
- Doctor patient interaction
- To maintain the medical documents
- Getting hospital details and services provided by the hospital
- Booking appointments
- Patient's registration

It is necessary to modify the existing system in order to include additional information and make the system efficient, flexible and secure. The existing system of ayurvedic wellness resort was manual. All the daily routines is carried out manually and the records are maintained in the record books or the registers.

2.3 DRAWBACKS OF EXISTING SYSTEM

- Less convenient
- Human effort is needed

2.4 PROPOSED SYSTEM

The user can view the availability of doctors and can view various medical files and data such as DICOM files, test reports, medicine prescription, bills and visiting history etc. In this application doctors have access to information related to their patients and can view their reports. It also provides the live interaction between the doctors and patients. The application also reminds the user about prescribed medicines by keeping in synchronize with his prescription updated in the database and providing notification or alerts. The server side of the system has added the doctors' details such as availability of each doctor, consultation hours and specialization etc. The admin also add the patients' details such as medical information files, test reports, medicine prescription, bills, visiting history etc. The application includes case studies of patients in DICOM format. Digital Imaging and Communications in Medicine (DICOM) is a standard for handling, storing, printing, and transmitting information in medical imaging. It includes a file format definition and a network. DICOM files can be exchanged between two entities that are capable of receiving image and patient data in DICOM format. All DICOM images needed to be securely exchanged over network.

2.5 ADVANTAGES OF PROPOSED SYSTEM

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has got following features:

➤ **One for all:-**

Everything comes under single roof creates less vagueness about all and need not to have separate application for each services.

➤ **Better security:-**

For data to remain secure measures must be taken to prevent unauthorized access. Security means that data are protected from various forms of destruction. The system security problem can be divided into four related issues: security, integrity, privacy and confidentiality.

Username and password requirement to sign in ensures security. It will also provide data security as we are using the secured databases for maintaining the documents.

➤ **Ensure data accuracy:-**

The proposed system eliminates the manual errors while entering the details of the users during the registration.

➤ **Better service: -**

The product will avoid the burden of hard copy storage. We can also conserve the time and human resources for doing the same task. The data can be maintained for longer period with no loss of data.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features: -

3.1.1 Economical Feasibility

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require. This term means the assessment and analysis of a project's potential to support the decision-making process by objectively and rationally identifying its strengths, weaknesses, opportunities and risks associated with it, the resources that will be needed to implement the project, and an assessment of its chances of success.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

The proposed system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development.

The cost of project, **Online Ayurvedic Wellness Resort** was divided according to the system used, its development cost and cost for hosting the project. According to all the calculations the project was developed in a low cost. As it is completely developed using open source software.

3.1.2 Technical Feasibility

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

- Does the existing technology sufficient for the suggest done?
- Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project requires High Resolution Scanning device and utilizes Cryptographic techniques. Through the technology may become obsolete after some period of time, due to the fact that newer version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using PHP in front end and MySQL in server in back end, the project is technically feasible for development. The system has been developed using PHP in front end and MySQL in server in back end, the project is technically feasible for development. The System used was also of good performance of Processor Intel i3 core; RAM 4GB and, Hard disk 1TB.

3.1.3 Behavioral Feasibility

The proposed system includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible. It helps in determining if the system requires special effort to educate, retrain, transfer, and changes in employee's job status on new ways of conducting business.

Online Ayurvedic Wellness Resort , GUI is simple so that users can easily use it. **Online Ayurvedic Wellness Resort** is simple enough so that no training is needed.

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor : Intel core i4 and above

RAM : 512 MB and above

HDD : 40 GB and above

3.2.2 Software Specification

Client on PC : Windows, Linux

Browser : Google Chrome, Microsoft Edge

Frontend : HTML, CSS

Backend : MySQL

Technologies used : JS, HTML5, AJAX, J Query, PHP, CSS

3.3 SOFTWARE DESCRIPTION

3.3.1 PHP

PHP is a server side scripting language designed for web development but also used as a general purpose programming language. PHP is now installed on more than 244 million websites and 2.1 million web servers. Originally created by Rasmus Lerdorf in 1995, the reference implementation of PHP is now produced by the PHP group. While PHP originally stood for personal Home page, it now stands for PHP:Hypertext Preprocessor, a recursive acronym. PHP code is interpreted by a web server with a PHP processor module which generates the resulting web page. PHP commands can be embedded directly into a HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface capability and can be used in standalone incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term PHP. PHP can be deployed on most web servers and also as a standalone shell on almost every operating system and platform, free of charge.

3.3.2 MySQL

MySQL, the most popular Open Source SQL database management system, is developed,

distributed, and supported by Oracle Corporation. The MySQL Web site provides the latest information about MySQL software.

- **MySQL is a database management system.**

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

- **MySQL databases are relational.**

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and “pointers” between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data. The SQL part of “MySQL” stands for “Structured Query Language”. SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax. SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986.

- **MySQL software is OpenSource**

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information

- **MySQL software is OpenSource**

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information.

- **The MySQL Database Server is very fast, reliable, scalable, and easy to use.**

If that is what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available.

- **MySQL Server works in client/server or embedded systems.**

The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs). We also provide MySQL Server as an embedded multi-threaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

4.2 UML DIAGRAM

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.

UML stands for **Unified Modeling Language**. UML is different from the other common programming languages such as C++, Java, COBOL, etc. UML is a pictorial language used to make software blueprints. UML can be described as a general purpose visual modeling language to visualize, specify, construct, and document software system. Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc. UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object oriented analysis and design. After some standardization, UML has become an OMG standard. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete. UML includes the following nine diagrams.

- Class diagram
- Use case diagram
- Sequence diagram
- Activity diagram
- State chart diagram
- Deployment diagram
- Component diagram
- Object diagram
- Collaboration diagram

4.2.1 USE CASE DIAGRAM

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems.

System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service-oriented task. For example, use cases in a product sales environment would include item ordering, catalog updating, payment processing, and customer relations. A use case diagram contains four components.

- The boundary, which defines the system of interest in relation to the world around it.
- The actors, usually individuals involved with the system defined according to their roles.
- The use cases, which are the specific roles are played by the actors within and around the system.
- The relationships between and among the actors and the usecases.

Use case diagrams are drawn to capture the functional requirements of a system. After identifying the above items, we have to use the following guidelines to draw an efficient use case diagram

- The name of a use case is very important. The name should be chosen in such a way so that it can identify the functionalities performed.

- Give a suitable name for actors.
- Show relationships and dependencies clearly in the diagram.
- Do not try to include all types of relationships, as the main purpose of the diagram is to identify the requirements.
- Use notes whenever required to clarify some important points.

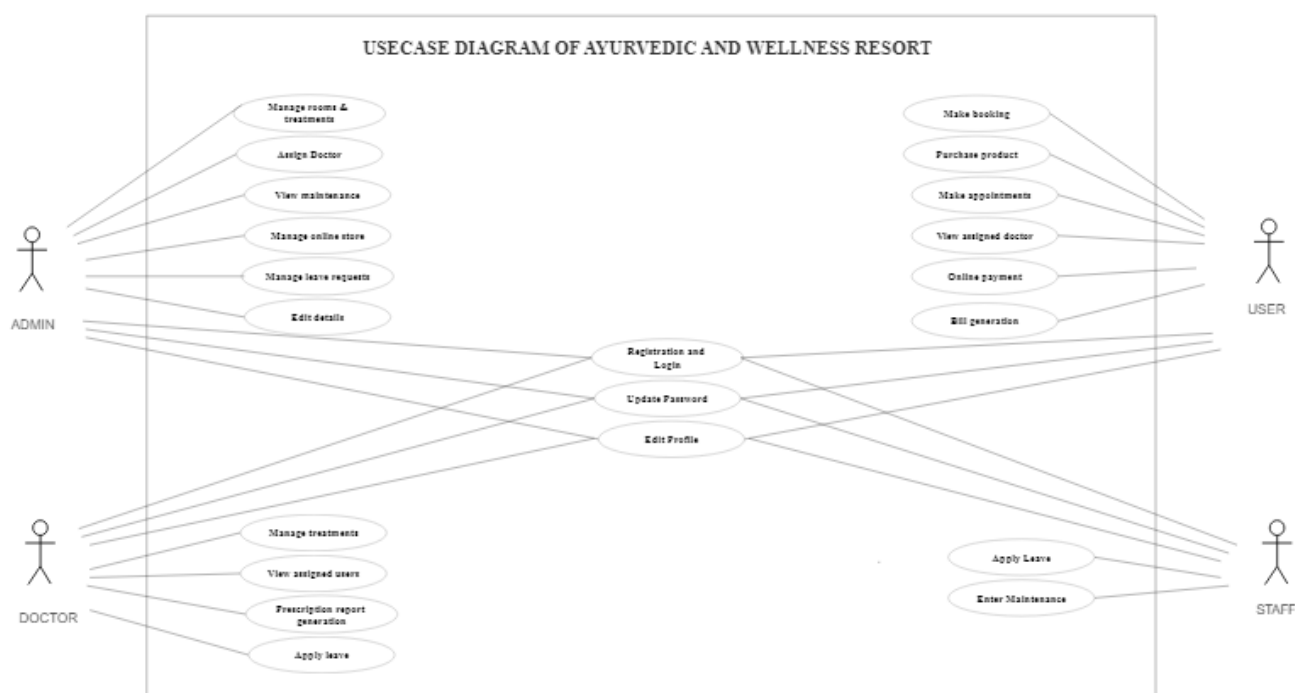


Fig 1 : Use case diagram for online ayurvedic and wellness resort

4.2.2 SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

Sequence Diagram Notations –

- i **Actors** – An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram. We use actors to depict various roles including human users and other external subjects. We represent an actor in a UML diagram using a stick person notation. We can have multiple actors in a sequence diagram.
- ii **Lifelines** – A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram.
- iii **Messages** – Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram.

Messages can be broadly classified into the following categories:

- Synchronous messages
 - Asynchronous Messages
 - Create message
 - Delete Message
 - Self-Message
 - Reply Message
 - Found Message
 - Lost Message
- iv. **Guards** – To model conditions we use guards in UML. They are used when we need to restrict the flow of messages on the pretext of a condition being met. Guards play an important role in letting software developers know the constraints attached to a system or a particular process.

Uses of sequence diagrams –

- Used to model and visualize the logic behind a sophisticated function, operation.
- They are also used to show details of UML use case diagrams.

- Used to understand the detailed functionality of current or future systems.
- Visualize how messages and tasks move between objects or components in a system.

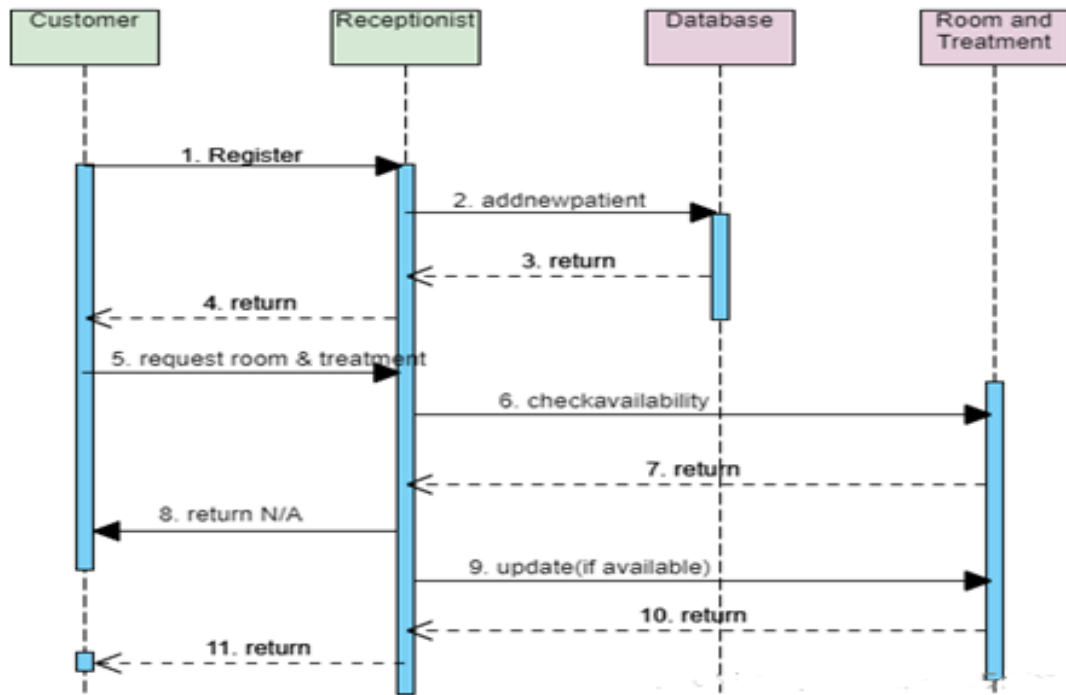


Fig 2 : Sequence diagram for online ayurvedic and wellness resort

4.2.3 CLASS DIAGRAM

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages.

UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application, however class diagram is a bit different. It is the most popular UML diagram in the coder community.

The purpose of the class diagram can be summarized as –

- Analysis and
- Design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.

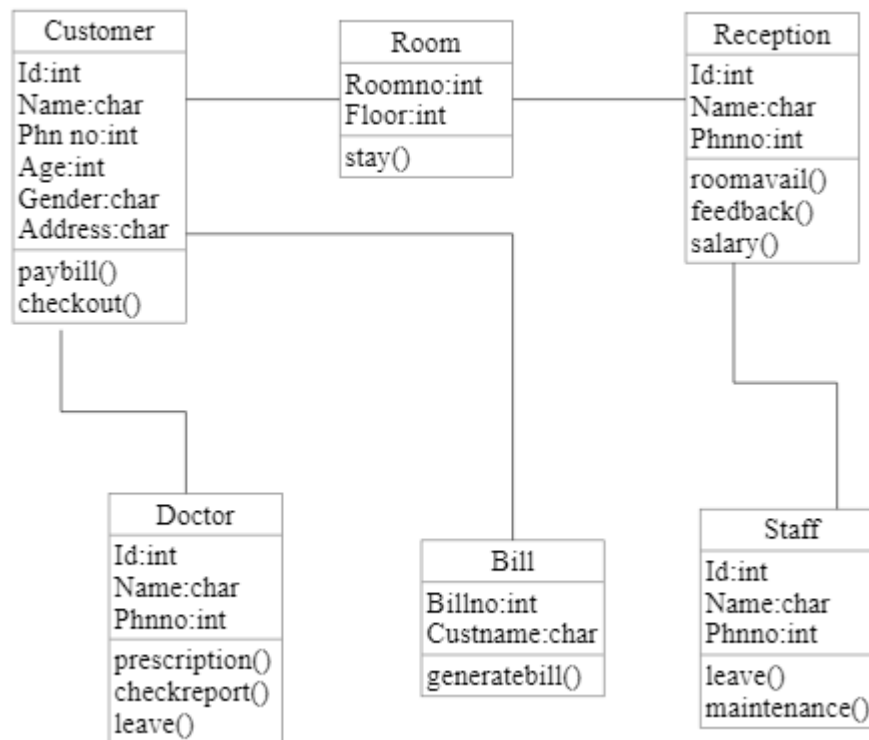


Fig3 : Class diagram for online ayurvedic and wellness resort

4.2.4 ACTIVITY DIAGRAM

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc

The basic purposes of activity diagrams is similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

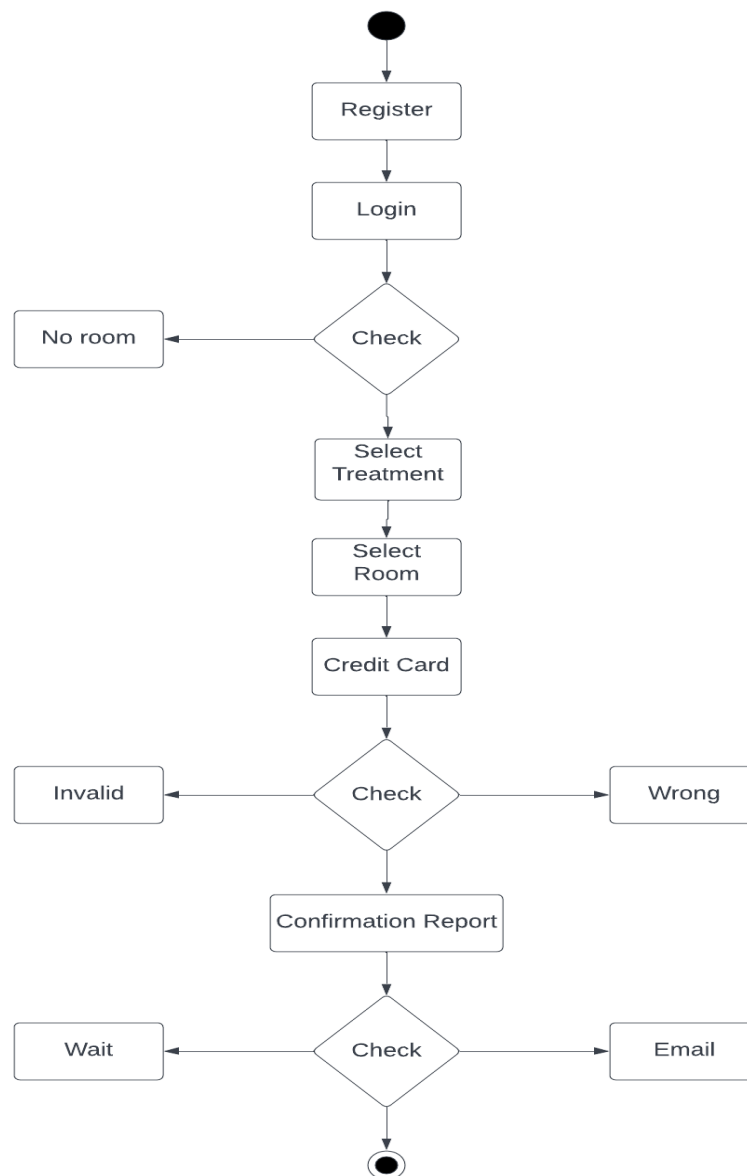


Fig 4 : Activity diagram for online ayurvedic and wellness resort

4.2.5 COMPONENT DIAGRAM

Component diagrams are different in terms of nature and behavior. Component diagrams are used to model the physical aspects of a system. Now the question is, what are these physical aspects? Physical aspects are the elements such as executables, libraries, files, documents, etc. which reside in a node. Component diagrams are used to visualize the organization and relationships among components in a system. These diagrams are also used to make executable systems.

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities. Thus from that point of view, component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files, etc. Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment.

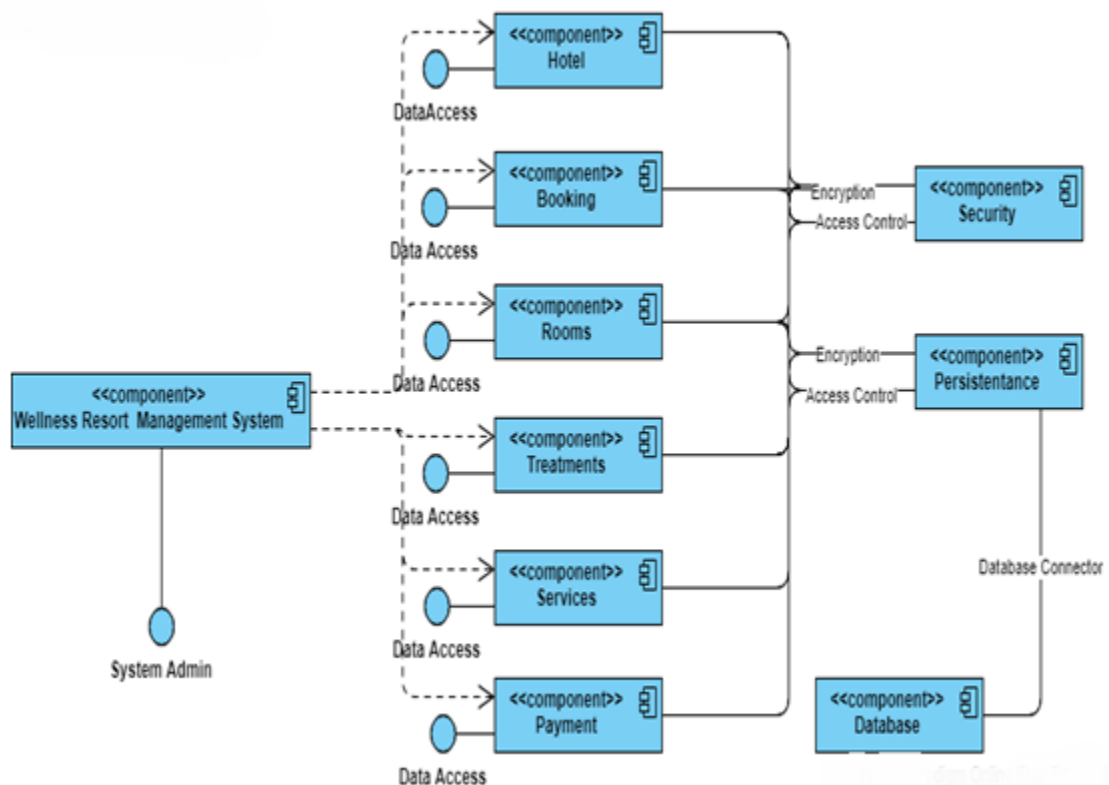


Fig 5 :Component diagram for online ayurvedic and wellness resort

4.2.6 DEPLOYMENT DIAGRAM

Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed. Deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships. The term Deployment itself describes the purpose of the diagram. Deployment diagrams are used for describing the hardware components, where software components are deployed. Component diagrams and deployment diagrams are closely related. Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware. UML is mainly designed to focus on the software artifacts of a system. However, these two diagrams are special diagrams used to focus on software and hardware components. It maps the software architecture created in design to the physical system architecture, where the software will be executed as a node.

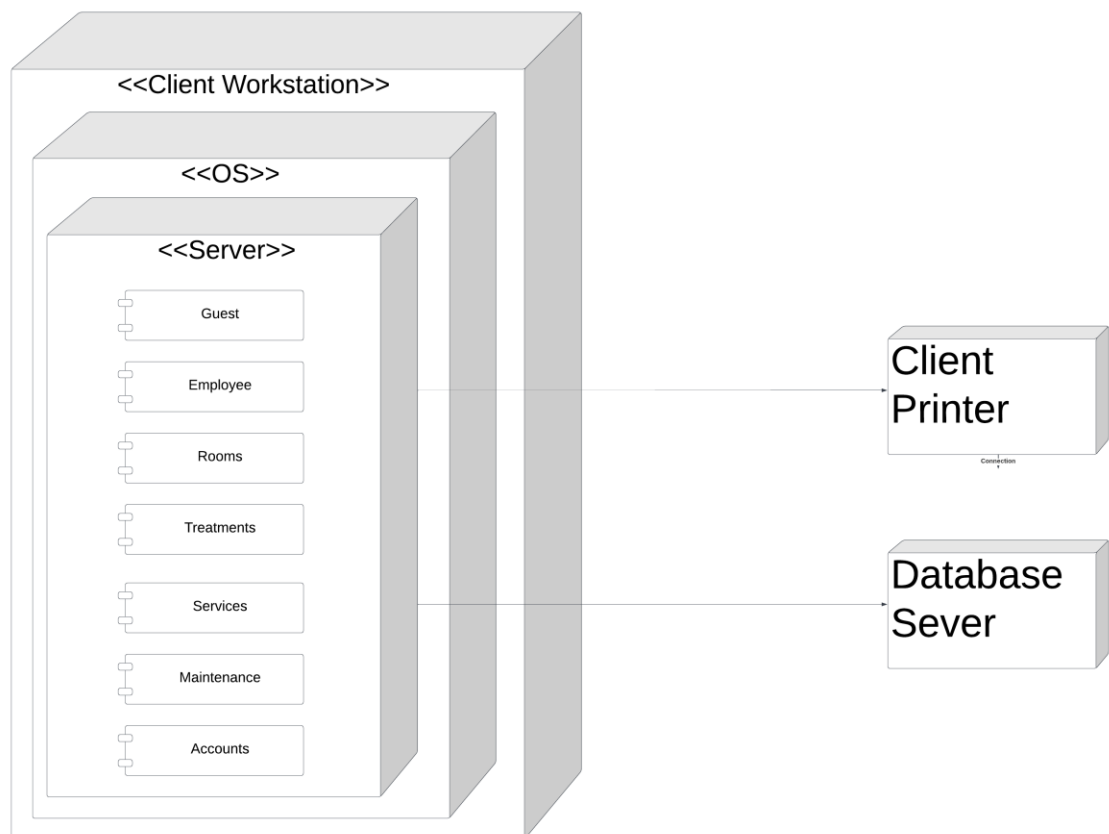


Fig 6 :Deployment diagram for online ayurvedic and wellness resort

4.2.7 STATE DIAGRAM

The name of the diagram itself clarifies the purpose of the diagram and other details. It describes different states of a component in a system. The states are specific to a component/object of a system.

A State chart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events.

Activity diagram explained in the next chapter, is a special kind of a State chart diagram. As State chart diagram defines the states, it is used to model the lifetime of an object.

State chart diagram is one of the five UML diagrams used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by events. State chart diagrams are useful to model the reactive systems. Reactive systems can be defined as a system that responds to external or internal events.

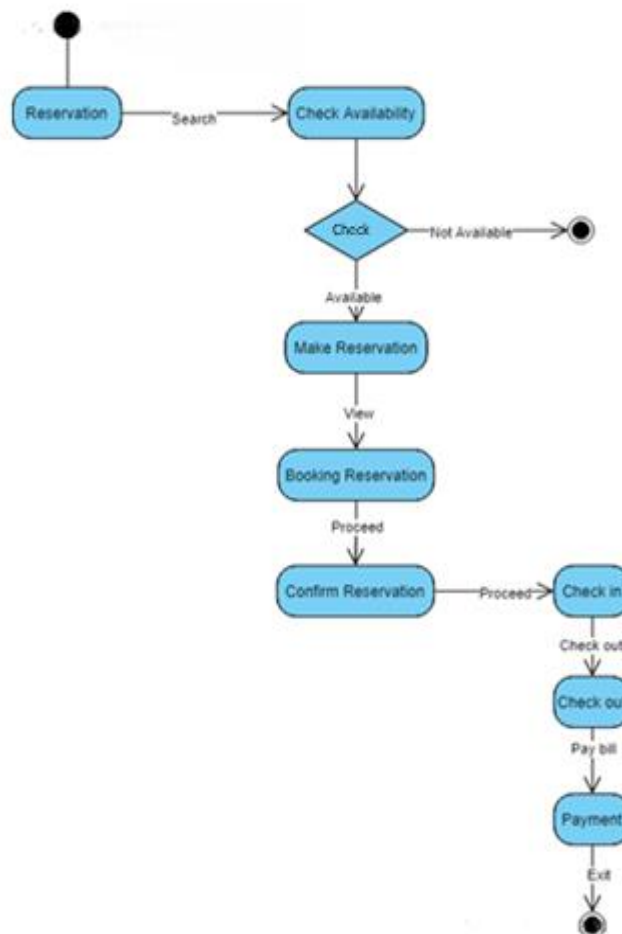


Fig 7 :State diagram for online ayurvedic and wellness resort

4.3 USER INTERFACE DESIGN

4.3.1-INPUT DESIGN

Form Name : User Login

LOGIN HERE

Enter your email id

Enter Password

Login

Don't have an account ?[Signup](#)

Form Name : User Registration

REGISTER HERE

First Name

Email

Mobile

Enter Password

Confirm Password

Register

Already have an account ?[Signin](#)

Form Name : Add products to online store

ADD PRODUCT

Product Name:

Image:

No file chosen

Amount:

4.3.1-OUTPUT DESIGN

User Login

[Home](#) [Ayurvedic Packages](#) [Accomodations](#) [Gallery](#) [Login](#)

LOGIN HERE

Enter your email id

Enter Password

Login

Don't have an account ?[Signup](#)

User Register

[Home](#) [Ayurvedic Packages](#) [Accomodations](#) [Gallery](#) [Login](#)

REGISTER HERE

First Name

Email


Mobile

Enter Password

Confirm Password

Already have an account ? [Signin](#)

Add Ayurvedic Packages



Tree of Life

- Dashboard
- Ayurvedic Packages
- Rooms
- Products
- Assign
- Leave Doctor
- Leave Staff
- Feedback
- Maintenance

ADD PRODUCT

Product Name:

Image:

No file chosen

Amount:

4.4 DATABASE DESIGN

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is a two level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

- Data Integrity
- Data independence

Relational Database Management System(RDBMS)

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables, each of which is assigned a unique name. A row in a table represents a set of related values.

Relations, Domains & Attributes

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain D is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values.

Every value in a relation is atomic, that is not decomposable.

Relationships

- Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.
- Entity Integrity enforces that no Primary Key can have null values.
- Referential Integrity enforces that no Primary Key can have null values.
- Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key are Super Key and Candidate Keys.

Normalization

Data are grouped together in the simplest way so that later changes can be made with minimum impact on data structures. Normalization is formal process of data structures in manners that eliminates redundancy and promotes integrity. Normalization is a technique of separating redundant fields and breaking up a large table into a smaller one. It is also used to avoid insertion, deletion, and updating anomalies. Normal form in data modelling use two concepts, keys and relationships. A key uniquely identifies a row in a table. There are two types of keys, primary key and foreign key. A primary key is an element or a combination of elements in a table whose purpose is to identify records from the same table. A foreign key is a column in a table that uniquely identifies record from a different table. All the tables have been normalized up to the third normal form.

As the name implies, it denotes putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These include:

- ✓ Normalize the data.
- ✓ Choose proper names for the tables and columns.
- ✓ Choose the proper name for the data.

First Normal Form

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words 1NF disallows “relations within relations” or “relations as attribute values within tuples”. The only attribute values permitted by 1NF are single atomic or indivisible values. The first step is to put the data into First Normal Form. This can be done by moving data into separate tables where the data is of similar type in each table. Each table is given a Primary Key or Foreign

Key as per requirement of the project. In this we form new relations for each non-atomic attribute or nested relation. This eliminated repeating groups of data. A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only.

Second Normal Form

According to Second Normal Form, for relations where primary key contains multiple attributes, no non-key attribute should be functionally dependent on a part of the primary key. In this we decompose and setup a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it. This step helps in taking out data that is only dependent on a part of the key. A relation is said to be in second normal form if and only if it satisfies all the first normal form conditions for the primary key and every non-primary key attributes of the relation is fully dependent on its primary key alone.

Third Normal Form

According to Third Normal Form, Relation should not have a non-key attribute functionally determined by another non-key attribute or by a set of non-key attributes. That is, there should be no transitive dependency on the primary key. In this we decompose and set up relation that includes the non-key attributes that functionally determines other non-key attributes. This step is taken to get rid of anything that does not depend entirely on the Primary Key. A relation is said to be in third normal form if only if it is in second normal form and more over the non key attributes of the relation should not be depend on other non-key attribute.

TABLE DESIGN**Table No 01: tbl_login**

Primary Key: login_id

Foreign Key: reg_id references tbl_register

type_id references tbl_usertype

Field Name	Type	Size	Description
login_id	int	50	Primary key
reg_id	int	50	Foreign Key
type_id	int	50	Foreign key

Table No 02: tbl_register

Primary Key: reg_id

Field Name	Type	Size	Description
reg_id	int	11	Primary key
reg_name	varchar	15	To store customer name
reg_email	varchar	50	To store customer email
reg_phone	varchar	12	To store customer mobile
password	varchar	100	To store customer password
status	int	10	To check availability

Table No 03: tbl_booking

Primary Key: book_id

Field Name	Type	Size	Description
book_id	int	10	Primary key
book_address	varchar	100	To store user address
book_gender	varchar	20	To store gender
book_dob	varchar	10	To store date of birth
book_city	varchar	50	To store customer city
book_pincode	varchar	10	To store pincode
book_treatment	varchar	100	To store package
book_room	varchar	50	To store room
book_ade	varchar	40	To store arrival date

Table No 04: tbl_doctdet

Primary Key: doctor_id

Foreign Key: reg_id references tbl_register

Field Name	Type	Size	Description
doctor_id	int	10	Primary key
reg_id	varchar	10	Foreign key
doctor_address	varchar	50	To store doctor address
doctor_gender	varchar	30	To store gender
doctor_dob	varchar	30	To store date of birth
doctor_spec	varchar	50	To store specialization
doctor_quali	varchar	255	To store qualification
doctor_idp	varchar	255	To store id proof
doctor_photo	varchar	255	To store photo

Table No 05 : tbl_room

Primary Key : room_id

Field Name	Type	Size	Description
room_id	int	20	Primary key
room_name	varchar	30	To store room name
room_photo	varchar	255	To store room photo
room_amt	varchar	30	To store room amount
room_status	int	10	To store room status

Table No 06: tbl_treatment

Primary Key: treatment_id

Field Name	Type	Size	Description
treatment_id	Int	11	Primary key
treatment_name	varchar	30	To store treatment name
treatment_photo	varchar	255	To store treatment photo
treatment_days	varchar	30	To store days
treatment_cost	varchar	30	To store cost
treatment_status	int	20	To store status

Table No 07 : tbl_usertype

Primary Key : type_id

Field Name	Type	Size	Description
type_id	int	20	To store user type id
type_name	varchar	50	To store usertype names

Table No 08: payment

Primary Key: id

Field Name	Type	Size	Description
id	int	11	Primary key
name	varchar	50	To store name
amount	int	11	To store amount
payment_status	varchar	50	To store payment status
payment_id	varchar	50	To store payment id
added_on	datetime		To store date and time

Table No 09: tbl_assign

Primary Key: assign_id

Foreign Key: doct_id references tbl_register

cust_id references tbl_usertype

Field Name	Type	Size	Description
login_id	int	20	Primary key
doct_id	int	20	Foreign Key
cust_id	int	20	Foreign key

Table No 10: tbl_cart

Primary Key: cart_id

Foreign Key: product_id references tbl_product

reg_id references tbl_register

Field Name	Type	Size	Description
assign_id	int	20	Primary key
product_id	int	20	Foreign Key
reg_id	int	20	Foreign key

Table No 11: tbl_feedback

Primary Key: feedback_id

Foreign Key: feedback_regid references tbl_register

Field Name	Type	Size	Description
feedback_id	int	20	Primary key
feedback_regid	int	20	Foreign Key
feedback	varchar	200	To store feedback

Table No 12: tbl_leavedoc

Primary Key: ld_id

Foreign Key: reg_id references tbl_register

Field Name	Type	Size	Description
ld_id	int	20	Primary key
reg_id	int	20	Foreign key
ld_date	varchar	50	To store date
ld_reason	varchar	50	To store reason
ld_status	int	20	To store status

Table No 13: tbl_leavestaff

Primary Key: ls_id

Foreign Key: reg_id references tbl_register

Field Name	Type	Size	Description
ls_id	int	20	Primary key of login table
reg_id	int	20	Foreign key of tbl_register
ls_date	varchar	50	To store date
ls_reason	varchar	50	To store reason
ls_status	int	20	To store status

Table No 14: tbl_maintenance

Primary Key: m_id

Field Name	Type	Size	Description
m_id	int	20	Primary key
m_room	varchar	100	To store room details
m_problem	varchar	200	To store problem

Table No 15: tbl_prescription

Primary Key: p_id

Field Name	Type	Size	Description
p_id	int	10	Primary key
p_name	varchar	150	To store name
p_age	int	20	To store age
p_prob	varchar	150	To store problem
p_pres	varchar	150	To store prescription

Table No 16: tbl_product

Primary Key: product_id

Field Name	Type	Size	Description
product_id	int	10	Primary key
product_name	varchar	50	To store name
product_image	varchar	255	To store image
product_amount	varchar	10	To store amout
product_status	int	10	To store status

TableNo 17: tbl_staffdet

PrimaryKey: staff_id

ForeignKey: reg_id references tbl_register

Field Name	Type	Size	Description
staff_id	int	10	Primary key of tbl_staffdet
reg_id	varchar	10	Foreign key of tbl_register
staff_address	varchar	50	To store address
staff_dob	varchar	30	To store date of birth
staff_spec	varchar	50	To store specialization
staff_quali	varchar	255	To store qualification
staff_idp	varchar	255	To store id proof

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

Software Testing is the process of executing software in a controlled manner, in order to answer the question- Does the software behave as specified? Software testing is often used in association with the terms verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis looks at the behavior of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is complete without testing, as it vital success of the system testing objectives, there are several rules that can serve as testing objectives. They are:

Testing is a process of executing a program with the intent of finding an error.

- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

If a testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrate that the software function appear to be working according to the specification, that performance requirement appear to have been met.

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Test for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

5.2 LEVELS OF TEST

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan.

The levels of testing include:

- ❖ Unit testing
- ❖ Integration Testing
- ❖ Data validation Testing
- ❖ Output Testing
- ❖ Automation Testing
- ❖ Selenium Testing

5.2.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design—the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is white-box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be Anticipated and error handling paths setup to reroute or cleanly terminate processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries. Unit

testing was done in Sell-Soft System by treating each module as separate entity and testing each one of them with a wide spectrum of test inputs. Some flaws in the internal logic of the modules were found and were rectified. After coding each module is tested and run individually. All unnecessary code where removed and ensured that all modules are working, and gives the expected result.

5.2.2 Integration Testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop. After performing unit testing in the System all the modules were integrated to test for any inconsistencies in the interfaces. Moreover differences in program structures were removed and a unique program structure was evolved.

5.2.3 Validation Testing or System Testing

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as Black Box testing or System tests.

Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program.

Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

5.2.4 Output Testing or User Acceptance Testing

The system considered is tested for user acceptance; here it should satisfy the firm's need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points:

- Input Screen Designs
- Output Screen Designs

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

5.2.5 Automation Testing

Automation testing is the process of testing software and other tech products to ensure it meets strict requirements. Essentially, it's a test to double-check that the equipment or software does exactly what it was designed to do. It tests for bugs, defects, and any other issues that can arise with product development. Automation testing can be run at any time of the day. It uses scripted sequences to examine the software. It then reports on what's been found, and this information can be compared with earlier test runs

5.2.6 Selenium Testing

Selenium is an open-source tool that automates web browsers. It provides a single interface that lets you write test scripts in programming languages like Ruby, Java, NodeJS, PHP, Perl, Python, and C#, among others. A browser-driver then executes these scripts on a browser-instance on your device. Selenium IDE is a Chrome and Firefox plugin that can log 'natural' interactions in the browser and generate its code in programming languages like C#, Java, Python, and Ruby, as well as Selenese. Testers can enable 'recording' within the IDE and 'play out' the test scenario on the browser.

Part of the reason why Selenium is so popular is its flexibility. Anyone who codes for the web can use Selenium to test their code/app—from individual freelance developers running a quick series of tests for debugging to UI engineers doing visual regression tests after a new integration.

In an enterprise environment, testing with Selenium falls under the purview of QA engineers. They are tasked with writing focused, non-flaky (i.e., deterministic) scripts to maximize test coverage and accuracy, refactoring old test suites for newer versions of the project, and maintaining test infrastructure (from the Hub to the test-case library).

5.2.1.1 Test Case for Login Page

Test Case 1					
Project Name : Online Ayurvedic and Wellness Resort					
Login Test Case					
Test Case ID: login			Test Designed By: Vaidehi M Nair		
Test Priority(Low/Medium/High):High			Test Designed Date: 23-05-2022		
Module Name: Login Screen					
Test Title : Verify login with valid username and password			Test Execution Date: 23-05-2022		
Description: Test the Login Page					
Pre-Condition :User has valid user name and password					
Test Case No	Test Step	Test Data	Expected Result	Actual Result	Status(Pass/Fail)
1	Navigation to Login Page		Login Page should be	Login page displayed	Pass
2	Provide Valid Username	User Name: jyothi@gmail.com	User should be able to Login	User Logged in and navigated to Admin Dashboard with records	Pass
	Provide Valid Password	Password: 123			
	Click on Login button				
3	Provide Invalid UserName or password	User Name: jyothi@gmail.com Password: 1234	User should not be able to Login	Message for enter valid user or password displayed	Pass
	Provide Null UserName or Password	UserName:null Password: null			
	Click on Login button				
Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database					

Code

```
package testcase;
import org.openqa.selenium.By;
import
org.openqa.selenium.WebDriver;
import
browserimplement.DriverSetup;
public class Firsts {
public static WebDriver driver;
public static void main(String[] args) throws InterruptedException {
// TODO Auto-generated method stub
driver = DriverSetup.getWebDriver("http://localhost/to/login.php");
//login-Invalid case
driver.findElement(By.name("email")).sendKeys("jyothi@gmail.com");
driver.findElement(By.name("psw")).sendKeys("123");
driver.findElement(By.name("submit")).click();
Thread.sleep(8000);
String actualUrl="http://localhost/to/user.php";

String expectedUrl= driver.getCurrentUrl();
if(actualUrl.equalsIgnoreCase(expectedUrl)) {
System.out.println("Test passed"); }
else {
System.out.println("Test failed"); }
driver.quit();
}
}
```

Output

```
eclipse-workspace - MainProject/src/testcases/Test1.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

# module-info.java module-info.java *Test1.java DriverSetup.java

1 package testcases;
2
3 import org.openqa.selenium.By;
4
5
6
7
8
9
10 public class Test1 {
11     public static WebDriver driver;
12     public static void main(String[] args) throws InterruptedException {
13         // TODO Auto-generated method stub
14         driver = DriverSetup.getWebDriver("http://localhost/tol/login.php");
15         //login-Invalid case
16         driver.findElement(By.name("email")).sendKeys("jyothi@gmail.com");
17         driver.findElement(By.name("password")).sendKeys("123");
18         driver.findElement(By.name("submit")).click();
19         Thread.sleep(8000);
20         String actualUrl="http://localhost/tol/user.php";
21         String expectedUrl= driver.getCurrentUrl();
22         if(actualUrl.equalsIgnoreCase(expectedUrl)) {
23             System.out.println("Test passed"); } else { System.out.println("Test failed"); }
24         driver.quit();
25     }
26 }
```

```
File Edit Source Refactor Navigate Search Project Run Window Help

# Declaration Console
terminated> Firsts [Java Application] C:\Users\User\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_16.0.1.v20210528-1205\jre\bin\javaw.exe (May 18, 2022, 12:40:12 PM - 12:40:29 PM)

SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Starting ChromeDriver 101.0.4951.41 (93c720db8323b3ec10d056025ab95c23a31997c9-refs/branch-heads/4951@{#904}) on port 60761
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
May 18, 2022 12:40:16 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
May 18, 2022 12:40:16 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found exact CDP implementation for version 101
Test passed
```


5.2.1.2 Test Case for Update Room

Test Case 2					
Project Name : Online Ayurvedic and Wellness Resort					
Login Test Case					
Test Case ID: updateroom			Test Designed By: Vaidehi M Nair		
Test Priority(Low/Medium/High):High			Test Designed Date: 23-05-2022		
Module Name: Login Screen					
Test Title : Update rooms details			Test Execution Date: 23-05-2022		
Description: Login to system and update room information, if some error occurs, test will fail					
Pre-Condition :User has valid user name and password					
Test Case No	Test Step	Test Data	Expected Result	Actual Result	Status(Pass/Fail)
1	Navigation to Login Page		Login Page should be	Login page displayed	Pass
2	Provide Valid User name	User Name: admin@gmail.com	User should be able to Login	User Logged in and navigated to Admin Dashboard with records	Pass
	Provide Valid Password	Password: admin			
	Click on Login button				
3	Provide room informations	Input room details	User will be redirected to dashboard	User will be redirected to dashboard	Pass
	Click on Update button				
	Provide invalid informations	Input invalid room details	User will be redirected to dashboard	User will be stay on that page showing error message	Pass
	Click on Update button				
Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database					

Code

```
package testcases;

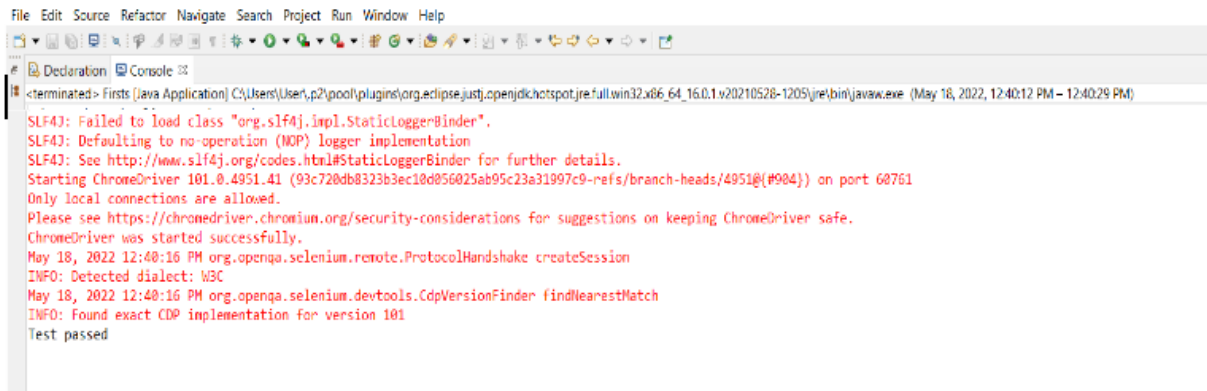
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import chromedriver.DriverSetup;

public class UpdateRoom {
    public static WebDriver driver;
    public static void main(String[] args) {
        driver = DriverSetup.getWebDriver("http://localhost/tol/login.php");
        driver.findElement(By.name("login")).click();
        driver.findElement(By.name("userName")).sendKeys("admin@ gmail.com");
        driver.findElement(By.name("pass word")).sendKeys("admin");
        driver.findElement(By.name("submitButton")).click();
        driver.get("http://localhost/tol/rooms.php");

        driver.findElement(By.name("name")).sendKeys("Suite Room");
        driver.findElement(By.name("image")).sendKeys("r1.jpg");
        driver.findElement(By.name("amount")).sendKeys("1500");
        driver.findElement(By.name("submit")).click();

        String actualUrl="http://localhost/tol/admin.php";
        String expectedUrl= driver.getCurrentUrl();
        if(actualUrl.equalsIgnoreCase(expectedUrl)) {
            System.out.println("Test passed");
        } else {
            System.out.println("Test failed");
        }
    }
}
```

Output



```
File Edit Source Refactor Navigate Search Project Run Window Help
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Starting ChromeDriver 101.0.4951.41 (93c720db8323b3ec18d056025ab95c23a31997c9-refs/branch-heads/4951@{#904}) on port 60761
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
May 18, 2022 12:40:16 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
May 18, 2022 12:40:16 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found exact CDP implementation for version 101
Test passed
```

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operation alone.

At this stage the main workload, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can create chaos and confusion.

Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after thorough testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system testing and changeover.

The implementation state involves the following tasks:

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.

6.2 IMPLEMENTATION PROCEDURES

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. In many

organizations someone who will not be operating it, will commission the software development project. In the initial stage people doubt about the software but we have to ensure that the resistance does not build up, as one has to make sure that:

- The active user must be aware of the benefits of using the new system.
- Their confidence in the software is built up.
- Proper guidance is imparted to the user so that he is comfortable in using the application.

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not up running on the server, the actual process won't take place.

User Training

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer based system, it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

Training on the Application Software

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the data entered. It should uncover information needed by the specific user/ group to use the system or part of the system while imparting the training of the program on the application. This training may be different across different user groups and across different levels of hierarchy

System Maintenance

Maintenance is the enigma of system development. The maintenance phase of the software cycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance

is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. Software maintenance is of course, far more than "Finding Mistakes".

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

It has been a great pleasure for me to work on this exciting and challenging project. This project proved good for me as it provided practical knowledge about the web development. It also provides knowledge about the latest technology used in developing web enabled application and client server technology that will be great demand in future. This will provide better opportunities and guidance in future in developing projects independently.

7.2 FUTURESCOPE

- The proposed system is designed in such a way that users can book treatments and rooms .
- Helps to find out common communities
- Communication through chats.
- Transferring data with high security
- Helps to have a virtual common area for doctors.
- Data security can be enhanced.

CHAPTER 8

BIBLIOGRAPHY

REFERENCES:

- Jon Duckett ., Java Script Programming, Wiley, 2014
- Bryan Basham, Kathy Sierra ., Head First EJB ,O'Reilly Media; Second edition, 2008
- Gary B. Shelly, Harry J. Rosenblatt, “System Analysis and Design”, 2009.
- Roger S Pressman, “Software Engineering”, 1994.
- PankajJalote, “Software engineering: a precise approach”, 2006.
- IEEE Std 1016 Recommended Practice for Software Design Descriptions.

WEBSITES:

- <https://www.tutorialspoint.com/css/index.htm>
- javatpoint.com/javascript-tutorial
- <https://www.w3schools.com/html/>
- <https://www.tutorialrepublic.com/php-tutorial/>

CHAPTER 9

APPENDIX

9.1 SAMPLE CODE

assign.php

```
<?php
include('dbconnect.php');
session_start();
if(isset($_POST['submit']))
{
    $uname=$_POST['name'];
    $mail=$_POST['email'];
    $mobile=$_POST['phone'];
    $cpassword=$_POST['cpassword'];
    $cpas=md5($cpassword);
    $sql="INSERT INTO `tbl_register` ( `reg_name`, `reg_email`, `reg_phone`, `password`, `status`)
        VALUES ('$uname','$mail','$mobile','$cpas',0)";
    if(mysqli_query($con,$sql))
    {
        $sql2= "SELECT * FROM `tbl_register` WHERE `reg_email`='$mail'";
        $data=mysqli_query($con,$sql2);
        if($res=mysqli_fetch_assoc($data))
        {
            $sql3="SELECT * FROM `tbl_usertype` WHERE `type_name`='Doctor'";
            $data1=mysqli_query($con,$sql3);
            if($row=mysqli_fetch_assoc($data1))
            {
                $reg=$res['reg_id'];
                $type=$row['type_id'];
                $sql4="INSERT INTO `tbl_login` ( `reg_id`, `type_id`) VALUES ('$reg','$type')";
                if(mysqli_query($con,$sql4))
                {
                    echo "success";
                }
            }
            else
            {
                echo mysqli_errno($con);
            }
        }
    }
}
```

```

}
}
}
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<!-- Boxicons -->
<link href='https://unpkg.com/boxicons@2.0.9/css/boxicons.min.css' rel='stylesheet'>
<!-- My CSS -->
<link rel="stylesheet" href="css/dashboard.css">
<title>Online Ayurvedic Wellness Resort </title>
</head>
<body>
<!-- SIDEBAR -->
<section id="sidebar">
<!-- LOGO -->
<a href="#" class="brand">
<br><br>
<imgsrc="images/logo.png" height="200" width="250" align="left">
</a>
<!-- SIDEBAR MENUS -->
<ul class="side-menu top">
<li class="active">
<a href="doctor.php">
<i class='bx bxs-dashboard'></i>
<span class="text">Dashboard</span>
</a>
</li>
<li>
<a href="treatment.php">
<i class='bx bxs-doughnut-chart'></i>
<span class="text">Ayurvedic Packages</span>

```

```

</a>
</li>
<li>
<a href="rooms.php">
<i class='bx bxs-message-dots'></i>
<span class="text">Rooms</span>
</a>
</li>
<li>
<a href="adddoctor.php">
<i class='bx bxs-group'></i>
<span class="text">Add Doctor</span>
</a>
</li>
</ul>
<ul class="side-menu">
<li>
<a href="login.php" class="logout">
<i class='bx bxs-log-out-circle'></i>
<span class="text">Logout</span>
</a>
</li>
</ul>
</section>
<!-- CONTENT -->
<section id="content">
<!-- MAIN -->
<main>
<table>
<div class="container">
<h2 style="color: #9f8e64;">ADD DOCTOR</h2><br>
<form method="POST" action="#" enctype="multipart/form-data" onsubmit="return validate();">
<label>Name:</label>
<input type="text" id="name" name="name" placeholder="Name">
<span style="color: red; margin-left:50px; font-size:12px"></span><br>
<label>Email:</label>
<input type="email" id="email" name="email" placeholder="Email">
<span style="color: red; margin-left:50px; font-size:12px"></span><br>

```

```

<label>Phone:</label>
<input type="text" id="phone" name="phone" placeholder="Phone number">
<span style="color: red; margin-left:50px; font-size:12px"></span><br>
<label>Password:</label>
<input type="password" id="cpassword" name="cpassword" placeholder="Password">
<input type="submit" name="submit" value="Submit">
  <span style="color: red; margin-left:250px; font-size:12px"></span><br>
</form>
</div>
</table>
</main>
</section>
<script src="js/script.js"></script>
</body>
</html>
<script type="text/javascript">
function validate() {
if (document.getElementById('name').value.length == 0 || document.getElementById('email').value.length ==
  0 ||
document.getElementById('phone').value.length == 0 || document.getElementById('cpassword').value.length
  == 0) {
span[8].innerText = "Complete the registration";
span[8].style.color = 'red';
return false;
}

}

let name = document.getElementById('name');
let span = document.getElementsByTagName('span');
let email = document.getElementById('email');
let phn = document.getElementById('phone');
let pass1 = document.getElementById('cpassword');
name.onkeyup = function() {
var regex = /^(.[\._a-zA-Z0-9]+)([a-zA-Z0-9 ]+){1,30}$/;
if (regex.test(name.value)) {
span[5].innerText = "";
span[5].style.color = '#28fc7a';
document.getElementById('mysubmit').disabled = false;
} else {

```

```
span[5].innerText = "enter a valid name";
span[5].style.color = 'red';
document.getElementById('mysubmit').disabled = true;
}
}
email.onkeyup = function() {
const regex = /^[^\_a-zA-Z0-9]+\)([a-zA-Z0-9]+\.[a-zA-Z0-9]){0,10}$/;
const regexo = /^[^\_a-zA-Z0-9]+\)([a-zA-Z0-9]+\.[a-zA-Z0-9]){0,10}\.[a-zA-Z0-9]{0,10}$/;
if (regex.test(email.value) || regexo.test(email.value)) {
span[6].innerText = "";
span[6].style.color = '#28fc7a';
document.getElementById('mysubmit').disabled = false;
} else {
span[6].innerText = "your email is invalid";
span[6].style.color = 'red';
document.getElementById('mysubmit').disabled = true;
}
}
phn.onkeyup = function() {
const regexn = /^[0-9]{10}$/;
if (regexn.test(phn.value)) {
span[7].innerText = "";
span[7].style.color = '#28fc7a';
document.getElementById('mysubmit').disabled = false;
} else {
span[7].innerText = "your number is invalid";
span[7].style.color = 'red';
document.getElementById('mysubmit').disabled = true;
}
}
window.onscroll = function() {
myFunction()
};
var navbar = document.getElementById("navbar");
var sticky = navbar.offsetTop;

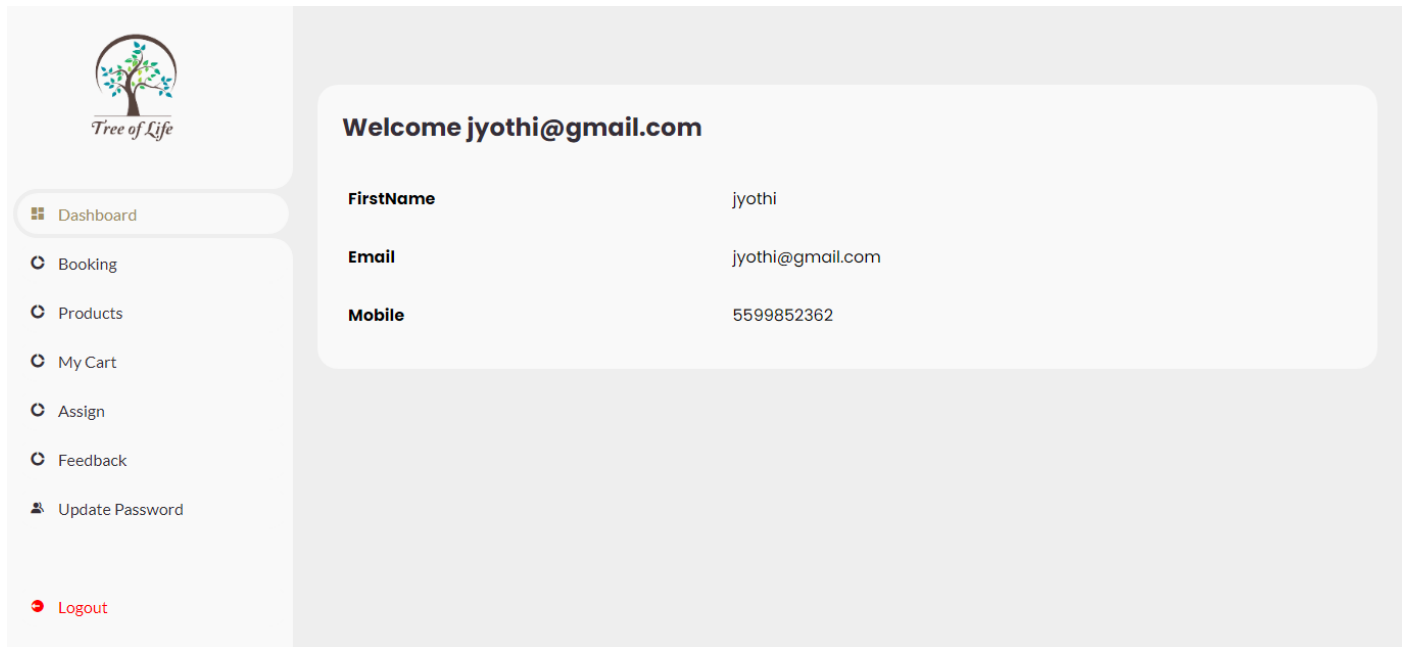
function myFunction() {
if (window.pageYOffset >= sticky) {
```

```
navbar.classList.add("sticky")
} else {
navbar.classList.remove("sticky");
}
}
</script>
```

9.2 SCREEN SHOTS

CUSTOMER PAGES

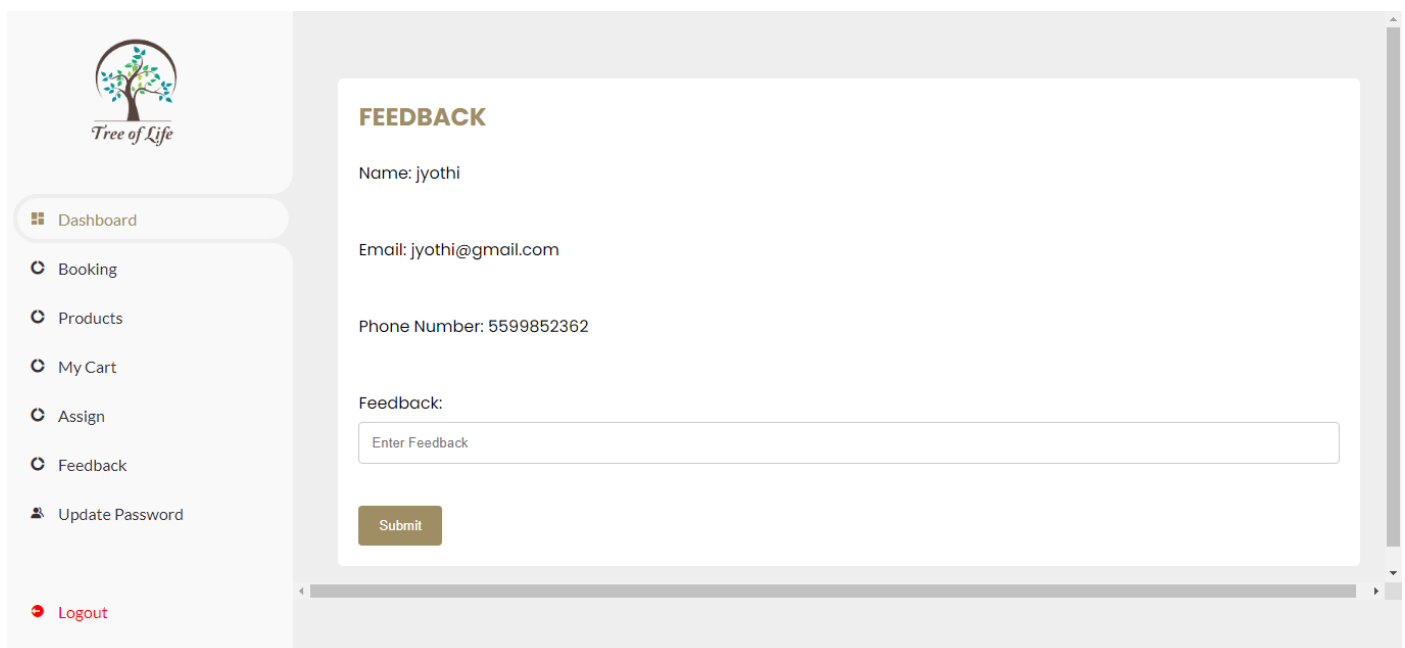
User Dashboard



Welcome jyothi@gmail.com

FirstName	jyothi
Email	jyothi@gmail.com
Mobile	5599852362

User Feedback



FEEDBACK

Name: jyothi

Email: jyothi@gmail.com

Phone Number: 5599852362

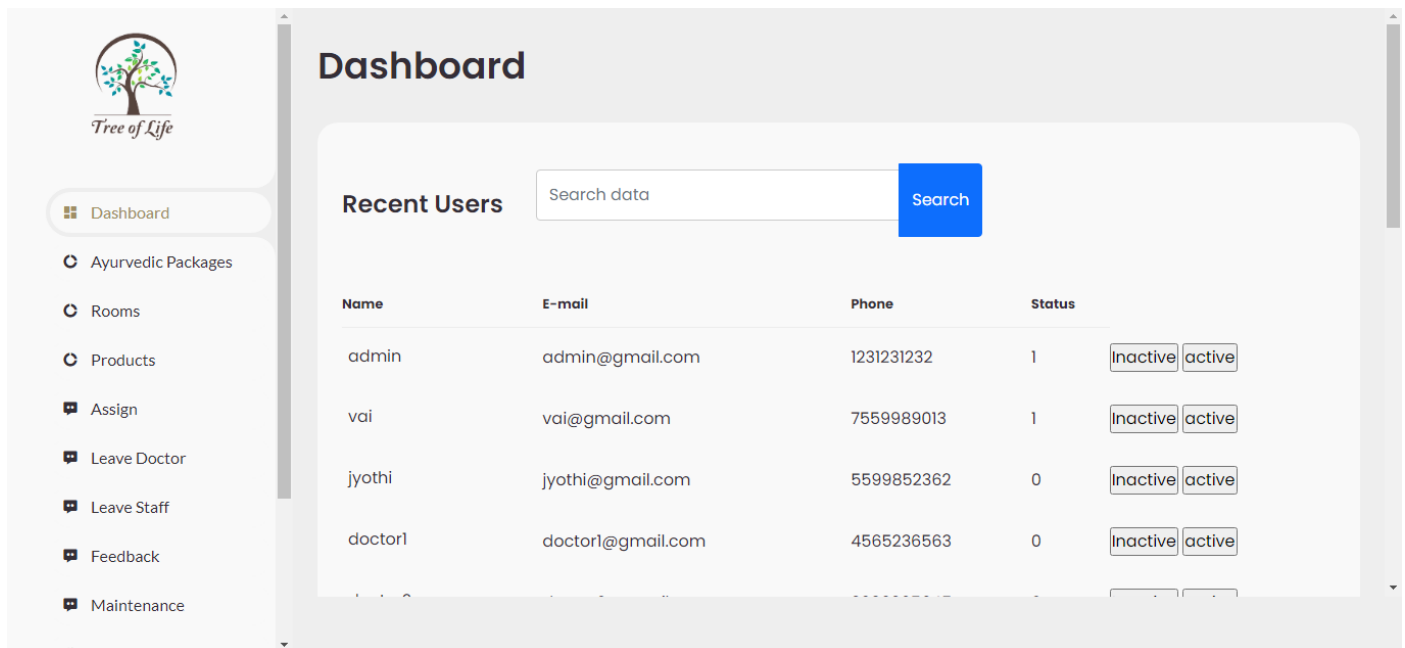
Feedback:

Enter Feedback

Submit

ADMIN PAGES

Admin Dashboard



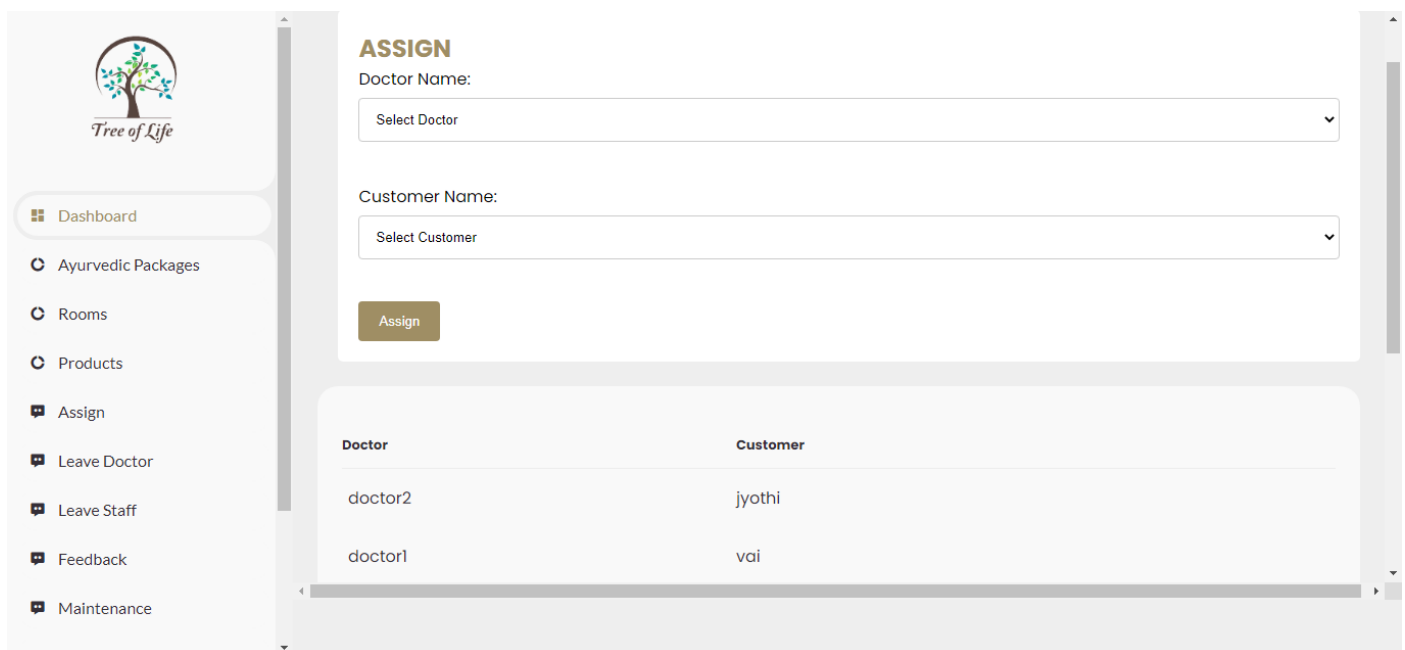
Dashboard

Recent Users

Search data

Name	E-mail	Phone	Status
admin	admin@gmail.com	1231231232	1 <input type="button" value="inactive"/> <input type="button" value="active"/>
vai	vai@gmail.com	7559989013	1 <input type="button" value="inactive"/> <input type="button" value="active"/>
jyothi	jyothi@gmail.com	5599852362	0 <input type="button" value="inactive"/> <input type="button" value="active"/>
doctor1	doctor1@gmail.com	4565236563	0 <input type="button" value="inactive"/> <input type="button" value="active"/>

Assign



ASSIGN

Doctor Name:

Select Doctor


Customer Name:

Select Customer

Doctor	Customer
doctor2	jyothi
doctor1	vai

DOCTOR PAGE

Assigned Users



Tree of Life


- Dashboard
- Assign
- Prescription
- Leave
- Add Details
- Update Password
- Logout

YOUR PATIENTS

Name	Email	Phone
jyothi	jyothi@gmail.com	5599852362
mannu	mannnu@gmail.com	7559989013
meenu philip	meenu@gmail.com	8129517046

STAFF PAGE

Apply Leave



Tree of Life

- Dashboard
- Maintenance
- Leave
- Leave Status
- Add Details
- Update Password
- Logout

APPLY LEAVE

Name: staff1

Email: staff1@gmail.com

Phone Number: 3656563445

Leave Date:

Reason:

Submit