

AYURVEDIC SOLUTION EXPERT

PROJECT REPORT

Submitted to the Calicut University in partial fulfilment of the requirement
for the award of the degree of

Bachelor of Computer Application (BCA)

By

SHIFNA SHERIN (ARATBCA005)

Under the guidance of

Ms. PRIYA P P

and

Ms. BASILA T M



PG DEPARTMENT OF COMPUTER SCIENCE

ANSAR WOMEN'S COLLEGE, PERUMPILAVU

2019-2022



ANSAR WOMEN'S COLLEGE, PERUMPILAVU

PG DEPARTMENT OF COMPUTER SCIENCE

CERTIFICATE

This is to certify that **Ms. SHIFNA SHERIN** (ARATBCA005) BCA students of ANSAR WOMEN'S College, Perumpilavu, Thrissur, has successfully completed the project work entitled "**AYURVEDIC SOLUTION EXPERT**" in PYTHON. As part of their curriculum under the guidance of **Ms. PRIYA P P** (Asst. Prof. Department of C S) and **Ms. BASILA T M** (Asst. Prof. Department of C S) during the period 2019 to 2022.

Ms. PRIYA P P:

Ms. C M SULAIKHA:

Ms. BASILA T M:

HoD, Department of C S

Asst. Prof. Department of C S

The presentation and viva-voce examination was conducted onat Ansar Women's College.

Internal Examiner:

External Examiner:

DATE:

DECLARATION

I, **SHIFNA SHERIN** of sixth semester, Bachelor of Computer Application, do hereby declare that the project report entitled “**AYURVEDIC SOLUTION EXPERT**”, is the result of my original work done by we under the guidance of **Ms. PRIYA P P** (Asst. Prof. Department of C S) and **Ms. BASILA T M** (Asst. Prof. Department of C S).

We also declare that this work has not been submitted earlier by we to University of Calicut or any other institutions for the fulfilment of the requirement of the course of study. The imperial findings in the report are based on the data collected by ourselves.

Place : PERUMPILAVU

Date:

SHIFNA SHERIN (ARATBCA005)

Signature:

ACKNOWLEDGMENT

The effort taken by the completion of this project would not have possible without the kind of support and help rendered by the many individuals and organization. We would like to acknowledge our sincere thanks to all of them. First, we would like to thank God Almighty for his glorious blessings which have been accompanied throughout the course of our work. We firmly believe it was with his help that we are able to complete this project successfully.

We highly indebted to **Dr. KAMALUDHEEN K T**, Our **Principal**, for his guidance and constant supervision as well as, providing all facilities to carry out our project as part of the curriculum. We extend our heartfelt gratitude to **Ms. C M SULAIKHA**, Head of the department of Computer Science, for providing us an excellent opportunity to work on this project. And also for her valuable support, help and inspiration for making this project a great success.

We express our deepest sense of gratitude and thanks to our supervising guides **Ms. PRIYA P P** (Asst. Prof. Department of C S) and **Ms. BASILA T M** (Asst. Prof. Department of C S), for their patience and support together with proper guidance throughout the completion of our project. We like to express our special gratitude and thanks to **Ms. LIYAT P S**, staff of OCIUZ Infotech Pvt.Ltd, Thrissur, for providing the sufficient guidelines and facilities in the lab. Words are inadequate to express our gratitude to our beloved parents and friends for their never ending encouragement throughout our work.

With thanks and regards,

SHIFNA SHERIN (ARATBCA005)

CONTENTS

1.0	INTRODUCTION	1
1.1	SYNOPSIS	2
1.2	ABOUT THE PROJECT	2
1.3	MAIN OBJECTIVE OF PROJECT	2
2.0	REQUIREMENT SPECIFICATION	4
2.2	SOFTWARE SPECIFICATION	5
2.1	HARDWARE SPECIFICATION	17
3.0	PROJECT PLANNING AND SHEDULING	18
3.1	PERT CHART	20
3.2	GANTT CHART	21
4.0	SYSTEM ANALYSIS	22
4.1	OVERVIEW	23
4.2	FEASIBILITY ANALYSIS	23
4.2.1	TECHNICAL FEASIBILITY	25
4.2.2	ECONOMICAL FEASIBILITY	25
4.2.3	OPERATIONAL FEASIBILITY	27
5.0	SYSTEM DESIGN	28
5.1	INPUT DESIGN	29
5.2	OUTPUT DESIGN	29

5.3	DATAFLOW DIAGRAM	30
5.4	DATABASE DESIGN	34
6.0	SYSTEM TESTING AND IMPLEMENTATION	39
6.1	SYSTEM TESTING	40
6.2	SYSTEM IMPLEMENTATION	45
6.3	SYSTEM MAINTENANCE	45
7.0	FUTURE ENHANCEMENT	48
8.0	CONCLUSION	49
9.0	BIBLIOGRAPHY	50
10.0	APPENDIX	51
10.1	SAMPLE CODE	52
10.2	SCREEN SHOTS	58

1.0 INTRODUCTION

1.1 SYNOPSIS

Ayurveda blends our modern lifestyle and health-oriented habits with the ancient wisdom of using natural substances, medicines and herbs to help us lead a healthy, happy, stress-free and disease-free life. The cardinal aim of Ayurveda is to restore the individual balance between mind, body and spirit. Ayurveda, a natural system of medicine, originated in India more than 3,000 years ago. The term Ayurveda is derived from the Sanskrit words Ayur (life) and Veda (science or knowledge). Thus, Ayurveda translates to knowledge of life. Based on the idea that disease is due to an imbalance or stress in a person's consciousness, Ayurveda encourages certain lifestyle interventions and natural therapies to regain a balance between the body, mind, spirit, and the environment.

1.2 ABOUT THE PROJECT

Ayurveda is a website where you learn, what is Ayurveda, its glory, strength and power. It is a store house of various facts relating to Ayurveda like, hospitals, school and herbal treatments, wide range of beauty products and tips, herbs and medicinal plants, home remedies for various diseases, therapy centres and resorts and lots more. Also Ayurveda hospitals are listed out and the doctors providing their services at these hospitals. Ayurveda therapy centres are places where treatment is done at a pleasant and healthy environment with all the beauty and power of nature. In Ayurveda website, different types of users are there. Mainly there will be an administrator to control the website. Also there will be doctor and normal users. Students can also be a part of this website for applying online courses provided in the website. Apart from treatment we get from hospitals and clinics, there are a lot of methods we can use at home for our diseases and for our well-being. Various such home remedies keep us safe from many diseases and illnesses. And here we list many of such home remedies for various purposes like Acidity, Acne, Arthritis, Asthma, Backache, Bad breath, Diabetes, Earache, Headache, pimples, etc. This website also provide facility for the students to apply courses related to Ayurveda.

1.3 MAIN OBJECTIVE OF THE PROJECT

The following are the objectives:

1. Provide online consultation platform.
2. Administrator controls the website.
3. Home remedies keep us safe from many diseases and illnesses.
4. Online consultation request processing.

5. Help to doctor registration.
6. Provide platform for online purchasing of medicines.
7. Provide facility for the students to apply courses related to Ayurveda.
8. Users can make online purchase of Ayurvedic products and plants.

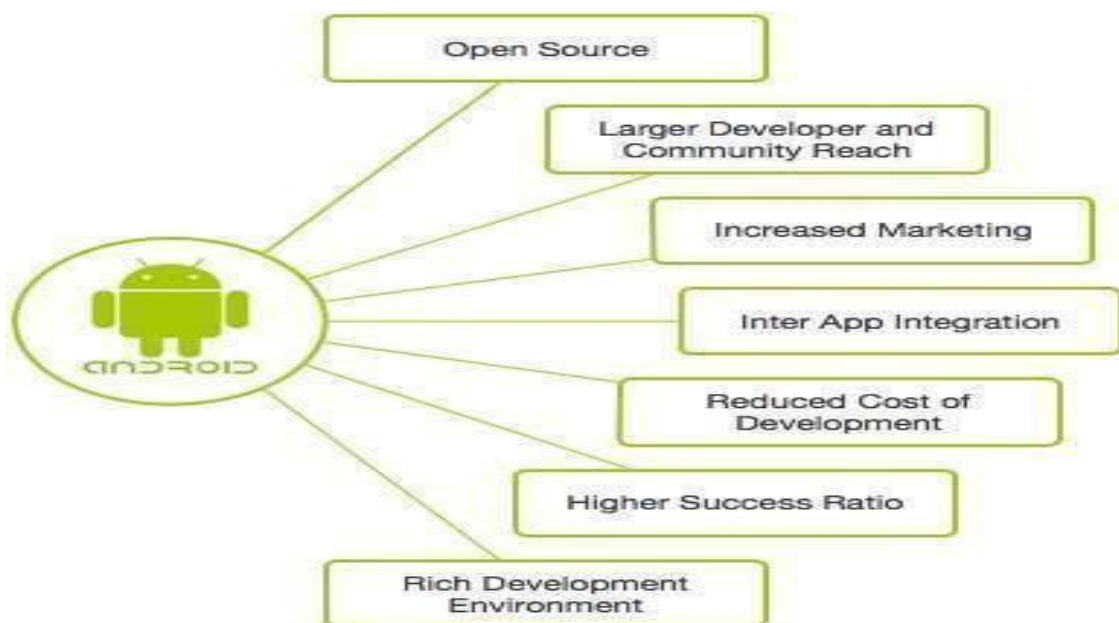
2. REQUIREMENT SPECIFICATION

2.1 SOFTWARE SPECIFICATION

Introduction to Android (Front end)

Android is an open source and Linux -based Operating System for mobile devices such as smartphones and tablet computers. Android was developed by the Open Handset Alliance, led by Google, and other companies. Android offers a unified approach to application development for mobile devices which means developers need only develop for Android, and their applications should be able to run on different devices powered by Android. The first beta version of the Android Software Development Kit (SDK) was released by Google in 2007 where as the first commercial version, Android 1.0, was released in September 2008. On June 27, 2012, at the Google I/O conference, Google announced the next Android version, 4.1 Jelly Bean. Jelly Bean is an incremental update, with the primary aim of improving the user interface, both in terms of functionality and performance. The source code for Android is available under free and open-source software licenses. Google publishes most of the code under the Apache License version 2.0 and the rest, Linux kernel changes, under the GNU General Public License version 2.

Why Android



Feature of Android:

- **Beautiful UI** - Android OS basic screen provides a beautiful and intuitive user interface.
- **Connectivity** - GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC and WiMAX.
- **Storage** - SQLite, a lightweight relational database, is used for data storage purposes.
- **Media support** - H.263, H.264, MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC, AAC 5.1, MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF, and BMP.
- **Messaging** - SMS and MMS.
- **Web browser** - Based on the open-source WebKit layout engine, coupled with Chrome's V8 JavaScript engine supporting HTML5 and CSS3.
- **Multi-touch** - Android has native support for multi-touch which was initially made available in handsets such as the HTC Hero.
- **Multi-tasking** - User can jump from one task to another and same time various application can run simultaneously.
- **Resizable widgets** - Widgets are resizable, so users can expand them to show more content or shrink them to save space.
- **Multi-Language** - Supports single direction and bi-directional text.
- **GCM** - Google Cloud Messaging (GCM) is a service that lets developers send short message data to their users on Android devices, without needing a proprietary sync solution.
- **Wi-Fi Direct** - A technology that lets apps discover and pair directly, over a highbandwidth peer-to-peer connection.
- **Android Beam** - A popular NFC-based technology that lets users instantly share, just by touching two NFC-enabled phones together.

Android is used for:

Android applications are usually developed in the Java language using the Android Software Development Kit.

Once developed, Android applications can be packaged easily and sold out either through a store such as **Google Play**, **SlideME**, **Opera Mobile Store**, **Mobango**, **F-droid** and the **Amazon Appstore**.

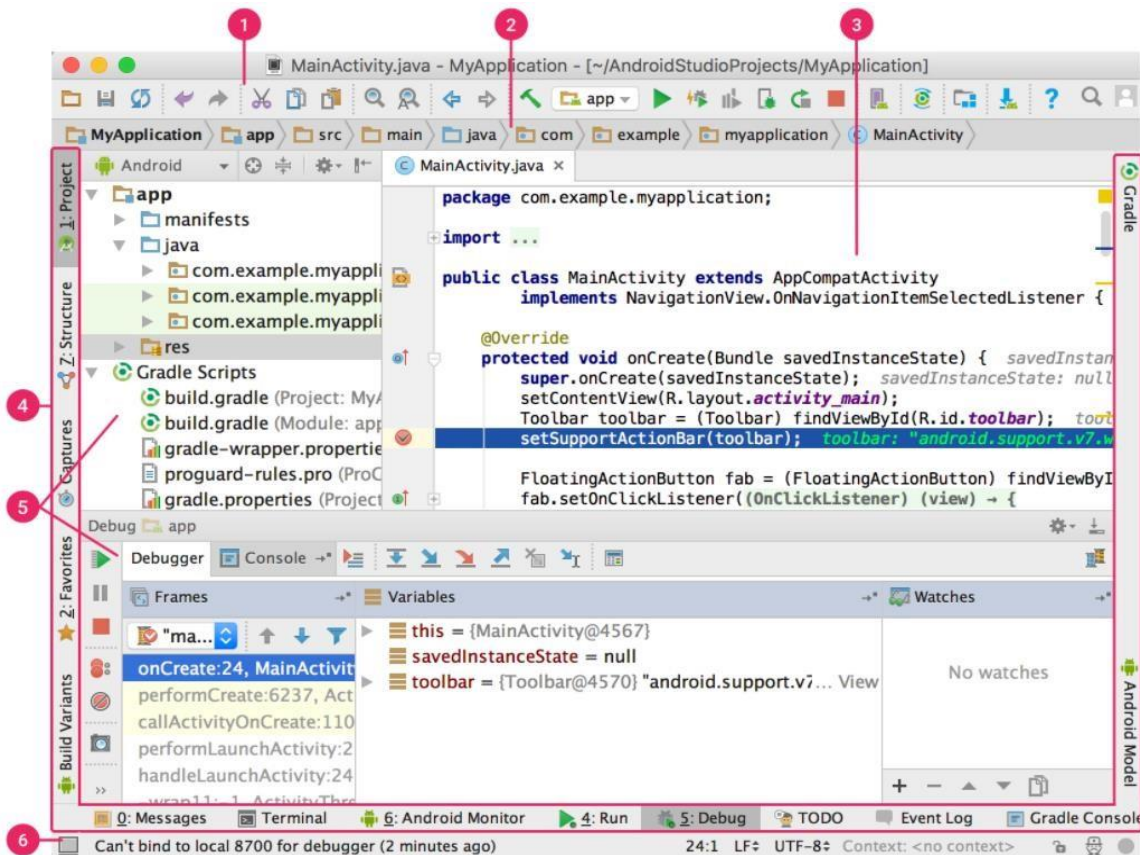
Android powers hundreds of millions of mobile devices in more than 190 countries around the world. It's the largest installed base of any mobile platform and growing fast. Every day more than 1 million new Android devices are activated worldwide.

This tutorial has been written with an aim to teach you how to develop and package Android application. We will start from environment setup for Android application programming and then drill down to look into various aspects of Android applications.

ANDROID STUDIO

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on [IntelliJ IDEA](#). On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

- A flexible Gradle-based build system
- A fast and feature-rich emulator
- A unified environment where you can develop for all Android devices
- Apply Changes to push code and resource changes to your running app without restarting your app
- Code templates and GitHub integration to help you build common app features and import sample code
- Extensive testing tools and frameworks
- Lint tools to catch performance, usability, version compatibility, and other problems
- C++ and NDK support
- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine.



- 1 The **toolbar** lets you carry out a wide range of actions, including running your app and launching Android tools.
- 2 The **navigation bar** helps you navigate through your project and open files for editing. It provides a more compact view of the structure visible in the **Project** window.
- 3 The **editor window** is where you create and modify code. Depending on the current file type, the editor can change. For example, when viewing a layout file, the editor displays the Layout Editor.
- 4 The **tool window bar** runs around the outside of the IDE window and contains the buttons that allow you to expand or collapse individual tool windows.
- 5 The **tool windows** give you access to specific tasks like project management, search, version control, and more. You can expand them and collapse them.
- 6 The **status bar** displays the status of your project and the IDE itself, as well as any warnings or messages.

Introduction to Python(Front-end)

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales.

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL).

Python is a popular programming language. It was created in 1991 by Guido van Rossum. It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** – You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** – Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** – Python is a great language for the beginner-level programmers and supports the development of a wide range of application.

Python's features include

- **Easy-to-learn** – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** – Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** – Python's source code is fairly easy-to-maintain.
- **A broad standard library** – Python's bulk of the library is very portable and crossplatform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** – You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** – Python provides interfaces to all major commercial databases.
- **GUI Programming** – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** – Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below –

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.

It supports automatic garbage collection.

Features of Python Programming

- A simple language which is easier to learn

Python has a very simple and elegant syntax. It's much easier to read and write Python programs compared to other languages like: C++, Java, C#. Python makes programming fun and allows you to focus on the solution rather than syntax. If you are a newbie, it's a great choice to start your journey with Python.

- Free and open-source

You can freely use and distribute Python, even for commercial use. Not only can you use and distribute software written in it, you can even make changes to the Python's source code. Python has a large community constantly improving it in each iteration.

- Portability

You can move Python programs from one platform to another, and run it without any changes. It runs seamlessly on almost all platforms including Windows, Mac OS X and Linux.

- Extensible and Embeddable

Suppose an application requires high performance. You can easily combine pieces of C/C++ or other language with Python code. This will give your application high performance as well as scripting capabilities which other languages may not provide out of the box.

Likewise, when you run Python code, it automatically converts your code to the language your computer understands. You don't need to worry about any lower-level operations.

A high-level, interpreted language. Unlike C/C++, you don't have to worry about daunting tasks like memory management, garbage collection and so on.

- Object-oriented

Everything in Python is an object. Object oriented programming

(OOP) helps you solve a complex problem intuitively.

With OOP, you are able to divide these complex problems into smaller sets by creating objects.

Script Mode Programming

Invoking the interpreter with a script parameter begins execution of the script and continues until the script is finished. When the script is finished, the interpreter is no longer active.

Let us write a simple Python program in a script. Python files have extension **.py**. Type the following source code in a test.py file –

Python Identifiers

A Python identifier is a name used to identify a variable, function, class, module or other object. An identifier starts with a letter A to Z or a to z or an underscore (_) followed by zero or more letters, underscores and digits (0 to 9).

Python does not allow punctuation characters such as @, \$, and % within identifiers. Python is a case sensitive programming language. Thus, **Manpower** and **manpower** are two different identifiers in Python.

Here are naming conventions for Python identifiers –

- Class names start with an uppercase letter. All other identifiers start with a lowercase letter.
- Starting an identifier with a single leading underscore indicates that the identifier is private.
- Starting an identifier with two leading underscores indicates a strongly private identifier.
- If the identifier also ends with two trailing underscores, the identifier is a language-defined special name.

APPLICATIONS OF PYTHON

Web Applications

You can create scalable Web Apps using frameworks and CMS (Content Management System) that are built on Python. Some of the popular platforms for creating Web Apps are: Django, Flask, Pyramid, Plone, Django CMS.

Sites like Mozilla, Reddit, Instagram and PBS are written in Python.

Scientific and Numeric Computing

There are numerous libraries available in Python for scientific and numeric computing. There are libraries like: SciPy and NumPy that are used in general purpose computing. And, there are specific libraries like: EarthPy for earth science, AstroPy for Astronomy and so on.

Also, the language is heavily used in machine learning, data mining and deep learning.

Creating software Prototypes

Python is slow compared to compiled languages like C++ and Java. It might not be a good choice if resources are limited and efficiency is a must.

choice if resources are limited and efficiency is a must.

However, Python is a great language for creating prototypes. For example: You can use Pygame (library for creating games) to create your game's prototype first. If you like the prototype, you can use language like C++ to create the actual game.

Good Language to Teach Programming

Python is used by many companies to teach programming to kids and newbies.

Django

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

Django is a high-level Python Web framework that encourages rapid development and clean pragmatic design. A Web framework is a set of components that provide a standard way to develop websites fast and easily. Django's primary goal is to ease the creation of complex database-driven websites. Some well known sites that use Django include PBS, Instagram, Disqus, Washington Times, Bitbucket and Mozilla

Django (/ˈdʒæŋɡoʊ/ *jang-goh*) is a free and open source web application framework, written in Python. A web framework is a set of components that helps you to develop websites faster and easier.

Luckily for you, other people long ago noticed that web developers face similar problems when building a new site, so they teamed up and created frameworks (Django being one of them) that give you ready-made components to use.

Frameworks exist to save you from having to reinvent the wheel and to help alleviate some of the overhead when you're building a new site.

Why do you need a framework?

To understand what Django is actually for, we need to take a closer look at the servers. The first thing is that the server needs to know that you want it to serve you a web page.

Imagine a mailbox (port) which is monitored for incoming letters (requests). This is done by a web server. The web server reads the letter and then sends a response with a webpage. But when you want to send something, you need to have some content. And Django is something that helps you create the content.

When a request comes to a web server, it's passed to Django which tries to figure out what is actually requested. It takes a web page address first and tries to figure out what to do. This part is done by Django's **urlresolver** (note that a website address is called a URL – Uniform Resource Locator – so the name *urlresolver* makes sense). It is not very smart – it takes a list of patterns and tries to match the URL. Django checks patterns from top to bottom and if something is matched, then Django passes the request to the associated function (which is called *view*).

Imagine a mail carrier with a letter. She is walking down the street and checks each house number against the one on the letter. If it matches, she puts the letter there. This is how the URL resolver works!

In the *view* function, all the interesting things are done: we can look at a database to look for some information. Maybe the user asked to change something in the data? Like a letter saying, "Please change the description of my job." The *view* can check if you are allowed to do that, then update the job description for you and send back a message: "Done!" Then the *view* generates a response and Django can send it to the user's web browser.

The description above is a little bit simplified, but you don't need to know all the technical things yet. Having a general idea is enough.

So instead of diving too much into details, we will start creating something with Django and we will learn all the important parts along the way!

Django is a free and open-source web application framework written in Python. A framework is nothing more than a collection of modules that make development easier. They are grouped together, and allow you to create applications or websites from an existing source.

This is how websites - even simple ones designed by a single person - can still include advanced functionality like authentication support, management and admin panels, contact forms, comment boxes, file upload support, and more. In other words, if you were creating a website from scratch you would need to develop these components yourself. By using a framework instead, these components are already built, you just need to configure them properly to match your site.

Report

Django as "a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel.

Django offers a big collection of modules which you can use in your own projects. Primarily, frameworks exist to save developers a lot of wasted time and headaches and Django is no different.

You might also be interested in learning that Django was created with front-end developers in mind. "Django's template language is designed to feel comfortable and easy-to-learn to those used to working with HTML, like designers and front-end developers. But it is also flexible and highly extensible, allowing developers to augment the template language as needed."

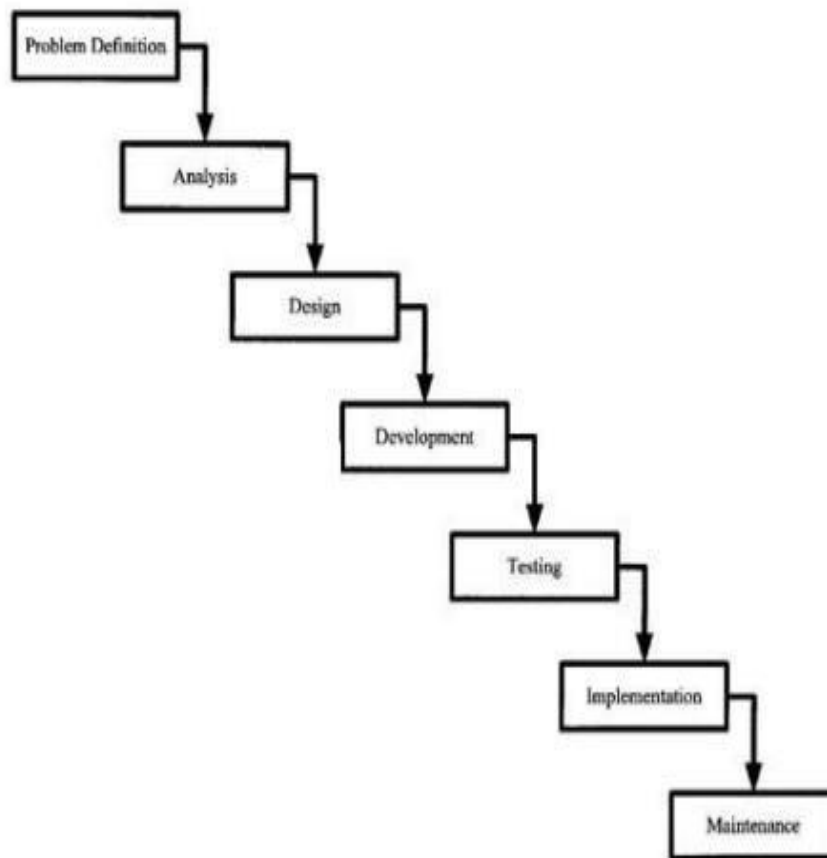
2.2 HARDWARE SPECIFICATION

In order to implement a new system the choice of a processor with maximum possible speed is made. There should be sufficient memory to store data and software tools for efficient processing.

- Processor : Intel Core i3
- RAM : 4GB
- Hard Disk : PC with 50GB
CPU Speed : 1.50 GHz
- Monitor : VGA Color
- Mouse : Two Button Roller Mouse Recommended
- Modem: Any Modem
- Keyboard: Standard Keyboard

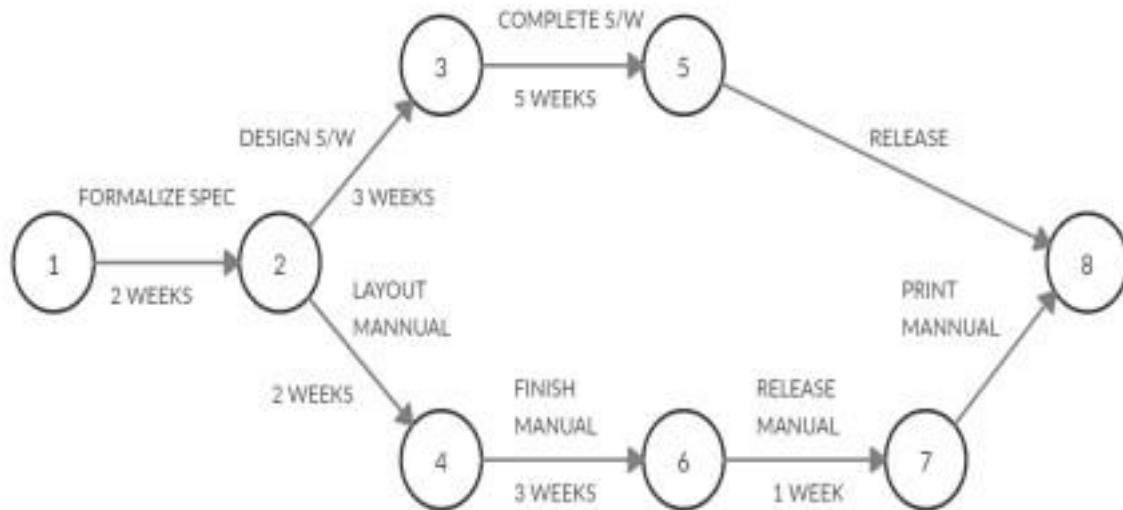
3. PROJECT PLANNING AND SHEDULING

The design starts after the requirement analysis is complete and the coding begins after the design is complete. Once the programming is completed, the testing is done. In this model the sequence of activities performed in a software development project are:



It was a well planned and well executed job. The details and the requirements for the system was collected from the client itself and all were properly tabulated. Then the next phase was to design all the outputs given by the client to well manageable design which have all the flexibility and which should withstand the current and upcoming technologies. More one design was made and from that a well acceptable one which got all the functionalities of the user requirements was selected. And we gathered information about the front ends and back ends that will suit for our job.

PERT CHART



STEPS



GANTT CHART

Tasks	Duration	NOV 1				DEC 2				JAN 3				FEB 4			MAR 5		
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	1	2	3
Project	5months																		
System study	2weeks																		
System analysis	4weeks																		
Design	3weeks																		
Coding and Testing	4weeks																		
Implementation	1 week																		

4. SYSTEM ANALYSIS

System Analysis is conducted with the following objectives.

- Identify the customers need.
- Evaluate the system concept for feasibility.
- Perform economic analysis.
- Technical analysis.
- Allocate function to hardware, software, people, database and other system elements.
- Establish cost and schedule constraints.
- Create system definition that forms the foundation for all subsequent engineering work.

4.1 OVERVIEW

System analysis is concerned with studying the existing system identifying the problems and creation of the requirement specification document. In the system analysis I study about the system to be computerized and the entire problem with the present are pin pointed.

The system analysis was planned and conducted in 3 stages, involving an initial investigation, feasibility and detailed analysis.

4.2 FEASIBILITY ANALYSIS

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 culminate in a written and oral feasibility report. 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.

To perform the feasibility study, the software engineer must first analyze the problem at global level. Indeed, the more the problem is understood, the better alternative solutions, the cost and their potential benefits for the user can be identified. Therefore, ideally, one should perform as much analysis of the problem as is needed to do a well-founded feasibility study. Since software developers cannot be sure that the offer will be accepted, they have a limited incentive for investing resource into analyzing the program. On the other hand, if the study produces results that are inaccurate, it may underestimate the resource needed to develop the application and that will result in serious budget problem.

The requirements analysis phase of system and take advantages of the opportunities identified during scope definition and it satisfies the requirement identified in the requirements analysis phase of system development. An estimate is made of whether the identified user needs may be satisfied using current software and hardware technologies. The study will decide if the proposed system will be cost effective from a business point study will decide if the proposed system will be cost effective from a business point of view and if it can be developed given existing budgetary constraints. The key considerations involved in the feasibility analysis are economic, technical, behavioral and operational. Three tests for project feasibility namely, Technical, Economical, and Operational feasibilities.

4.2.1 TECHNICAL FEASIBILITY

Technical feasibility determines whether the work for the project can be done with the existing equipment, software technology and available personnel. Technical feasibility is concerned with specifying equipment and software that will satisfy the user requirement. This project is feasible on technical remarks also, as the proposed system is more beneficiary in terms of having a sound proof system with new technical components installed on the system. The proposed system can run on any machines supporting Windows and Internet services and works on the best software and hardware that had been used while designing the

As far as our project is concerned, we have the necessary expertise so that the proposed solution can be made feasible. Some projects are initiated with specific deadlines. In our case first we have given three months' time but due to some problems regarding time and the constraints of expertise it has been extended to six months. Now there are some organizational constraints that have not yet given us the opportunity to install the system.

4.2.2 ECONOMICAL FEASIBILITY

Economical feasibility determines whether there are sufficient benefits in sufficient benefits in creating to make the cost acceptable, or is the cost of the system too high. As this signifies cost-benefit analysis and savings. On the behalf of the cost-benefit analysis, the proposed system is feasible and is economical regarding its pre-assumed cost for making a system.

Economical feasibility has great importance as it can outweigh other feasibilities because costs affect organization decisions. The concept of Economic Feasibility deals with the fact that a system that can be developed and will be used on installation must be profitable for the Organization. The cost to conduct a full system investigation, the cost of hardware and software, the benefits in the form of reduced expenditure are all discussed during the economic feasibility.

During the economical feasibility test we maintained the balance between the Operational and Economical feasibilities, as the two were conflicting. For example the solution that provides the best operational impact for the end-users may also be the most expensive and, therefore, the least economically feasible.

As we know that the system development costs are usually one-time costs that will not recur after the project has been completed. For calculating the Development costs we evaluated certain cost categories viz.

- (i) Personnel costs
- (ii) Computer usage
- (iii) Training
- (iv) Supply and equipment costs
- (v) Cost of any new computer equipment and software.

In order to test whether the Proposed System is cost-effective or not we evaluated it through three techniques viz.

- Payback analysis
- Return on Investment
- Net present value

4.2.3 OPERATIONAL FEASIBILITY

Operation feasibility is a measure of how people feel about the system. Operational Feasibility criteria measure the urgency of the problem or the acceptability of a solution. Operational Feasibility is dependent upon determining human resources for the project. It refers to projecting whether the system will operate and be used once it is installed.

If the ultimate users are comfortable with the present system and they see no problem with its continuance, then resistance to its operation will be zero. Behaviorally also the proposed system is feasible. A particular application may be technically and but may fail to produce the forecasted benefits, because the company is not able to get it to work. For the system, it is not necessary that the user must be a computer expert, but any computer operator given a little bit of knowledge and training can easily operate. Our Project is operationally feasible since there is no need for special training of staff member and whatever little instructing on this system is required can be done so quite easily and quickly as it is essentially This project is being developed keeping in mind the general people who one have very little knowledge of computer operation, but can easily access their required database and other related information. The redundancies can be decreased to a large extent as the system will be fully automated.

5 SYSTEM DESIGN

5.1 INPUT DESIGN

The input design is the link between the computers and the users. It composes developing specification and procedure for data preparation and those steps that are necessary to put transaction data into a usable form for processing data entry. Input design is one of the most expensive phase of the development of a system. A large number of problem with a system can usually be raised due to the fault input design and method. Therefore needed to say, the input data is the life block of a system and has to be analyzed and designed with the most consideration.

- The decision made during input design must help.
- To provide cost effective method of input.
- To achieve highest level of accuracy.

System analysis decide the following input design details like, what data item to input, how the data should be arranged, coded data items and transactions needing validations to detect errors and finally dialogue to guide the users in providing input. The design of input involves identifying the data needed, specifying the characteristic of each data item, capturing and for computer processing and ensuring correctness of data.

5.2 OUTPUT DESIGN

The output design refers to the result and information that are generated by the system. For many users output is the main reason for developing a system and the basis on which they evaluate the usefulness of an application.

The objective of a system leads to the determination of output. The analysis of a system leads to the determination of output. Output of a system can take various forms. The most common reports are, screen displays, printed forms, graphical forms etc. the output also varies in terms of their contents, frequency, timing and format. The users of the output, its purpose and sequence of details to be printed are all considered. The output form a system is the justification for its existence. If the output is inadequate in anyway, the system itself in inadequate. The basics requirements of the output are that it should be accurate, timely and appropriate, in terms

of content, medium and layout for its intended purpose. Hence it is necessary to design output so that the objective of the system is met in the best possible manner.

5.3 DATAFLOW DIAGRAM

A data flow diagram (DFD) is graphical representation of the “flow” of data through an information system. DFD can also be used for the visualization of the data processing (structured design). A DFD provides no information about the timing or ordering of processes, or about whether processes will operate in sequence or in parallel. Dataflow diagrams can be used to provide the end user with physical idea of where the data they input ultimately has an effect upon the structure of the whole system from order to dispatch to record. How many systems are developed can be determined through a data flow diagram. The circle or bubble represents a transformation process and the label inside the bubble describes the process, using an active verb. Data flows are directed lines that identify the input dataflow and output data flows at each process bubble. Data storage is represented by an open-ended rectangle with a label that identifies the data store or file. The square is labelled to identify an external entity that is a source or destination of a data flow.

Data flow diagrams are used to define the flow of the system and its resources such as information. DFD are way of expressing system requirements in a graphical manner. DFD represents one of the most ingenious tools used for the structured analysis. It has the purpose of clarifying system requirements and identifying major transformation that will become programs in the system design. It is the major point in the design phase that functionality decomposes the requirements specification down the lowest level of details.

Flow diagrams in general are usually designed using designed using simple symbols such as a rectangle, an oval or a circle depicting a processes, data stored or an external entity, and arrows are generally used to depicts the data flow one step to another. Data flow diagrams present the logical flow of information through a system in graphical or pictorial form.

Basic Dataflow Diagram Symbols

- **Dataflow**

Used to represent functions or process Packets of data to travel from one point to another. Data may flow from a source to process and from data store or process. An arrow line depicts the flow, with arrow head pointing in the direction of the flow. Arrowhead Indicating the direction in which the data is flowing. Can have two arrow heads when a process is altering existing records in a data store.

- **Process**

Circle stands for processes that convert data into information. A process represents transformation where incoming data flows. A data flow is a route, which enables are changed into outgoing data flows. Must have at least one input and at least one output. When naming process, avoid glossing over them, without really understanding their roles.

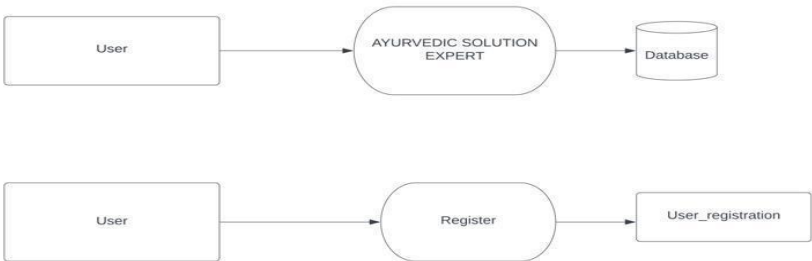
- **Data store**

A data store is a repository of data that is to be stored for use by a one or more process may be as simple as buffer or queue or sophisticated as relational database. Tjey should have clear name. If a process merely uses the content of store and does not alter it.

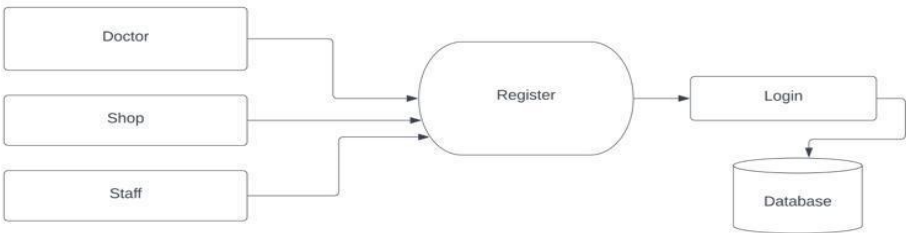
- **Source/Destination**

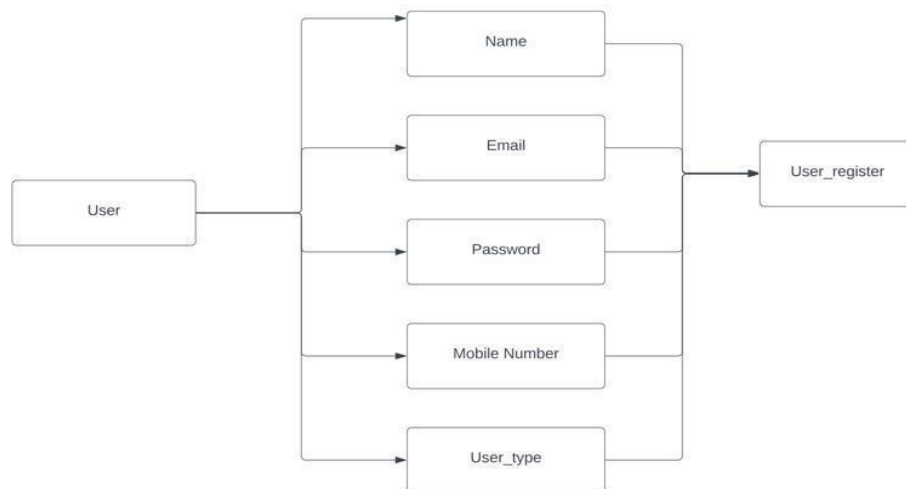
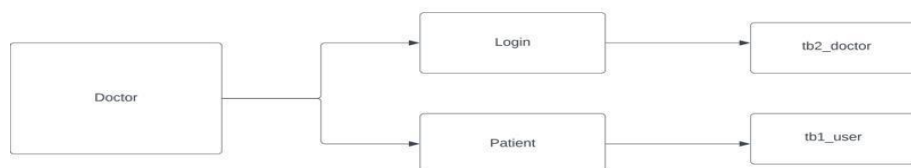
A source or sink is a person or part of an organization which enters or receives information from the system, but is considered to be outside the content of data flow model. It is normal for all the information represented with a system to have been obtained from or to be passed onto an external source or recipient. These external entities may be duplicated on a diagram, to avoid crossing dataflow lines, they are duplicated a stripe is drawn across the left hand corner.

CONTEXT LEVEL (LEVEL 0):

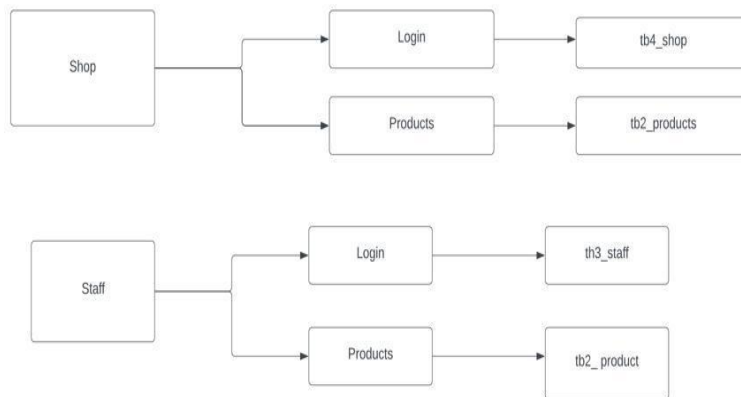


LEVEL 1



LEVEL 2**LEVEL 3**

LEVEL 4



5.4 DATABASE DESIGN

A database is a collection of inter related data. A database system provides the enterprise centralized control of its operational data. In this project the data are stored at specially designed in-memory database.

5.4.1 IN SHARE DATABASE

Tables in the database are implemented as flat files which can be accessed in random mode. Four base tables of the database are Tbl_user, Tbl_user_registration, Tbl_route, Tbl_feedback. Tbl_user stores the details of all the users in the database with a id number like name ,email_id, mobile number, gender etc.. and also name and email of admin. Tbl_route stores details of route where user going. On mounting the database, complete records of all the tables are read in to memory. Request for data is served from the memory copy and any updates are first written in the memory copy and subsequently flushed in to the physical file.

TABLE STRUCTURES:

Table 1:tbl_user

Field Name	Datatype	Description
Id	Int	User id
Name	Varchar(15)	User Name
Password	password	User password
Email	email	User Email
phone	text	User phone
User_type	Varchar(5)	User type
place	Varchar(15)	User place

Table 1:tb2_doctor

Field Name	Datatype	Description
Id	Int	Doctor id
Name	Varchar(15)	Doctor Name
Password	password	Doctor password
Email	email	Doctor Email
phone	text	Doctor phone
User_type	Varchar(5)	Doctor type
place	Varchar(15)	Doctor place

Table 1:tb3_staff

Field Name	Datatype	Description
Id	Int	Staff id
Name	Varchar(15)	Staff Name
Password	password	Staff password
Email	email	Staff Email
phone	text	Staff phone
User_type	Varchar(5)	Staff type
place	Varchar(15)	Staff place

Table 1:tb4_shop

Field Name	Datatype	Description
Id	Int	Shop id
Name	Varchar(15)	Shop Name
Password	password	Shop password
Email	email	Shop Email
phone	text	Shop phone
User_type	Varchar(5)	Shop type
place	Varchar(15)	Shop place

6. SYSTEM TESTING AND IMPLEMENTATION

6.1 SYSTEM TESTING

Software testing is defined as the process by which one detects the defects in the software. It is considered as the final opportunity for covert/ rectify and us to detect any defects that were in the software. Testing is a process which is done with the explicit intention of finding errors that make the program fail .In short, system and quality assurance is a review of the software products and related documentation for completion, correctness, reliability and maintainability. The first step in system testing is to prepare a plan that will test all aspects of the system.

System testing is an expensive, but critical process that may take as much as 50% of the budget for program development. Testing performs a very critical role for quality assurance and for ensuring the reliability of software. In a software development project, errors can be injected at any stage during development. Testing is the phase where the errors remaining from the earlier phases also must be detected. The common view of testing is that it is performed to prove that there are no errors in the program. But this is quite difficult since the analyst cannot prove that software is free from all sorts of errors. Therefore, the most useful and practical approach is with the understanding that testing is the process of executing program with the explicit intension of finding errors that is to make the program fail. A successful test can be therefore, one that finds an error.

The philosophy behind testing is finding errors. Test cases are devised with this purpose in mind. A test case is a set of data that the system will process as normal input. However, the data is created with the express intent of determining whether the system will process it correctly. Each test case is designed with the intent of finding errors in the way the system will process it. There are two general strategies for testing software, code testing and specification testing.

Code Testing

The code testing strategy examines the logic of the program. To follow this testing method, the analyst develops test cases that result in executing every instruction in the program or

module, that is, path through the program is tested. A path is specific combination of conditions that is handled by the program. On the surface, code testing seems to be an ideal method for testing software. However, even if code testing can be performed in its entirety, it does not guarantee against software failures.

This testing strategy does not indicate whether the code meets its specification nor does it determine whether all aspects are even implemented. Code testing also does not check the range of data that the program will accept, even though, when software failures occur in actual case, it is frequently because users submitted data outside of expected ranges.

Specification Testing

To perform specification testing, the analyst examines the specifications stating what the program should do, and how it should perform under various conditions. Then test cases are developed for each condition or combination of conditions and submitted for processing. By examining the results, the analyst can determine whether the program performs according to its specified requirements. Software testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before the live operation commences. It is the critical element of software quality assurance and ultimate review of specification, design and coding. The debugging process is the most unpredictable part of the testing procedure.

Syntax Testing

We use syntax testing to eliminate errors in the software. In the system, we have input fields like text, numeric fields. We should allow only numeric fields.

Unit Testing

Unit testing comprises the set of tests performed by an individual programmer prior to integration of the unit into a layer system. These are four categories of tests that can be performed on a program unit.

- Functional Unit
- Performance Unit
- Stress Unit
- Structure Unit

Integration Testing

34

Integration testing involves bottom up integration, top-down integration and sandwich integration strategy. Bottom-up integration starts with the traditional strategy used to integrate the components of software system into a whole functioning. Top-down integration starts with the main routine and one or two immediate subroutine in the system structure. Sandwich integration is predominantly top-down, bottom-up techniques are used in some modules and systems.

Acceptance Testing

Acceptance testing involves planning and execution of functional tests, performance tests and stress tests in order to demonstrate the implemented system satisfies its requirements. Functional test causes involve exercising the code with nominal files for which the expected outputs are known.

Thus the software system developed in the above manner is one that satisfies the user needs, confirms to its requirements and the design satisfactions and exhibits an absence of errors.

Validation Testing

Validation refers to the process of using the new software for the developed system live environment. i.e. new software inside the organization, in order to find out the error the validation phase reveals the failure and the bugs in the developed system. It will come to know about the practical difficulties the system focus. When operated in the true environment by testing the code of the implement software, the logic of the program can be examined.

A specifications test is conducted to check whether and specifications starting the program are performing under various conditions. Apart from these test are some special test conducted which are given below:

- **Peak load test:** This determines whether the new system will handle the volume of activities demand. The test has revealed that, the new software for the college is capable of handling demands at the peak time.
- **Storage testing:** This determines the capacity of the new system to store transaction data on a disk or on other files. The proposed software has the required storage space available, because of the use of a number of hard disks.
- **Performance time testing:** This test determines the length of the time used by the system to³⁵ process transaction data.

White Box Testing

White box testing is a test case design method that causes the control structure of the procedural design to derive test cases. Using white box testing methods, the software engineer can derive test cases that

- a. All independent paths within a module have been exercised at least once.
- b. Exercise all logical decisions on their true and false sides.
- c. Execute all loops at their boundaries and within their Operational bounds
- d. Exercise internal data structures to ensure their validity.

Black Box Testing

Black box testing methods focus on the functional requirement 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.

- a. Incorrect or missing functions
- b. Interface errors
- c. Errors in data structures or external database access.

- d. Performance errors and
- e. Initialization and termination errors

Black box testing is not an alternative to White box testing. Rather it is a complimentary approach that is likely to uncover a different class of errors than white box methods. Unlike white box testing, that is performed early in the testing process. Black box testing to be applied during later stages of testing.

6.1 SYSTEM IMPLEMENTATION

Implementation is a process of ensuring that the information system is operational. Implementation allows the users to take over its operation for use and evaluation. It involves training the users to handle the system and plan for a smooth conversion.

The personnel in the system must know in detail what their roles will be, how they can use the system, and what the system will or will not do. The success or failure of well designed and technically elegant systems can depend on the way they are operated and used.

Systems operators must be trained properly such that they can handle all possible operations, both routine and extraordinary. The operators should be trained in what common malfunctions may occur, how to recognize them, and what steps to take when they come.

Training involves creating troubleshooting lists to identify possible problems and remedies for them, as well as the names and telephone numbers of individuals to contact when unexpected or unusual problems arise.

Training also involves familiarization with run procedures, which involves working through the sequence of activities needed to use a new system.

*

6.2 SYSTEM MAINTENANCE

Software is always evolving and it is never finished as long as it is used; partly to accommodate for the ever changing world we live in. The evolution of your software might be motivated by a variety of reasons; to keep the software up and running, to upgrade to the latest release, enhance features or to rework the system for future maintainability. No matter the motivation, software change is vital for the evolution and success of it. Therefore, software will have to undergo changes, and understanding the different types of changes your software can go through is important to realize that software maintenance is more than just *bug fixing*. In fact, a study suggests that over 80% of software change is attributed to non bug related changes.

There are four categories of software maintenance:

- Corrective maintenance
- Adaptive maintenance
- Perfective maintenance
- Preventive maintenance

CORRECTIVE MAINTENANCE

Corrective change, most commonly referred to as “bugs,” is the most typical change associated with maintenance work. Corrective changes address errors and faults in your software that could affect various areas of your software; design, logic or code. Most commonly, these changes are sprung by bug reports created by users. It is important to note that sometimes problem reports submitted by users are actually enhancements of the system not bugs.

ADAPTIVE MAINTENANCE

Adaptive change is triggered by changes in the environment your software lives in. An adaptive change can be triggered by changes to the operating system, hardware, software dependencies and even organizational business rules and policies. These modifications to the environment can trigger changes within other parts of your software. For example, updating the server, compilers, etc or modifications to shipping carriers and payment processors can affect functionality in your software.

PERFECTIVE MAINTENANCE

Perfective changes refers to the evolution of requirements and features in your existing system. As your software gets exposed to users they will think of different ways to expand the system or suggest new features that they would like to see as part of the software, which in turn can become future enhancements to the system. Perfective changes also includes removing features from a system that are *not effective and functional to the end goal of the system*. Surprisingly, 50-55% of most maintenance work is attributed to perfective changes.

PREVENTIVE MAINTENANCE

Preventive changes refer to changes made to increase the understanding and maintainability of your software in the long run. Preventive changes are focused in decreasing the deterioration of your software in the long run. Restructuring, optimizing code and updating documentation are common preventive changes. Executing preventive changes reduces the amount of unpredictable effects a software can have in the long term and helps it become scalable, stable, understandable and maintainable.

7. FUTURE ENHANCEMENT

Future of Ayurveda

Ayurveda has been proving its worth since forever, and its positive results have been attracting a large number of audiences to adopt the same in their life. The ancient concept of Ayurveda holds its relevance to date and will continue to do so in the coming future. Increasing awareness about the efficacy of natural and traditional medicines, a surge in research and developmental activities, and less dependency on the allopathy healthcare system, the emergence of self-learned customers, and prevalence of chronic ailments could help in expanding the scope of Ayurveda across the world.

I think we all need to adapt this wholesome, healthy, and soothing approach for our betterment. The more natural the remedies, the lesser are the side effects, leading us towards a healthy and pandemic free future

8. CONCLUSION

Ayurvedic theories and practices on health, food, and nutrition are quite different from those of biomedicine and modern nutrition. Systematic exploration can provide new insights to health and nutritional sciences to provide contemporary solutions in healthcare, for instance, how one can modulate the diet and lifestyle to suit one's prakriti, age, and season. Rasayana in particular is an area worth exploring for new ways of rejuvenation and anti-aging. Healthcare costs are a major concern to the government exchequers of both developing and developed countries. Knowledge as to how to manage health at an individual level can help bring down sky rocketing healthcare costs through providing wellness. The Ayurvedic principles and practices can potentially become relevant for designing an integrated health care strategy. Concepts in Ayurveda, such as the rasa of a material being an indicator of its action on the body, are new to biomedicine and the modern nutritional sciences and can provide practical ways to create balanced diets.

9. BIBLIOGRAPHY

References

1. Rajib Mall, Fundamentals of software Engineering: 3rd Edition, PHI learning pvt, ltd.
2. Herbert Scheldt, Java The Complete Reference, 8 th Edition, Tata Mcgraw Hill Education Private Limited, ISBN: 9781259002465
3. Internetworking with TCP/IP: Douglas E Comer

URL

<http://softwaretestingfundamentals.com/system-testing/>

https://en.wikipedia.org/wiki/Software_engineering

<http://www.lucidchart>

10. APPENDIX

10.1 SAMPLE CODE:

Views.py

```
from django.shortcuts import render
from .models import *
from django.contrib.auth import logout
from django.http import HttpResponse, HttpResponseRedirect
from django.core.files.storage import FileSystemStorage
import datetime
from django.db.models import Sum
from django.http import JsonResponse
from django.core.mail import send_mail
from django.conf import settings
from django.template.loader import render_to_string
from django.views.decorators.cache import cache_control
from django.utils.datastructures import MultiValueDictKeyError

from datetime import date

def index(request):
    return render(request, 'index.html')

def aboutus(request):
    return render(request, 'about.html')

def contact(request):
    return render(request, 'contact.html')

def enquiry(request):
    if request.method=="POST":
```

```

Name=request.POST["Name"]

Email=request.POST["Email"]

MobileNumber=request.POST["MobileNumber"]

Message=request.POST["Message"]

a=Tbl_Enquiry(Name=Name,Email=Email,MobileNumber=MobileNumber,
              Message=Message)

a.save()

return render(request,'contact.html')

@cache_control(no_cache=True,must_revalidate=True,no_store=True)

def login(request):

    if request.method == "POST":

        email = request.POST["email"]

        password = request.POST["pass"]

        Superadmin=Tbl_User.objects.filter(User_Email=email,User_Password=password,User_Type="Admin")

        Doctor=Tbl_Doctor.objects.filter(Dctr_Email=email,Dctr_Password=password,Type="Doctor")

        Shops=Tbl_Shops.objects.filter(Shop_Email=email,Shop_Password=password,Shop_Status="Approved")

        Staff=Tbl_Staff.objects.filter(Staff_Email=email,Staff_Password=password,Staff_Type="Staff")

    if Superadmin:

        for x in Superadmin:

            request.session['id'] = x.id

            return render(request,'Admin/headerfooter.html',{'success':'Succesfully LogedIn'})

    elif Doctor:

        for x in Doctor:

            request.session['id'] = x.id

            return render(request,'doctor/headerfooter.html',{'success':'Succesfully LogedIn'})

    elif Shops:

```

```

    for x in Shops:
        request.session['id'] = x.id
    return render(request,'shops/headerfooter.html',{'success':'Succesfully Login'})
elif Staff:
    for x in Staff:
        request.session['id'] = x.id
    return render(request,'staff/headerfooter.html',{'success':'Succesfully Login'})
else:
    return render(request, 'login.html', {'error': 'Invalid login credentials.!'})
else:
    return render(request, 'login.html')

def logout(request):
    if request.session.has_key('id'):
        del request.session['id']
        logout(request)
    return HttpResponseRedirect('/')

def add_doctor(request):
    if request.session.has_key('id'):

        if request.method=="POST":
            Dctr_Name=request.POST["name"]
            Dctr_Email=request.POST["email"]
            Dctr_Password=request.POST["password"]
            Dctr_Phone=request.POST["phone"]
            Dctr_Place=request.POST["district"]
            Dctr_Department=request.POST["department"]
            Dctr_Photo=request.FILES["image"]
            var=Tbl_Doctor.objects.all().filter(Dctr_Email=Dctr_Email)
            if var:

```

```

return render(request, 'admin/add_doctor.html',{'error':'Email ID already existed'})

else:

    a=Tbl_Doctor(Type="Doctor",Dctr_Photo=Dctr_Photo,Dctr_Name=Dctr_Name,Dctr_Email=Dctr_Email,
    Dctr_Password=Dctr_Password,Dctr_Phone=Dctr_Phone,Dctr_Place=Dctr_Place,Dctr_Department=Dctr_
    Department)

a.save()

subject = 'AyurExpert'

message=render_to_string('shops/email_msg.html',
                        {'user':Dctr_Email,'psw':Dctr_Password,})

email_from = settings.EMAIL_HOST_USER

recipient_list = [Dctr_Email, ]

send_mail( subject, message, email_from, recipient_list )

return render(request, 'admin/add_doctor.html',{'success':'Succesfully Added'})


else:

    return render(request, 'admin/add_doctor.html')

    def view_doctor(request):

var=Tbl_Doctor.objects.all()

return render(request,'admin/view_doctor.html',{'db':var})


def view_users(request):

    var=Tbl_User.objects.all().filter(User_Type="User")

    return render(request,'admin/view_users.html',{'db':var})


def shop_reg(request):

if request.method=="POST":

Shop_Name=request.POST["name"]

Shop_Email=request.POST["email"]

Shop_Password=request.POST["password"]

Shop_Phone=request.POST["phone"]

```

```

Shop_Place=request.POST["district"]

Shop_License=request.POST["License"]

var=Tbl_Shops.objects.all().filter(Shop_Email=Shop_Email)

if var:

return render(request,'shop_register.html',{'error':'EmailId already existed'})

else:

    a=Tbl_Shops(Shop_License=Shop_License,Shop_Place=Shop_Place,Shop_Phone=Shop_Phone,Shop_Stat
    us="Pending",Shop_Name=Shop_Name,Shop_Email=Shop_Email,Shop_Password=Shop_Password)

a.save()

text="""<script>alert('SuccessfullyBooked');window.location=/shop_reg/;</script> """

return HttpResponse(text)

else:

return render(request,'shop_register.html')

def shop_list(request):

    return render(request,'admin/shop_reg.html')


def remove_doctor(request):

    if request.session.has_key('id'):

        ii=request.GET['id']

        var=Tbl_Doctor.objects.all().filter(id=ii).delete()

        return HttpResponseRedirect('/view_doctor/')


def view_shops(request):

    if request.session.has_key('id'):

        var=Tbl_Shops.objects.all().filter(Shop_Status="Pending")

        var1=Tbl_Shops.objects.all().filter(Shop_Status="Approved")

        var2=Tbl_Shops.objects.all().filter(Shop_Status="Rejected")

```

```

var_count=Tbl_Shops.objects.all().filter(Shop_Status="Pending").count()

var1_count=Tbl_Shops.objects.all().filter(Shop_Status="Approved").count()

var2_count=Tbl_Shops.objects.all().filter(Shop_Status="Rejected").count()

    return
    render(request,'admin/view_shops.html',{'db':var,'db1':var1,'db2':var2,'var_count':var_count,'var1_count':va
r1_count,'var2_count':var2_count})

def approve_shops(request):

    if request.session.has_key('id'):

ii=request.GET['id']

var=Tbl_Shops.objects.all().filter(id=ii).update(Shop_Status="Approved")

var1=Tbl_Shops.objects.all().filter(id=ii)

for x in var1:

email=x.Shop_Email

psw=x.Shop_Password

subject = 'Ayurvedha Expert'

message=f'Hi , thank you for registering.you got approval from admin.and your password is :{psw}'

email_from = settings.EMAIL_HOST_USER

recipient_list = [email, ]

print("mail sent")

send_mail( subject, message, email_from, recipient_list )

return HttpResponseRedirect('/view_shops/')

def reject_shops(request):

    if request.session.has_key('id'):

ii=request.GET['id']

var=Tbl_Shops.objects.all().filter(id=ii).update(Shop_Status="Rejected")

var1=Tbl_Shops.objects.all().filter(id=ii)

for x in var1:

email=x.Shop_Email

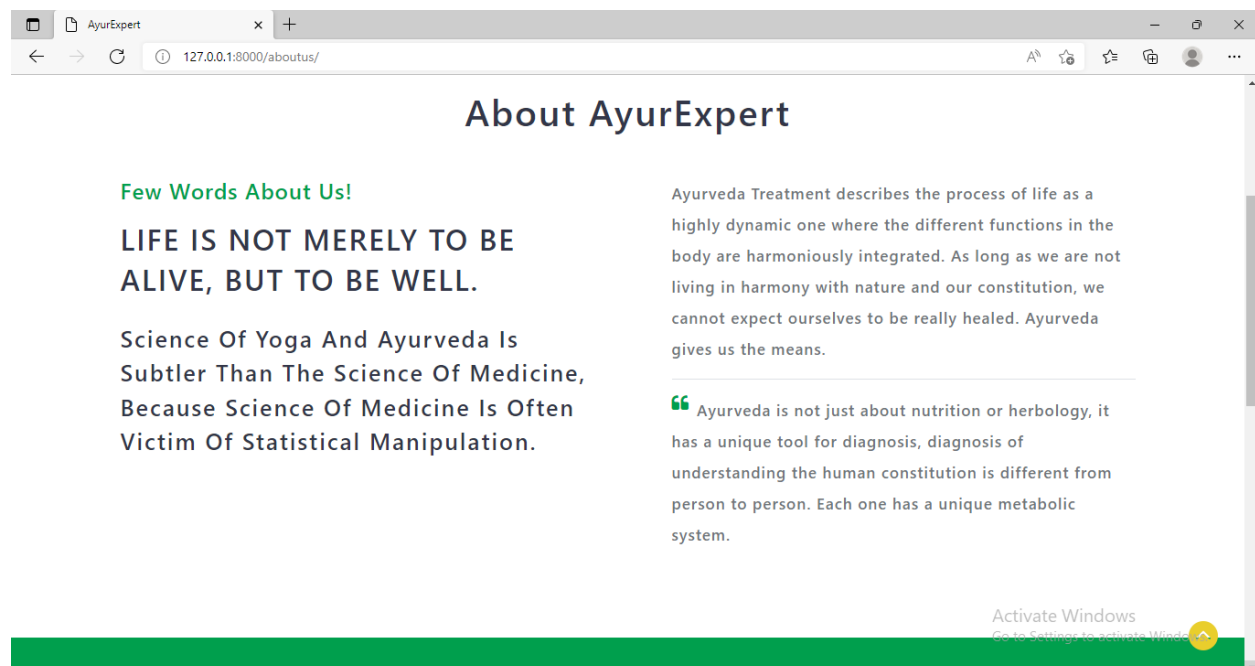
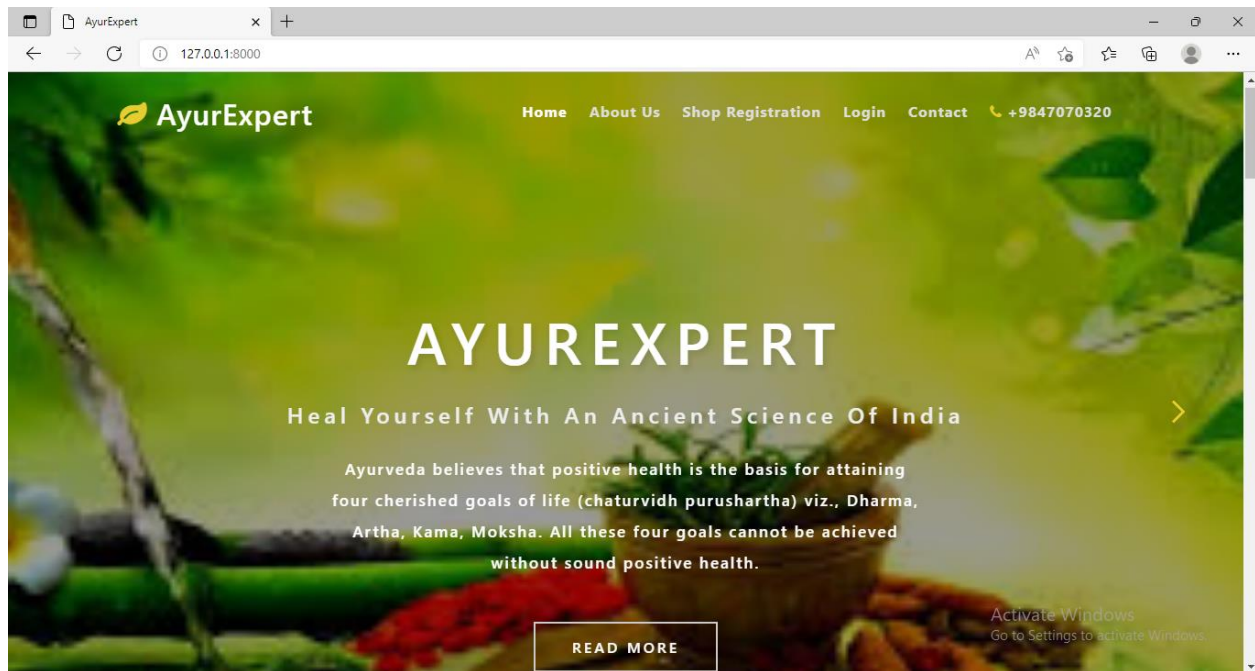
```

psw=x.Shop_Password

subject = 'Ayurvedha Expert'

message = f'Hi , you are rejected'

10.2 SCREEN SHOTS:



AyurExpert

Home About Us Shop Registration Login Contact +9847070320

SHOP REGISTRATION

Shop Name

Name

Mobile number

Phone

Email ID

Email

Password

Password

Activate Windows
Go to Settings to activate Windows

Name

Mobile number

Phone

Email ID

Email

Password

Password

District

select district

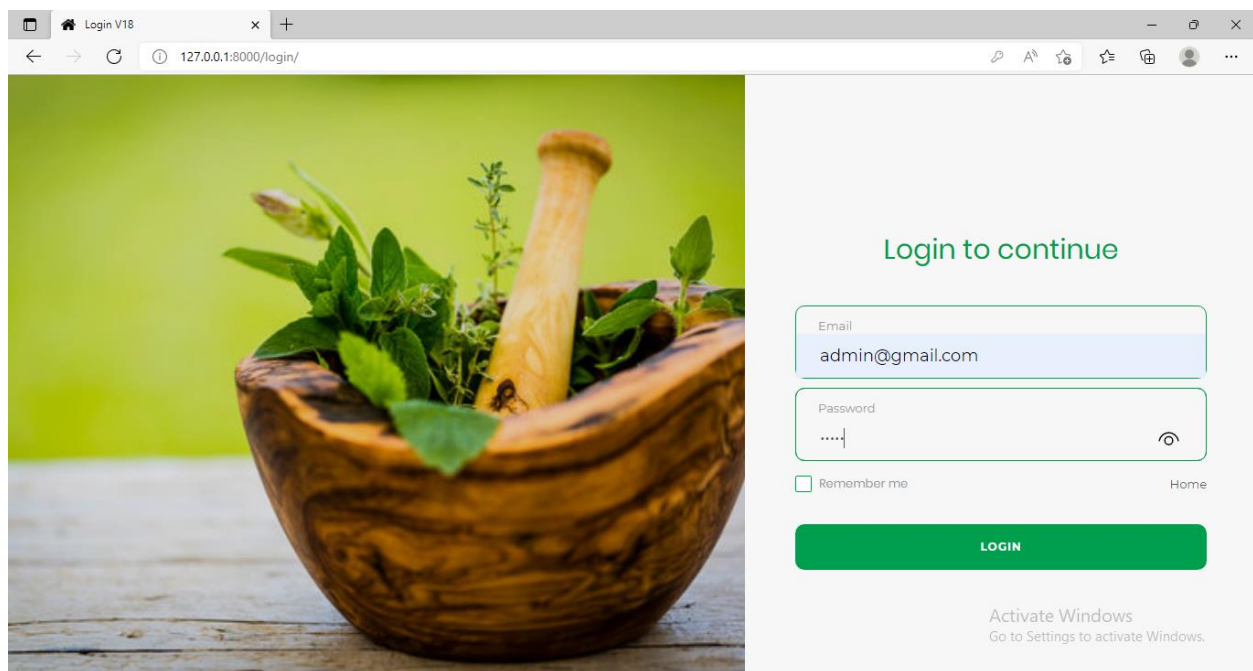
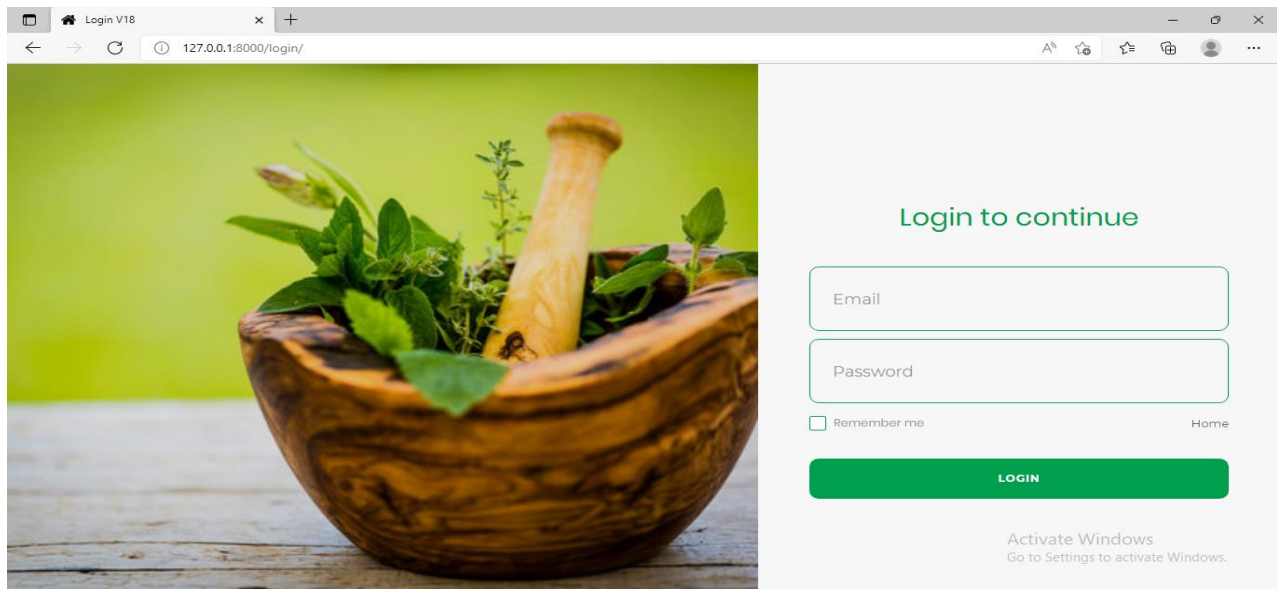
Shop License

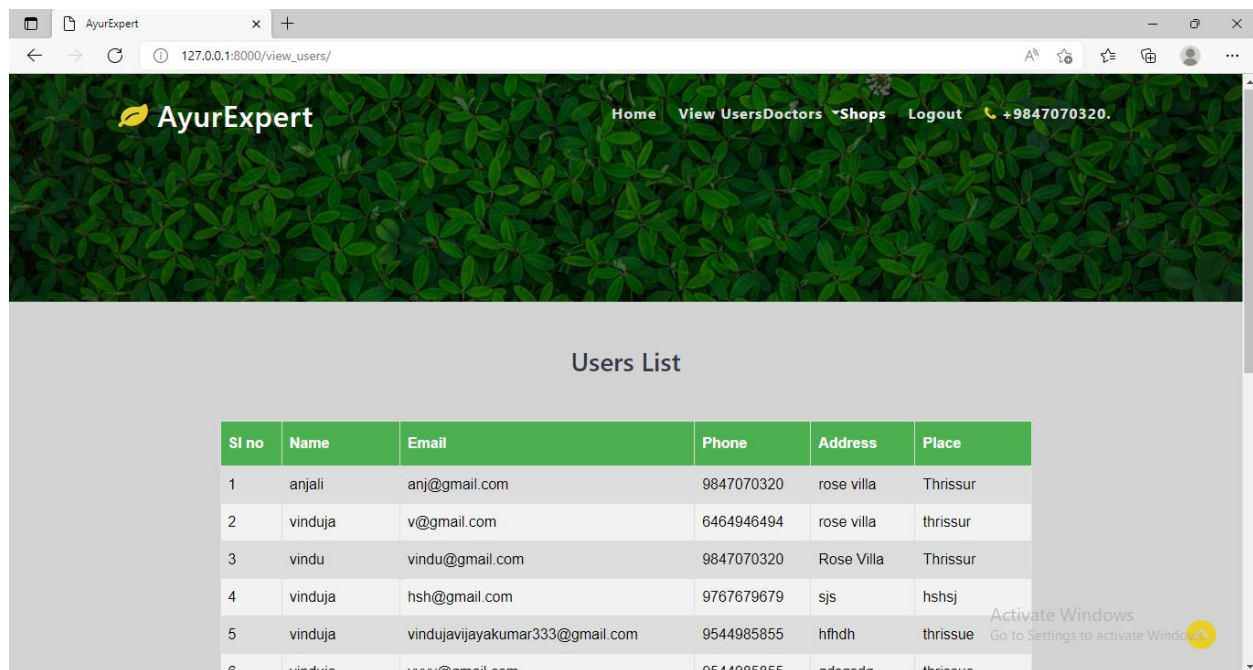
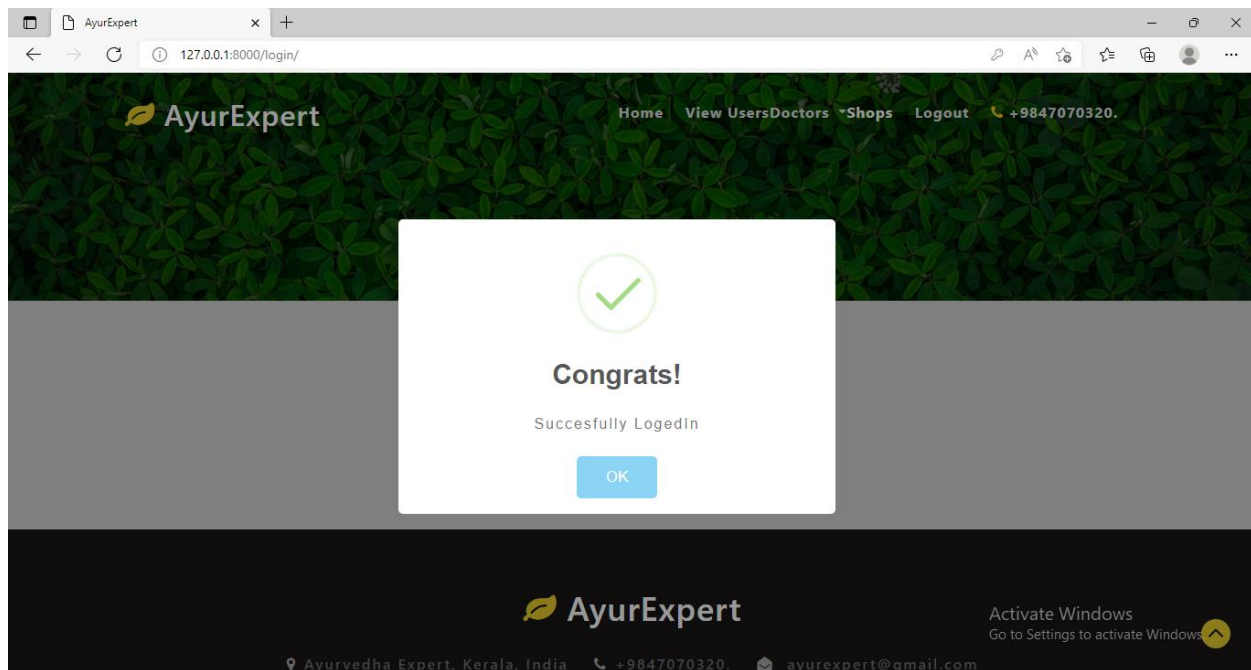
Enter License

ADD

AyurExpert

Activate Windows
Go to Settings to activate Windows





AyurExpert

Home View UsersDoctors Shops Logout +9847070320.

Add Doctors
View Doctors

ADD DOCTOR

Doctor Name

Name

Mobile number

Phone

Email ID

Email

Password

127.0.0.1:8000/add_doctor/#ssword

Activate Windows
Go to Settings to activate Windows

AyurExpert

127.0.0.1:8000/add_doctor/

Name

Mobile number

Phone

Email ID

Email

Password

Department

Enter Department

District

select district

Image

Choose File No file chosen

ADD

Activate Windows
Go to Settings to activate Windows

A screenshot of a web browser displaying the "Doctor List" page of the AyurExpert application. The browser address bar shows the URL "127.0.0.1:8000/view_doctor/". The page features a green leafy background at the top. The main content area is titled "Doctor List" and contains a table with the following data:

Sl no	Name	Email	Phone	Department	Place	Fees	Photo	Remove
1	aiswarya	aiswaryaas101@gmail.com	9878474887	ortho	idukki	250		Remove
2	vinduja	vinduja@gmail.com	9847070320	ortho	kannur	200		Remove
3	riya	riya@gmail.com	9847070320	ryrty	palakkad	200		Remove

An "Activate Windows" watermark is visible in the bottom right corner of the browser window.

A screenshot of a web browser displaying the "Contact Us" page of the AyurExpert application. The browser address bar shows the URL "127.0.0.1:8000/contact/". The page features a green leafy background at the top. The main content area is titled "Contact Us" and includes the following sections:

- Visit Us:** AyurExpert, Kerala, INDIA.
- Mail Us:** ayurexpert@gmail.com
- Call Us:** +9847070320

Below these sections are input fields for "Your Name", "Email", and "Message". An "Activate Windows" watermark is visible in the bottom right corner of the browser window.

Visit Us
AyurExpert, Kerala, INDIA.

Mail Us
ayurexpert@gmail.com

Call Us
+9847070320

Your Name

Email

Mobile Number

Message

SUBMIT

Thrissur
Kerala
View larger map

Directions

Map showing Thrissur, Kerala, and surrounding areas including Eranielur, Killannur, Peramangalam, Madakkathara, Patikkad, Kuthiran, Vadakkancherry, Vandanazhi, Karady, Palakuzhy, Kizhakkancherry, Mangalam Dam, and Beechi Vazhani Wildlife Sanctuary.