

Find Your Cure

**Project Report submitted in partial fulfillment of
the requirement for the degree of**

BACHELOR OF TECHNOLOGY

IN

Computer Science and Engineering

By

Manish Sharma (151254)

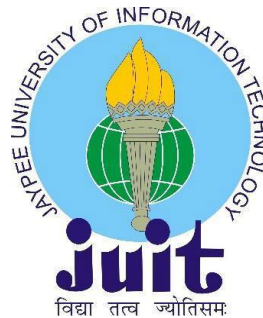
Rishabh Singh (151284)

Devesh Kashmiri (151295)

UNDER THE GUIDANCE OF

Dr. Hemraj Saini

To



**Department of Computer Science & Engineering and Information Technology Jaypee
University of Information Technology Waknaghat, Solan-173234 Himachal Pradesh**

Candidate's Declaration

I hereby declare that the work presented in report entitled Find Your Cure in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried over a period from 4th February 2019 to 17th May 2019 under Industrial Internship manager and mentors.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

**Devesh Kashmiri (151295)
Rishabh Singh (151284)
Manish Sharma(151254)**

ACKNOWLEDGEMENT

We are grateful to every member of Education and Training department, Mysore for their continuous guidance and supervision as well as the computer division for providing us with all the necessary software and hardware requirements and the information to complete our project.

We would also like to express our gratitude towards Our Mentor, for her continuous guidance and co-operation and helping us to tackle the complicated parts of our project.

We are also grateful to faculty of Computer Science Department of Jaypee University of Information Technology, Wagnaghat for their guidance over the last 4 years and nurturing us into professionals.

**Devesh Kashmiri (151295)
Rishabh Singh (151284)
Manish Sharma (151254)**

Abstract

FindYourCure is a venture guided towards healthcare discovery. It offers a common platform to patients, seeking the best healthcare options, and to doctors, who provide their services, based on their area of specialty. It will cater to the needs of patients by allowing them to register and create their own profile. Doctors, based on their area of specialization, should also be allowed to create and maintain a profile. Patients should be allowed to key-in details regarding their symptoms, gender, age etc., while doctors should be allowed to enter certain parameters such as visiting hours, fee structure, area of specialization, among others. The application will incorporate a data analysis algorithm that will help patients find the most suitable doctor, as per the details of the patient condition

TABLE OF CONTENT

1. Introduction

1.1 Introduction.....	1
1.2 Problem Statement.....	4
1.3 Objectives.....	4
1.4 Methodologies.....	4
1.4 Methodologies.....	14

2. Literature Survey

2.1 Introduction to Literature Survey.....	16
2.2 Summary of Papers.....	16

3. System Development

3.1 Technologies Used	18
3.2 Requirements	31
3.3 Database	34
3.4 Data Access Layer.....	38

4. Performance Analysis

4.1 Algorithm analysis and screenshots.....	39
---	----

5. Conclusion

5.1 Conclusions.....	54
5.2 Future Scope.....	54
5.3 Goals Achieved.....	55

CHAPTER-1

INTRODUCTION

1.1 Introduction

Health is essential for everyone. Health can't be bought by anything. Health is of utmost importance no matter what country we are from or the race we belong to, of what age we are or what our gender is. Some important details related to the patient like the past allergy, history pertaining to medications and medical history are often overlooked by the doctors when they diagnosing their patient's. The procedure for diagnosing also consumesthe most precious resource, time, even if the patient comes with minor problems that they can settle by themselves if they have the right help and knowledge.

The government often finds it difficult to have enough medical experts to get a good ratio of medical experts to patients. Furthermore, the doctors have totake into consideration a lot of issues before they can prescribe an accurate treatment for the patient.

Computer and automation basedapproaches are oftenused so as to improve the quality of the services provided by the medical sector so that the patient and the doctor are benefited equally. Generally speaking, in the not so developed areas, the citizenslive in conditions thatlack facilities of having experts to diagnose disease. So, to make health care a service that can be consumed by all, it is the need of the hour to provide the much needed expertise of medical specialists using the technology of the current era. Expert systems, that are rule based, includes both conventional techniques, such as database management systems (DBMSs), and artificial intelligence (AI) techniques, such as knowledge-based systems (KBSs) or expert systems (ESs). Medical diagnosis is a field of high activity and research as far as the introduction of these above techniques is concerned. In medical diagnosis, DBMSs are used for storing, retrieving and generally manipulating patient data, whereas ESs are mainly used to perform diagnoses similar to the way of a specialist based on patient data, since they can naturally represent the way experts reason.

Computer technology also has undergone significant transformations and these transformations have influenced many domains and revolutionized the way they function. The field that has been modernized and revolutionized by computer technology is that of medicine and healthcare. It has brought about significant transformations to these domains ranging from the organization of patient's data to the management related to even that of the operations theaters. Medical expert systems are also developed with many application systems so as to complement the duties performed by the doctor.

In today's modern society we have MBBS doctors, limitless medicine for limitless diseases. If we feel sick, we go to the doctor's clinic and doctor his diagnosis us, followed by medical prescription. Within a few days of taking the prescribed medicine we are all good to go. But if you fall ill at some remote place and you don't have any doctor nearby. What you do then? In Africa there is only 1 doctor among 5000 people. Presently, Kenya has slightly more than 5,000 doctors. There are many places where people are dying because of some common diseases. Just some common medicine can help them.

Even in areas where medical services are readily available, people often waste a good chunk of their valuable time by standing in queue. This act could get hazardous during emergency especially when it's a medical trauma. The patients got no idea what kind of disease they are facing which forces them to visit a doctor for mere consultation. This act can be gradually reduced, which could result in saving time as well as saving money for the Patience.

Similarly, doctors often have to deal with not just the disease that a patient is carrying but also their temper, thus acting as a boulder when it comes to healing the patient. Often the doctor has no records apart from the prescription that is provided by him. Thus, not keeping the doctor up-to-date with their patient.

The solution to all this is based on a simple question. What if you could visit a doctor, any time of day or night, without leaving your home and that doctor could review your history, diagnose your problem and also give you prescription? All this can be achieved by developing software applications to perform medical diagnosis and act as a bridge between the patient and specialist.

Machine learning is one of the most essential and key methodology that is used to perform analysis and subsequent predictions in the modern world. With the IT industry boom, and the rise of big data; it has become critical to analyze and predict trends. Slowly over the years, machine learning has forked out into almost every major industry, and is there performing functions that were almost unheard a mere few years ago. Machine learning is there to perform all kinds of predictions using Supervised, Semi-Supervised and Unsupervised learning algorithms. Supervised algorithms can be classified into two sub categories namely Regression Algorithms and Classification algorithms. Regression algorithms are used to predict values that are continuous in nature. For instance, Linear regression and Polynomial regression algorithms. The use of Classification algorithms is to predict outcome in terms of discrete values. For example, Decision Tree Classifications, Random Forrest Classifications, XGradient Boosting for classification. Python is the widely used language to develop and run machine learning algorithms. To run such algorithms, python open source community has provided varied number of modules and software applications with cross platform capability. Scikit Learn (sklearn in short) which is a module that is built on top of SciPy library in python version 2 onwards. Scikit Learn is a python library that solely focuses on data science and the various classifications, regression and clustering algorithms including SupportVector Machines, k-Nearest Neighbours, Random Forests, Logistic Regression, Gradient Boosting, Naive Bayes, k-Means, and Decision Tree, and is designed to operate within the python script for the Python numerical and scientific libraries NumPy and SciPy.

1.2 Problem Statement

FindYourCure is a venture guided towards healthcare discovery. It offers a common platform to patients, seeking the best healthcare options, and to doctors, who provide their services, based on their area of specialty. It will cater to the needs of patients by allowing them to register and create their own profile. Doctors, based on their area of specialization, should also be allowed to create and maintain a profile. Patients should be allowed to key-in details regarding their symptoms, gender, age etc., while doctors should be allowed to enter certain parameters such as visiting hours, fee structure, area of specialization, among others. The application will incorporate a data analysis algorithm that will help patients find the most suitable doctor, as per the details of the patient condition

1.3 Objectives

The objectives of the proposed expert system are:

To implement the concepts and logic of IT in real world problems.

To assist doctors for various diseases associated with symptoms i.e. to be a home assistant for doctors.

To provide hazard free experience to the patient.

To provide better treatment to the patient.

Choose the doctors based on their reviews

Save precious time by booking an appointment with the doctor and having contact over email available.

Remove the hassle of moving here and there depending on where the doctor refers your case. For every refer by the doctor, the patient is notified via email of the new appointment details.

1.4 Methodology

Since the project demands modification and flexibility as and when it progresses, the software development method that is used here is the Agile method.

Agile Methodology

Agile software development is a canopy term for a set of outlines and practices based on the standards and ideologies articulated in the Manifesto for Agile Software Development and the 12 Principles behind it. When you approach software development in a particular manner, it is good practice to adhere to these values and principles and use them to help find the proper methodology to be used regarding that particular context.

One distinctive thing about Agile when compared to other approaches of software development is the focus on the team members or developers and how they function as a single unit. Software solutions change through partnership between self-organizing cross-functional teams applying the fitting practices as per their context.

Agile software development community largely focuses on the cooperation and the self-organizing nature of the development team. That doesn't mean that there aren't managers. It translated to the fact that the team possesses the ability to discover how they're going to tackle things on their own. It simply means that these teams are cross-functional. The team doesn't have to have predefined roles tangled so much as that when you put the team together, you are critical of the right skill sets on the team.

This self-organizing nature doesn't mean that there still is no place for managers. Managers make sure team members have, or obtain, the right skill sets. The environment that thrives the team to success is all credit to the Managers. They mostly step back and let their team

discover how they are going to bring product idea to reality, but they step in when even after all the efforts they are unable to resolve issues.

When most development teams and establishments start doing Agile software development, they have their emphasis on the practices that help with cooperation and shaping the work. However, another key set of guidelines that are not so commonly followed but should be are detailed technical practices that directly come in contact with developing the software in a way that can help the team deal with ambiguity and uncertainty. Those technical guidelines are critical and shouldn't be overlooked.

A Short History of Agile

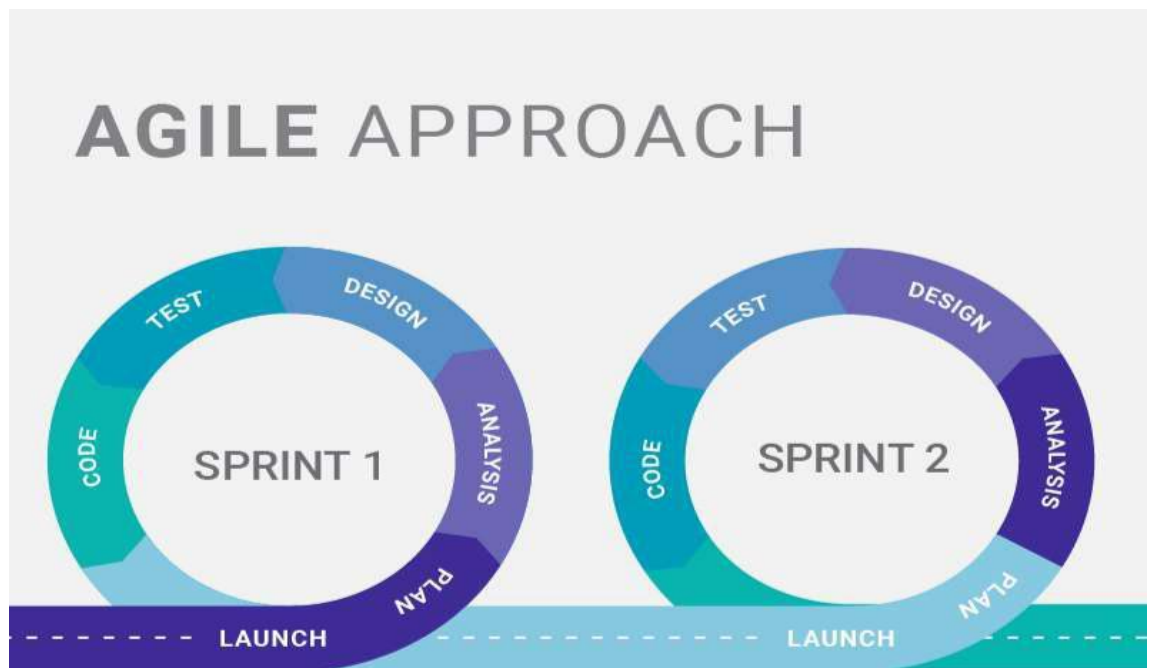


Figure 1. The Agile Methodology

There are many Agile procedures used across the world. The one used in this project is Scrum.

Scrum

Scrum is a framework within which developers can speak about intricate adaptive difficulties, while efficiently and imaginatively delivering products of the highest standard. Scrum in itself is a simple framework for effective team cooperation on intricate products. Scrum co-creators Ken Schwaber and Jeff Sutherland have written The Scrum Guide to explain Scrum clearly and succinctly. This Guide contains the definition of Scrum. This definition consists of Scrum's roles, events, artifacts, and the rules that bind them together.

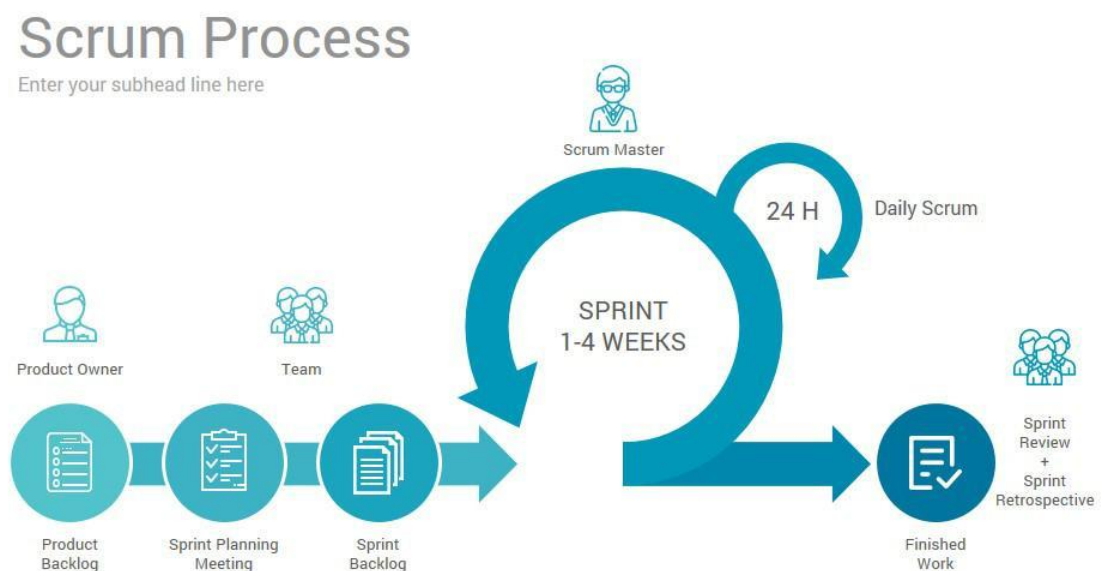


Figure 2. The Scrum Framework

Scrum Master

Cannot be deemed as a leading coordinator but can surely be deemed as one who aids and assists the Scrum process

Scrum master just makes sure that the work flow happens as expected in the sprint and thus making sure that the team does not deviate from the desired goal

The scrum master might not be the leader of the pack but he is the one behind all the time constraints and the work requirement calculation.

Scrum Meetings

The meetings are organized by the scrum master, who has absolutely no decision making authority.

Once the sprint has begun, a short meeting is always held between the owner of the product and the developers, where they decide the workflow for the entire duration of the sprint. The team extracts the work that they have to do from a pool of pending work known as the product backlog.

Once the meet has ended with product owner, the developers tend to segregate themselves in self-contained groups and target the work decided in the meeting for the current duration of the sprint.

There exists a time constraint on the planning of the work for the sprint. Generally speaking, a standard duration of 8 hours is allocated for a sprint as long as 30-days. This duration is inversely proportional to the time for which the sprint is decided to take place.

Daily Scrum and Sprint Execution

During the start of the day, before any work has begun, a short meeting is held to brief the members about the previous day's work, the work that has to been done on that particular date, and what all work could not be accomplished and the reason for the same. The main purpose is to make sure that the team is aware of the progress and the issues in development.

The guidelines dictate that the meeting occurs as the first thing to the start of the day and that no other work should begin till the point the work has been discussed.

Sprint Review Meeting

Once the sprint is over, it is the duty of the development team to make sure that they demonstrate the working product as in iteration to the product desired to all parties interested.

The meeting should not be a report based meet, but one involving a live demonstration. Once the demonstration is over it is the job of the project owner to provide feedback and tell the development team of the shortcomings simultaneously tallying the product backlog. This is done so as to make the team aware of all the vision the idea owner has regarding the project. All user stories that could not be completed are returned back to the pool of tasks and are prioritized according to the comments made by the owner.

The Sprint Review Meeting is the proper gathering for outer partners (even end clients) to visit. It is the chance to review and adjust the item as it develops, and iteratively refine everybody's comprehension of the necessities. New items, especially programming items, are difficult to picture in a vacuum. Numerous clients should almost certainly respond to a bit of working programming to find what they will really need. Iterative advancement, an esteem driven methodology, permits the formation of items that couldn't have been indicated in advance in an arrangement driven methodology.



Figure 3. Scrum Process

1.5 Organization

Chapter 1 highlights and emphasizes the basic explanation of an Expert System and why is it needed. It talks about the current scenario of the doctor-patient appointment, and how this project helps in overcoming the issues and shortcomings of this existing process. The problem statement is defined as creating an online portal that helps a patient to select a specialist for the cure of his/her disease based on the symptoms entered without having to face doctor referrals time and again. It then talks about the objectives that the project intends to achieve. Finally, it explains the methodology used in making this project which is the Agile Scrum methodology.

Chapter 2 consists of all the research papers that we went through to gain enough knowledge and information that we would have required before starting off with the project. It has a collection and review of all the literature collected from various journals, web portals, conferences, published papers, etc. It assisted us in ways as to how to go about our project and helped us learn about new and effective things that we could apply to make our project even more efficient.

Chapter 3 covers the development of the system, the algorithms and methodologies applied, the database and much more. It covers the entire project at length from beginning till end. It starts off from the requirements that we gathered which were supposed to be implemented. These include the functional and the non-functional requirements. Then, it presents the database which has all the tables and their entities used in the project. The entire flow of the project from registration to login to booking appointment, referring a patient, etc. All the screenshots of the website are provided there. It also explains in detail the technology used which is HTML, CSS for front-end, Python Django, Python

Django REST_Framework and SQLite for back-end. Basically, the entire project and its development is explained and shown in this chapter.

Chapter 4 presents the performance analysis of the project. It takes all the modules and puts them into test. Each and every functionality of the modules are tested. A number of testing techniques are used. It checks for the correct functionality of the module. It should work when correct data is entered and display errors when incorrect data is entered. The entire project is tested, its performance is analyzed and outputs are shown at various stages.

Chapter 5 concludes the entire project. It gives the summary of everything done in the project. It revises the aim and objectives of the project and whether the project was able to achieve that. A quick briefing of every chapter is done in this. It also talks about the future scope and all the extensive functionalities that can be applied later in the future to extend the project and make it even more effective. It also has the application's contributions, new inventions, innovative work and new ideas generated from this project.

CHAPTER-2

LITERATURE SURVEY

2.1 Introduction to Literature Survey

The project involved the prediction of diseases from the symptoms and henceforth the development required research the field of analytics and Data Mining. Following is a brief on the content and sources that were referred to as standard literature for the implementation and completion of the project.

2.2 Summary of Papers

2.2.1

Title: WEB BASED ONLINE MEDICAL DIAGNOSIS SYSTEM (WOMEDS)

Authors: Raenu Kolandaisamy, Rafidah Md Noor

Publishing: International conference on computer applications

Summary: The system proposed by the author was to perform health problem diagnostic for the patient and patient medical history management on the doctor's side. Not only this the system also proposed mechanisms to provide medical history records to the hospital administration. The users were tracked using simple username and password configuration. The doctor was provided due access to the history of medical illness and medications that the patient has mentioned in the database. The system was developed as the current doctor-patient system wasted the greatest resource 'time'. The patients had to spend the day in its entirety for an appointment only to be referred to another doctor. The project was segmented into three modules namely the registration module, administration module and diagnostic module. Future scope of the project was termed to be involved in the laboratory work.

2.2.2

Title: Disease Diagnosis System

Authors: Dipanwita Biswas, Sagar Bairagi , Neelam Panse & Nirmala Shind

Publishing:

Summary: The system proposed aimed to produce data of relevance for consultation purposes. To implement the same modules were created to manage and track the data flow and provide expert system diagnosis. The system also implemented a clinical exam correlation on the diagnosis that the expert system made. The symptoms were all associated with a weight and thus were used as input to the expert system. The system also proposed methodologies to prioritize the disease and provide with diagnosis

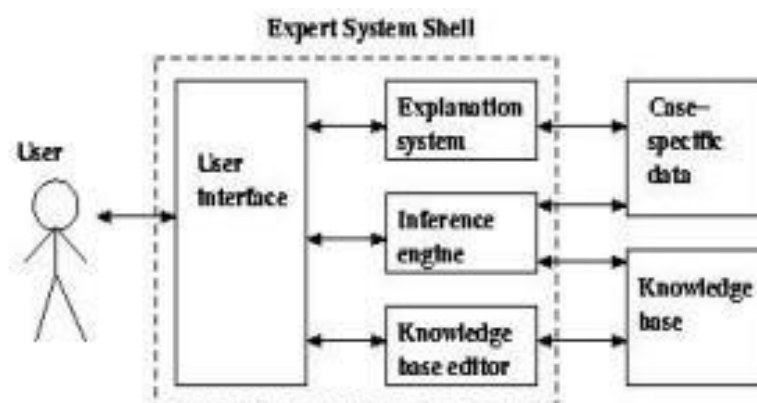


Figure 2: Expert System Architecture.

CHAPTER – 3 SYSTEM DEVELOPMENT

The entire project was done using the Agile Methodology. Agile requires dividing the project into sprints. This project was done by dividing it into two sprints of 14 days each.

The web based application is developed using HTML, CSS and JavaScript for the front end development with Python Django Template engine and Python Django along with Python Django-Rest-Framework for backend development. The core backend language used is Python. The tool used for programming is Visual Studio Code and for the database, SQLite3 is used.

3.1 Technologies Used:

Python

Python is an interpreted and simultaneously compiled, object-oriented, high-level programming language with dynamic semantics that allow ease of variable declarations and subsequent use. The fact that it contains high-level data structures built-in, along with the feature of dynamic typing and dynamic binding, make python an eye-catching language for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python is simple, very primitive to learn. Greatly focuses on code readability and therefore the cost associated with the maintenance of the program. Python supports modules and packages, which promotes code reusability and modularity. The language is entirely open sourced and thus Python interpreter and the extensive standard library are available in source or binary form are available for all major platform at apparently no cost at all, and can be freely distributed. This open source license allows the use of python and python interpreter to be used in all domains, be it for personal use or for commercial use in the industry

MTV Web Architecture

The server and client side component communicating logic is determined by the Web-Architecture, the software logic that determines the server and client component structure.

The project is developed following a Model-Template-View architecture:

The MTV architecture splits the application into the following subcomponents:

Model: The Model the representation of the database structure and tables in the form of classes and column in the form attributes of the class. Each object of the model class is representative of each row the database tables.

View: The request handler function or method is known as a view, which receives HTTP requests and returns HTTP responses either in form or data or web pages that are to be rendered by the browser. To satisfy the requests sent to the view, it accesses the data through the models, and send the response to the templates.

Models: The Model is a logical data structure represented by a database. Models are Python objects representative of each row the database tables, and possess the capability to manage (add, modify, delete) and query data into the database.

Templates: A template is a text file defining the structure or layout of a file (such as an HTML page), with placeholders used to represent actual content. A view can dynamically render an HTML page using an HTML template, showcase data in it with content from the model. It is not necessary for a template to be an HTML page and can be of any language that supports browser rendering.

Django Web Framework

Django is a web framework, written in Python, and is thus open source and free to use. Django follows the model–template–view (MTV) architectural pattern. It is maintained by the Django Software Foundation (DSF), an established as independent organization.

Django was developed with the idea the framework to be developed should simplify the process of development with code reusability and the idea of do not repeat yourself. Python is the core language used throughout, be it for settings, files, and data models. Django also provides the administrative and user database tables to use for the application with minimum effort and thus upholds to the fact that it is web framework for perfectionists with a deadline.

RESTful API

REST stands for REpresentational State Transfer and is an architecture style for designing loosely coupled applications over HTTP, that is often used in the development of web services. Generally speaking, it is the technical description of how the World Wide Web works. A REST Application Programming Interface (REST API) is a kind of web server that empowers a client worked or robotized customers, to get to assets that model a framework's information and capacities. REST architectural pattern does not impose implementation rules at lower level, it just enforces the high level guidelines and leaves the implementation to the hands of the developer.

Uniform interface

Uniform interface is fundamental to the design of the REST service. The constraints imposed define the server client interface. The four supervisory ideologies are:

Resource-Based: Singular assets are considered in obtaining and utilizing URIs as asset identifiers and are detached from responses for the client.

Actions on Resources through Representations: When a client receives resource representation, inclusive of the metadata involved, it possesses significant information to modify or delete the resource on the server, considering the fact that it has been granted the permission to do so.

Self-descriptive Messages: An individual message has a short and accurate help text that depicts the ways the use and process it. Their cache ability is also significant in their responses.

HATEOAS: The state is delivered from the client side via body contents, query-string parameters, request headers and the requested URI. The REST Services ends the state to clients via body content, codes in responses, and similarly through headers.

Client-server

There has to be distinction between the clients and the server and that is provided by the uniform interface. This translates to the fact that the client is not concerned with ideas and methodologies behind the storage of data and it thus remains internal to each server, so as to improve the compactness and modularity of the client code. Servers are not involved with anything pertaining to user interface or user state so as to make them more scalable and portable. This entire process makes the servers and clients largely independent and can be used interchangeably, so long as there aren't any modifications in the interface. Primarily there has to be an interface intact, one may provide the clients an online REST API documentation or have a dummy representation of the REST APIs to help the start the process of connecting from the clients and the make it accessible.

Stateless

The URI contains the essential state that is requires to operate the request. The same can also be contained in query-string parameters, body, or headers. The identification of the resource is done through the URI, and the body is composed of the state. Therefore, after

the processing is done by the server, the suitable state, or the fragments of it are communicated back to the client via headers, status and response body.

Cacheable

As the clients have the capability to cache responses, they necessarily have to be implicitly or explicitly defined as cacheable or not so as to prevent clients from referring to the same state or incorrect data in response to further requests. A caching that is well-managed partially or completely improves the overall performance and renders the client–server interactions to a minimum.

Layered system

Micro services or Layered system is where the client cannot normally tell about the connectivity of the client in terms of a direct or indirect connection to the server. Intermediate servers or API Gateways improve the scalability of the system by promoting the concept of load balancing providing caches that are shared. This layered system architecture also promotes the security policies.

Code on demand

Servers have the ability extend and alter the functionality provided to the client by sending a logic to it so that it is capable to execute it. Some examples of code on demand includes Java applets and client-side scripts such as JavaScript. This one constraint of the six is an optional constraint.

REST API building blocks

The Most important fundamental building blocks of an API:

Resources (URIs): Resources should describe the objects that are descriptive of the API, along an identifier that distinctively identifies each object.

HTTP methods: HTTP methods are used to change the objects existing state, with a list of methods that are authorized and also supported.

HTTP headers: HTTP headers are used for transferring the required and mandatory argument that can be for authentication, accepted content types, etc.

Query parameters: Any optional arguments are sent via Query parameters and can also be used for filtering (searching). They are not mandatory to initialize and can be omitted as required.

Return codes should mirror the meaning and semantics of the core HTTP specification so as to ensure that the code is informative and blown up with extra facts and figures in the load wherever required.

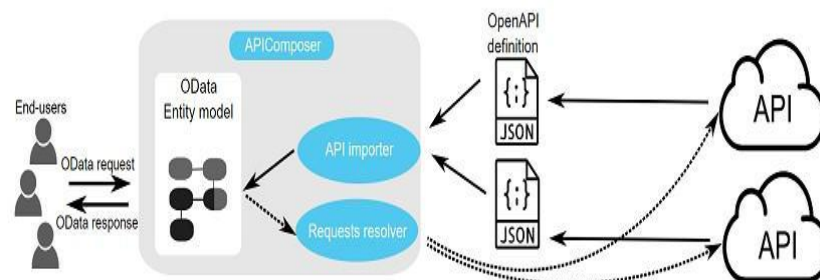


Figure 3. Scrum Process

Django REST-Framework

REST stands for REpresentational State Transfer and is an architecture style for designing loosely coupled applications over HTTP, that is often used in the development of web services. Generally speaking, it is the technical description of how the World Wide Web works. A REST Application Programming Interface (REST API) is a kind of web server

that empowers a client worked or robotized customers, to get to assets that model a framework's information and capacities. REST architectural pattern does not impose implementation rules at lower level, it just enforces the high level guidelines and leaves the implementation to the hands of the developer.

Some of the HTTP methods that are commonly used in REST architecture are:

GET– It is a simple read operation from the database and just return the data back to the requesting source. So GET is used to perform a read operation on any of the resource available.

PUT- Put is a simple update operation when talking about the database operations. This operation just changes the state of the resource, which can be an object, file or block.

POST– Post translates to a simple add operation and is primarily used to create that resource.

DELETE- Just like the delete operation in database, Delete method is used to clear the API of any resource.

Django REST Framework is an application used for rapidly building RESTful APIs based on Django models.

Now any web service that use a REST architecture, they are called Restful APIs or REST APIs. Commonly heard of REST APIs are Facebook API, Google API, Twitters AP. Making a restful API requires any device access to website data through a common format that, in most cases is, JSON (JavaScript Object Notation as it a data format that is significantly device independent. We use JSON to communicate within a specific format, so that we can build an android app,iOS app or any other software. Also, one key feature that comes into picture is that, if you have an API that performs a certain operation, then there is no need to rewrite that as part of main source code. Thereby, it reduces the size of the code and modularizes the components.

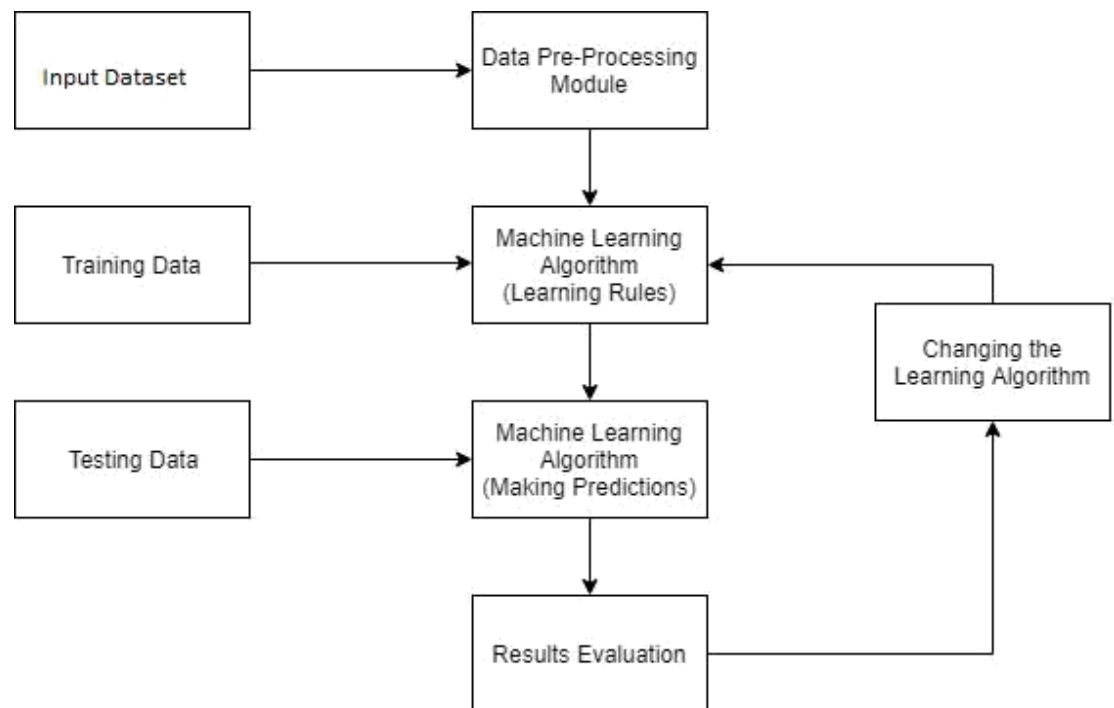
Machine Learning Models

To automate the process and find the disease and subsequently the specialist doctor, machine learning models are a must. In order to generate ML Model, we need sample data with target attribute given and a machine learning algorithm to train the model according to the nature of the target variables. Training a machine learning models follows a standard set of procedures as stated below.

Input the training dataset and run the algorithm on the input data so that the model can understand the relations between the input and the target variables. This is termed as the process of initial learning.

Tune the parameters to control the learning of the algorithm so that we can get a desired and a realistic accuracy from the model. This step is critical in deciding the future performance of the model. If the tuning process is not accurately, the model learning could lead to overfitting or under fitting of the input data and consequently result in not so realistic predictions.

After the algorithm finishes learning, the model is finally built.



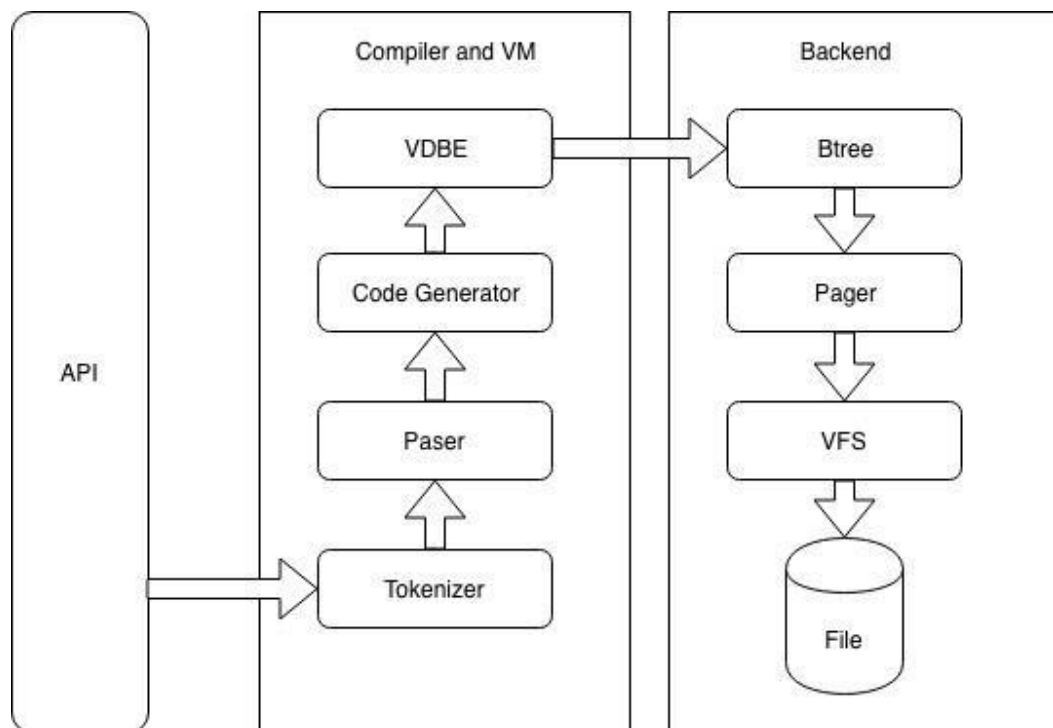
Now, when a new dataset comes in for prediction, it is passed to the model. The model that is built by learning the past sample data, thus predicts the output.

SQLite Database

SQLite is an in-process library that implements a self-contained, server less, zero-configuration, transactional SQL database engine and provides relational database management system that is light weight and significantly easy to use. The term lite in the name describes the lightweight nature the process has in terms of setup, database administration and required resources. The code for SQLite is open source and is in the public domain so that it is free for use for any purpose, be it commercial or private.SQLite is an embedded SQL database engine. Contrasting to other SQL databases, SQLite does not require a server to run and store the information. The database is implemented in the form of a file system and performs reads and writes directly to ordinary disk files that store the entire state of the database. A complete SQL database with all the features, is contained in a

single disk file. The database file format is cross-platform and can thus be used in either of the 32-bit or 64-bit systems.

The architecture of the SQLite database is segmented into two different units that are referred to as core and backend. The core unit contains Interface, Tokenizer, Parser, Code generator, and the virtual machine, which creates an order of execution that primarily affects the database transactions. Backend accesses the file system and is contained of B-tree, Pager and OS interface. Tokenizer, Parser and code generator are collectively named as the compiler and is responsible for generating a set of opcodes that are executed on a Virtual Machine.



The database and application integration approach used is code first approach.

Code first approach

In the Code First approach involves the manual creation of the entity domain model (EDM) classes so as to make the application communicate with the database. The database is then migrated from these classes that are created in the language being used. This provides direct and easy control on the database and any modifications can be done easily. The Domain-Driven Design (DDD) principles are the core guidelines that the code first follows.

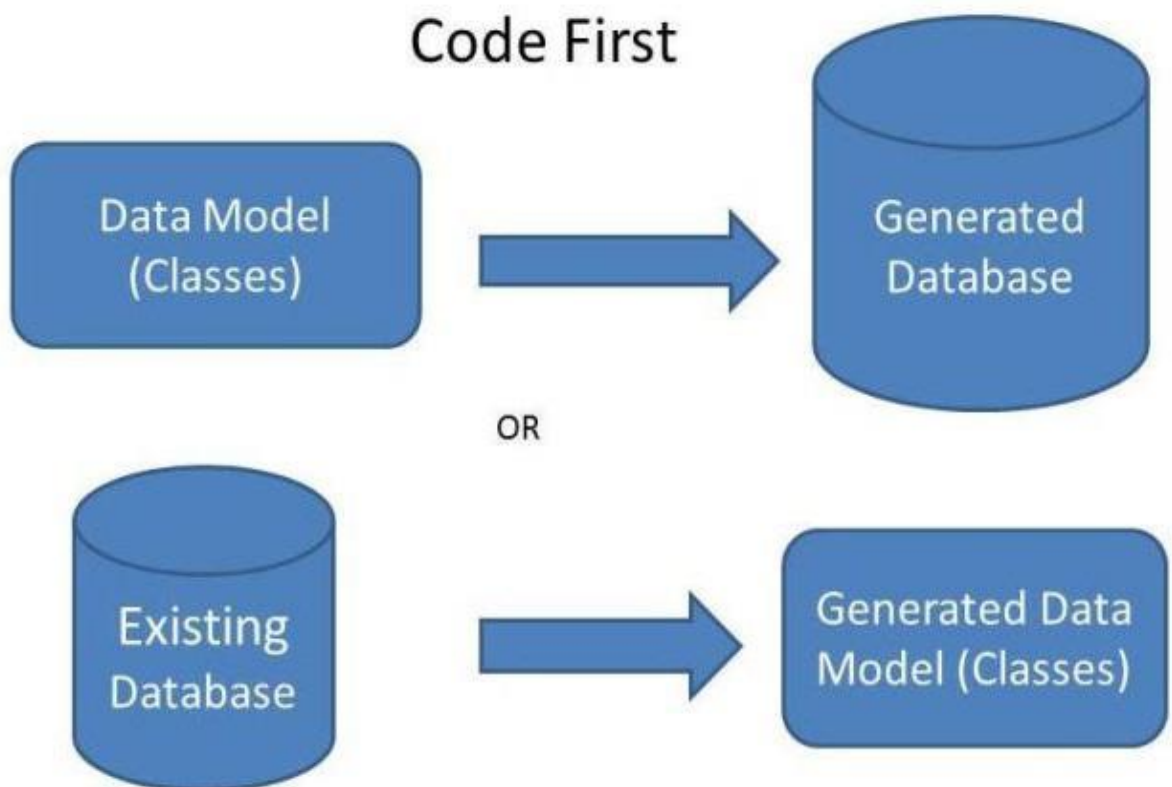
Benefits of Code First

The developer has full control over the code and the database. Unlike the database first approach, this approach does not provide the developer with any auto generated code. If any modifications are done in the auto generated codes, it might break your code and result in the loss of data.

The person responsible for the design of the database schema is not the database administrator but the developer himself. He is responsible for all the constraints, checks and associations in the database. Any modifications being done in the database schema just requires a new class or attribute followed by the process of migration to the database. This makes modifications in the database very easy as the database might require significant amount of changes as the development process progresses.

Code first migrations automatically generate a date-time stamped migration script and consequently apply these migrations to the database. Any migration made are auto generated to a migration script. Roll down capability is also provided in the code first approach as the migrations are always available with the developers.

Code initially can be utilized in both existing or new databases. If there should be an occurrence of a current database, you may need to physically code model classes to coordinate the database. The database first methodology will make all the area classes for you.



3.2 Requirements

Hardware Requirements

Processor/RAM	: Any multicore processor/4 or 8GB RAM
Web server	: Any suitable servers can be used
Database Server	: Not required

Software Requirements

Client End	: Web Browser(any), Operating System(any)
Development End	: Django, Anaconda, Visual Studio Code, Operating System(any)
Servers	: Localhost server
Services	: Django REST_framework services

Functional Requirements

As a Guest User I should be able to get information about the application.

As a Guest User I should be able to Register as Doctor or Patient.

**As a patient I should be able to reset my
password. As a Patient I should be able to Login.**

As a Patient I should be able to view my dashboard after login.

**As a Patient I should be able to Update my profile from the
dashboard. As a Doctor I should be able to Login.**

As a Doctor I should be able to view my dashboard after login.

**As a Patient on my dashboard I should be given my current appointment
details. As a Doctor I should be able to reset my password.**

**As a Doctor on my dashboard I should be given my appointments for
the day. As a Patient I should be able to provide my medical history.**

As a Doctor I should be able to Update my profile from the dashboard.

As a Doctor I should be able to view patient's medical history.

As a Patient I should be able to provide my symptoms.

As a Patient I should be able to find the doctors list on the basis of symptoms.

As a Patient I should be able to choose doctor and confirm appointment.

As a Patient I should be able to view doctor's details.

As a patient I can give feedback to the doctor.

As a doctor I should able to close the appointment after action.

As a Doctor I should be able to suggest any other doctor to patient. As a Patient I should be able to cancel my appointment.

As a Doctor I can register a patient by providing all mandatory details. As a Doctor I should be able to reschedule my appointment.

As a Doctor I should be able to view daily/monthly appointments.

As a Doctor I should be able to view previous appointments with the particular patient with whom he has current appointment.

As a Doctor I should be able to create new appointment with my patient from past and present appointments

As a Doctor I can send notifications of prescription

Reschedule appointment notification to patient

As a patient I should able to get my doctor's location.

As a Doctor I can provide a prescription to patient

As a Patient I can view about diseases

As a Doctor I can search about previous appointments based on date

EXTRA REQUIREMENTS

Email notifications.

Disease search.

Email prescription.

USER INTERFACE

Should be user friendly.

Should be responsive to screen sizes.

A User Story is the prerequisite or requirements of the user that have to be fulfilled.

Following are the User Stories or the tasks that have to be accomplished in the project:

US-01: As a Guest User I should be able to get information about the application.

US-02: As a Guest User I should be able to Register as Doctor or Patient.

US-03: As a patient I should be able to reset my password.

US-04: As a Patient I should be able to Login.

US-05: As a Patient I should be able to view my dashboard after login.

US-06: As a Patient I should be able to Update my profile from the dashboard.

US-07: As a Doctor I should be able to Login.

US-08: As a Doctor I should be able to view my dashboard after login.

US-09: As a Patient on my dashboard I should be given my current appointment details.

US-10: As a Doctor I should be able to reset my password.

US-11: As a Doctor on my dashboard I should be given my appointments for the day.

US-12: As a Patient I should be able to provide my medical history.

US-13: As a Doctor I should be able to Update my profile from the dashboard.

US-14: As a Doctor I should be able to view patient's medical history.

US-15: As a Patient I should be able to provide my symptoms.

US-16: As a Patient I should be able to find the doctors list on the basis of symptoms.

US-17: As a Patient I should be able to choose doctor and confirm appointment.

US-18: As a Patient I should be able to view doctor's details.

US-19: As a patient I can give feedback to the doctor.

US-20: As a doctor I should able to close the appointment after action.

US-21: As a Doctor I should be able to suggest any other doctor to patient.

US-22: As a Patient I should be able to cancel my appointment.

US-23: As a Doctor I can register a patient by providing all mandatory details.

US-24: As a Doctor I should be able to reschedule my appointment.

US-25: As a Doctor I should be able to view daily/monthly appointments.

US-26: As a Doctor I should be able to view previous appointments with the particular patient with whom he has current appointment.

US-27: As a Doctor I should be able to create new appointment with my patient from past and present appointments

US-28: As a Doctor I can send notifications of prescription

US-29: Reschedule appointment notification to patient

US-30: As a patient I should be able to get my doctor's location.

US-31: As a Doctor I can provide a prescription to patient

US-32: As a Patient I can view about diseases

US-33: As a Doctor I can search about previous appointments based on date

3.3 Database:

All the tables used are listed below:

Users -

First_name-First Name of the user. (varchar)

Last_name-Last Name of the user. (varchar)

Email-Primary Key, emailid of the registered user. (varchar)]

gender – Gender of the user. (char(1))

Age- Age of the user. (int)

Password-password of the user(password). (varchar)

Role-role of the user(char)

Doctor

Doctor_id-Primary Key,doctor identification number of the doctor.(int)

Email_id-Foreign key,emailid of the registered user.(varchar)

Fee_structure- Consultation fee of the doctor.(int)

Visiting_start- Time from which the doctor is available(datetime)

Visiting_end-Time till which the doctor is available(datetime)

Years_of_ex-Number of years of experience a doctor as (int)

Street-Street name of the Doctor's clinic/hostpital(vvarchar)

City-City of the doctor's clinic.(varchar)

Postal_code-Postal code of the doctor's clinic(vvarchar).

DoctorToSpecialization

Doctor_id-Foreign Key,doctor identification number of the doctor.(int)

Specialization_code-3 to 4 alphabet code depending on the doctor specialization.

Feedback

Feedback_id-Primary Key,Feedback id of the given feedback.(int)

Doctor_id-Foreign Key,doctor identification number of the doctor.(int)

Rating- Rating given to the doctor by the user.(int)

Comments- Comments given to the doctor (varchar)

Appointments

Appointment_id-Primary Key,Appointment id of the booked appointment.(int)

Doctor_id-Foreign Key,doctor identification number of the doctor.(int)

Patient_id-Foreign Key, identification number of the patient.(int)

Appointment_date- Appointment date of the patient (date)

Symptoms-Symptoms of the Patient(varchar)

Referred- Yes or No flag(Boolean)

PatientData

History of illness- Shows the history of illness of the patient

(varchar) History_of_medication-Shows the history of medication

of the doctor(varchar) Patient

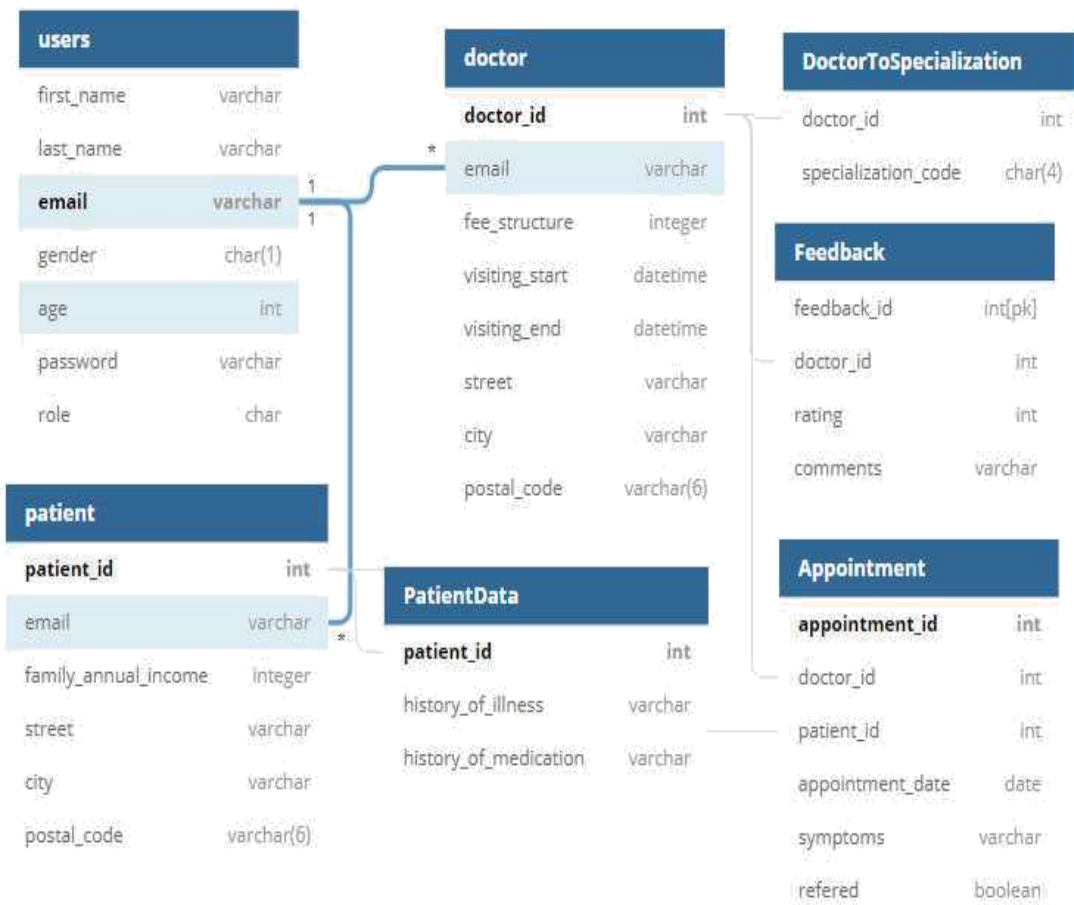
Patient_id-Primary Key, identification number of the patient.(int)

Family_anual_income-Shows the annual income of the family(varchar)

Street-Street name of the patient(varchar)

City-City of patient.(varchar)

Postal_code-Postal code of the patient's residence(varchar).



3.4 Data Access Layer

This is an abstraction layer between the database and the process requesting data or information from the database. All the database operations are directly done by the data access layer as this is the only piece of code that communicates with the actual database.

The functions implemented in Data Access Layer are:

Add_user()	add_appointment()
Find_user()	get_all_patient_appointments()
Get_user_info()	get_all_doctor_appointments()
User_data()	get_all_referable_doctors()
Add_patient()	send_email_confirmation()
Add_doctor()	add_feedback()
Get_patient_id()	get_total_rating()
Profile_completion()	get_upcoming_doctor_appointments(
Medical_history_completion())
Update_patient_data()	get_previous_doctor_appointment()
update_doctor_data()	update_appointment()
add_specialization()	get_referred_appointments()
update_medical_data()	update_password()
get_medical_records()	get_medical_records()
list_doctors_specialization()	get_upcoming_patient_appointments
generate_appointment_id()	()
find_available_doctors()	get_previous_patient_appointments()

CHAPTER – 4 PERFORMANCE ANALYSIS

4.1 Algorithms Analysis and Screenshots

The algorithm that performed the best in our case was the random forest machine learning algorithm. The ensemble learning algorithm showed acceptable overfitting and presented an accuracy of 85.678%. The model was changed from decision tree because of the overfitting the model presented as the data set size was very small. The input features were 132 symptoms mapped to a total of 41 diseases. The dataframe was a kind of sparse matrix with each row mapped to an output disease.

Following are the screenshots of the application.

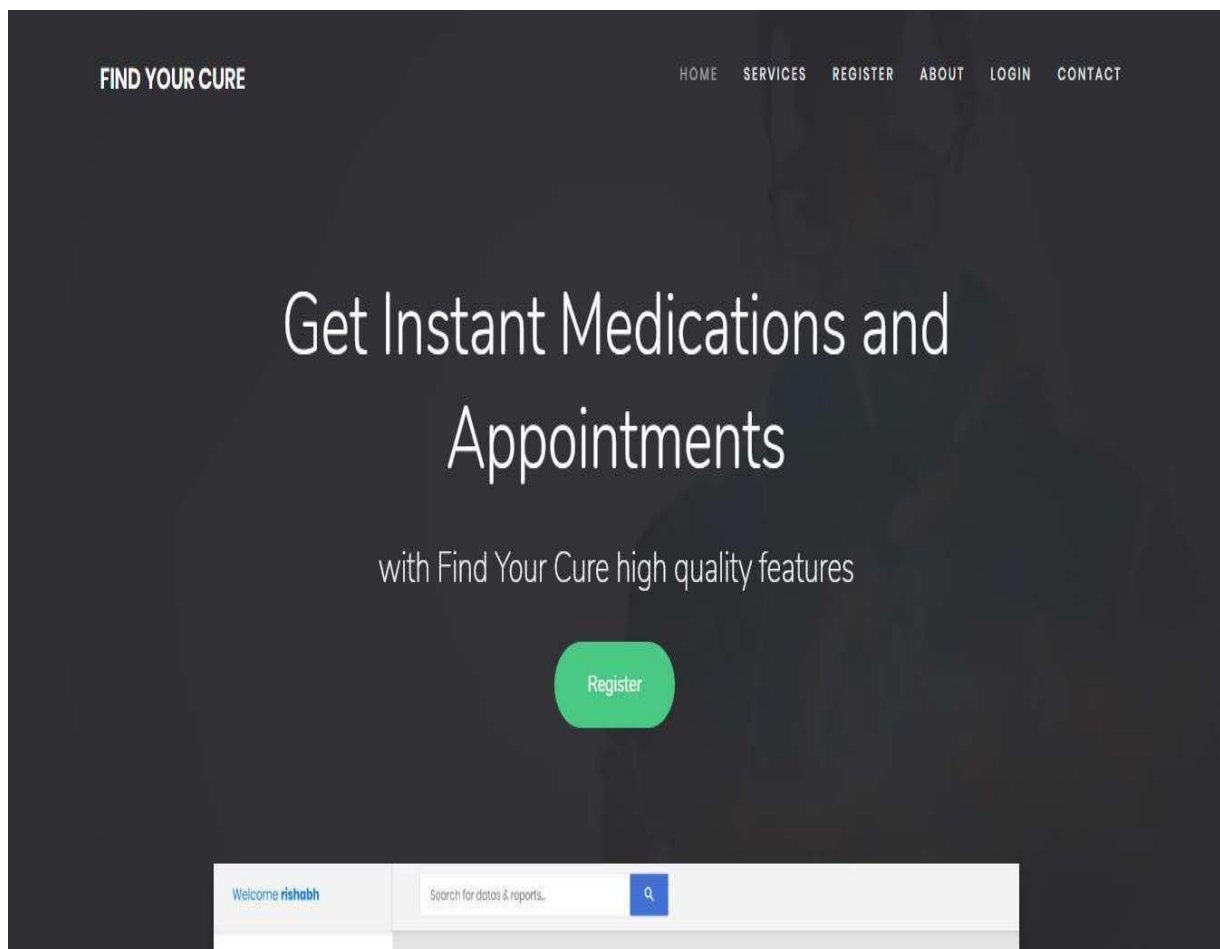


Fig 4.1

CREATE ACCOUNT

Your First Name

Your Last Name

Enter Your Age

Your Email

Male Female Other

Password ⊗

Repeat your password

Fig 4.2

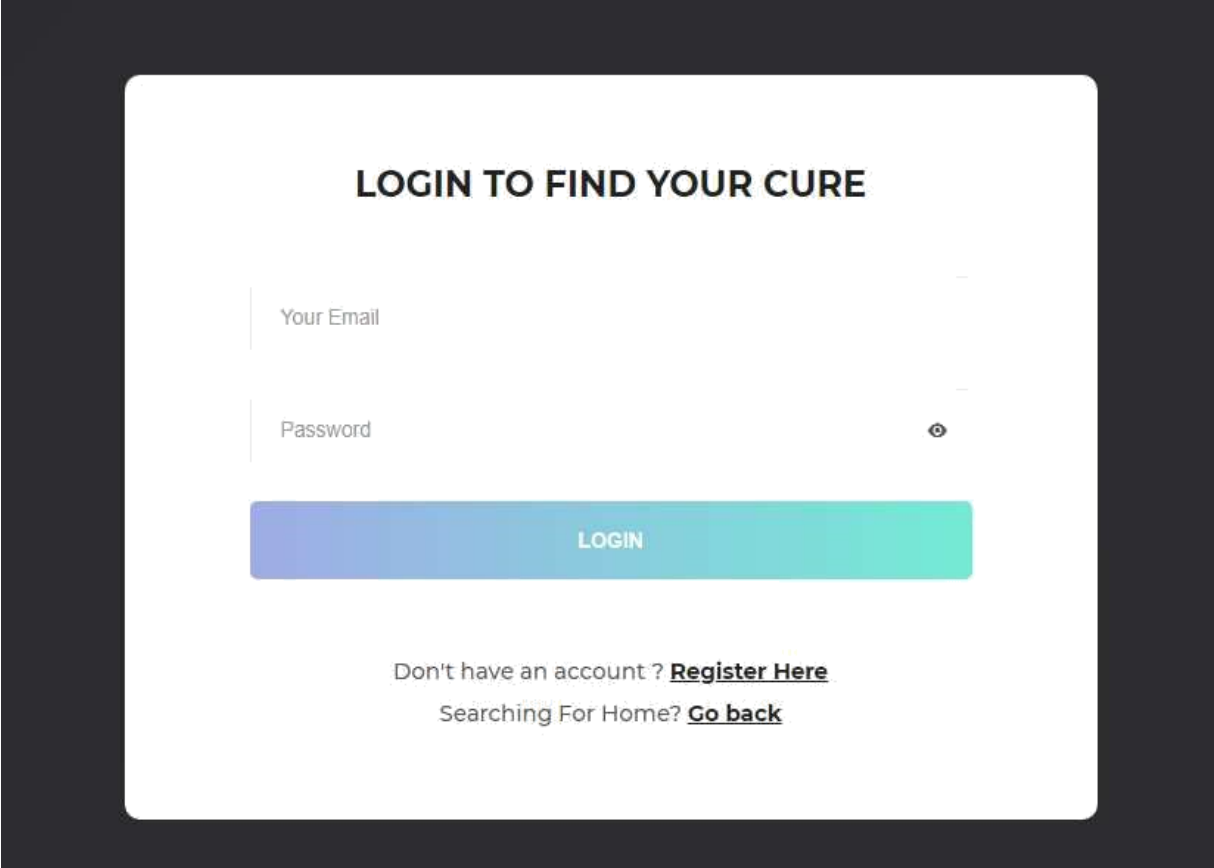


Fig 4.3

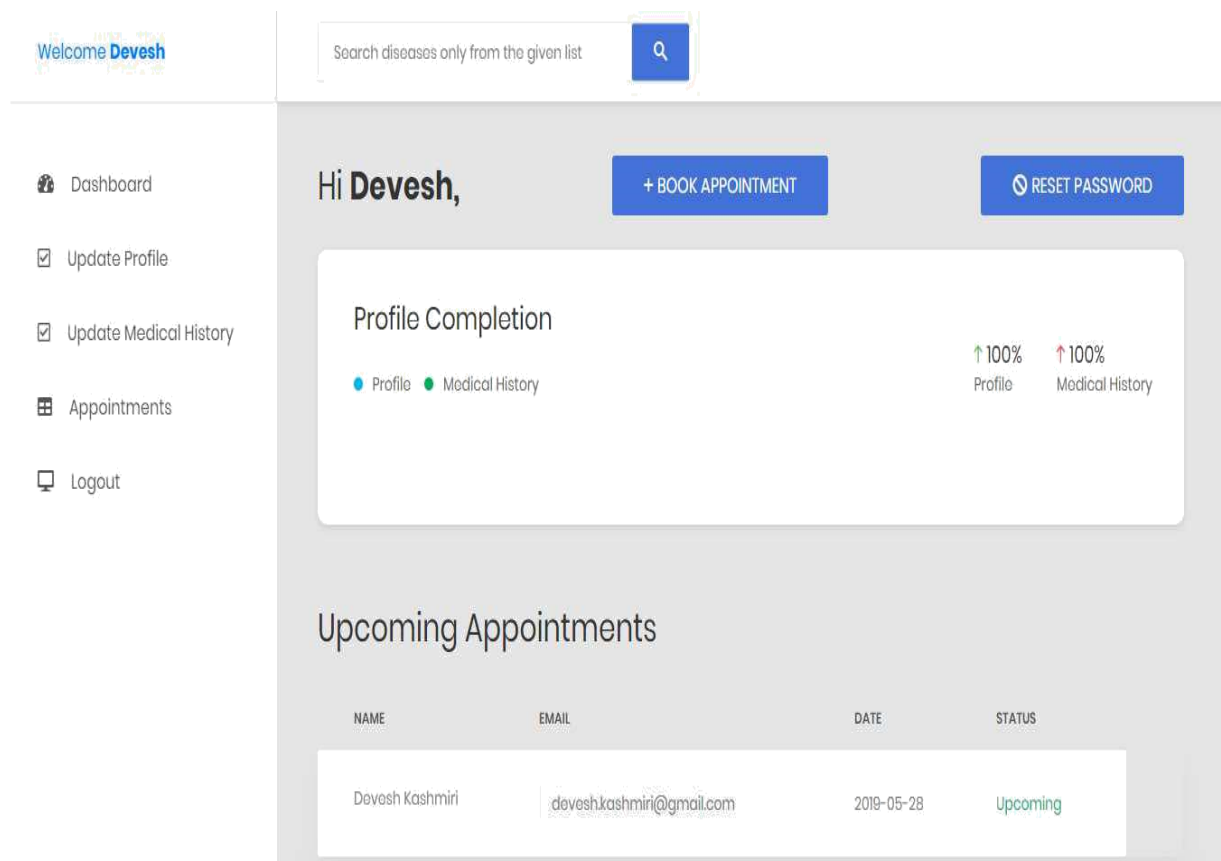


Fig 4.4

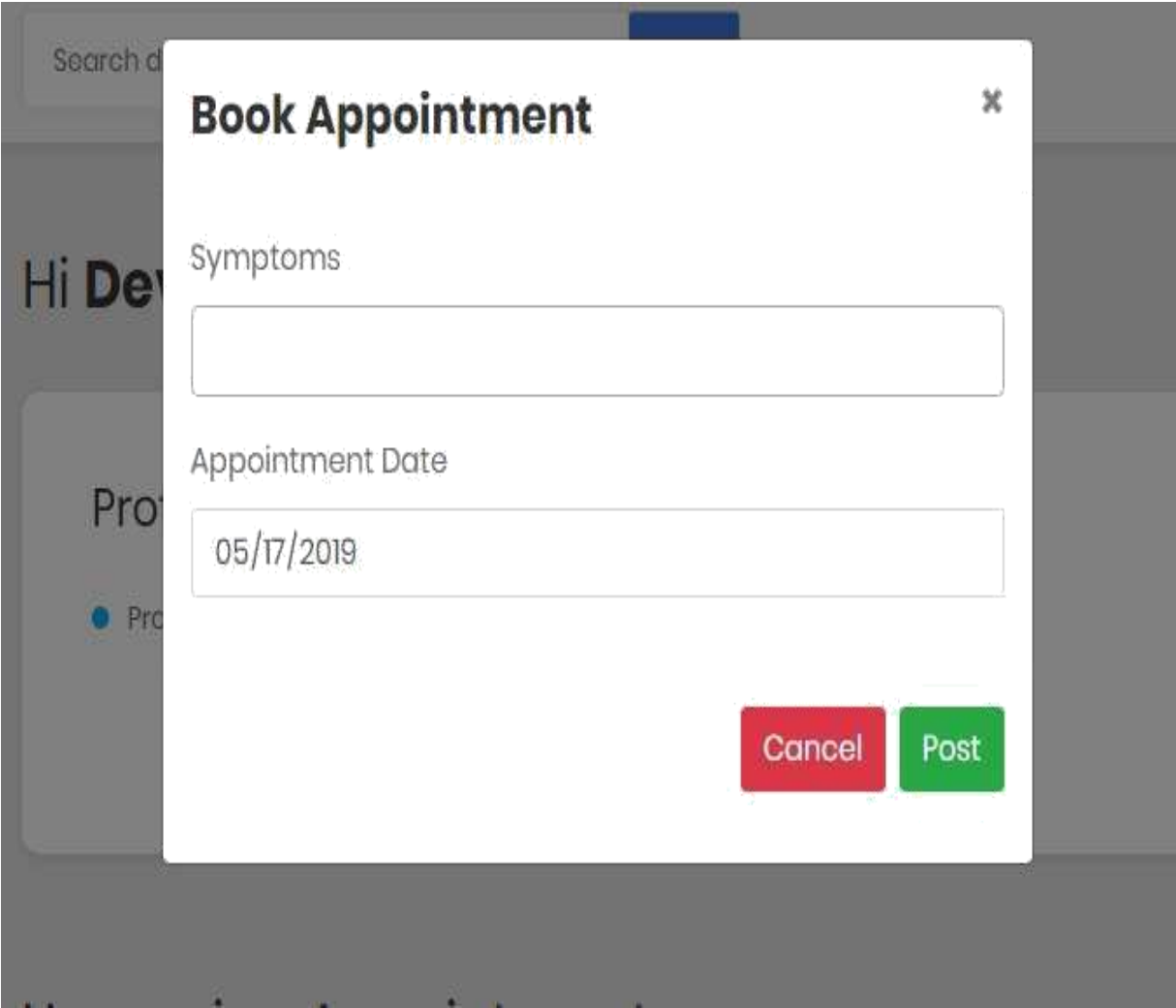


Fig 4.5

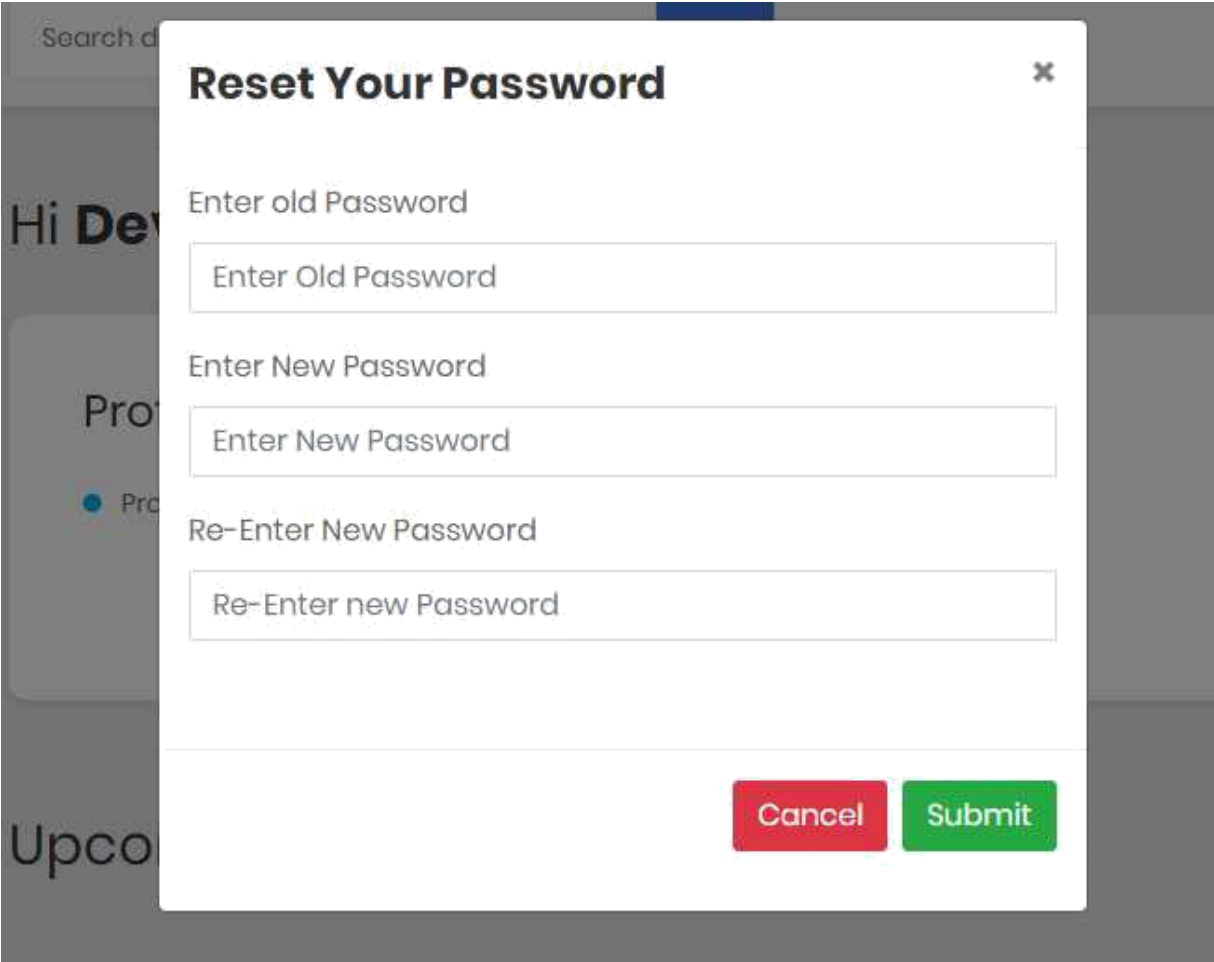


Fig 4.6

Welcome **Devesh**

Search diseases only from the given list



- Dashboard
- Update Profile
- Update Medical History
- Appointments
- Logout

Profile

First Name

Devesh

Last Name

Kashmiri

Age

22

Street

Hootagalli

City

Mysuru

Postal Code

570027

Family Annual Income

365000

Fig 4.7

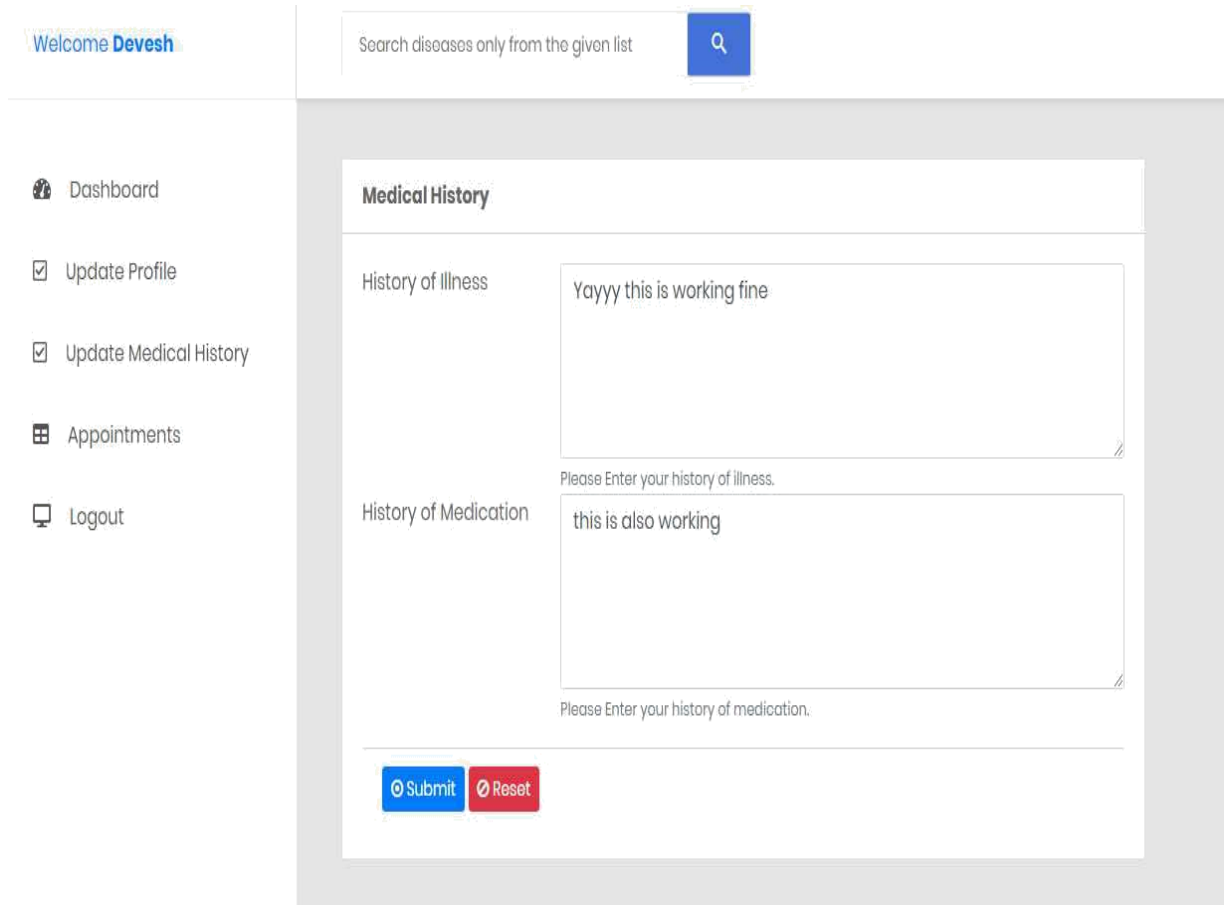


Fig 4.8

Welcome **Devesh**

Search diseases only from the given list

[+ MAKE NEW APPOINTMENT](#)

All Appointments

NAME	EMAIL	PROBABLE DISEASE	DATE	
Dr. Devesh	deveshkashmiri@gmail.com	None	2019-05-31	
Dr. Devesh	deveshkashmiri@gmail.com	None	2019-05-28	
Dr. Tyrion	whoremonger@imp.com	None	2019-05-15	
Dr. Devesh	deveshkashmiri@gmail.com	None	2019-05-13	

Navigation menu:

- Dashboard
- Update Profile
- Update Medical History
- Appointments
- Logout

Fig 4.9

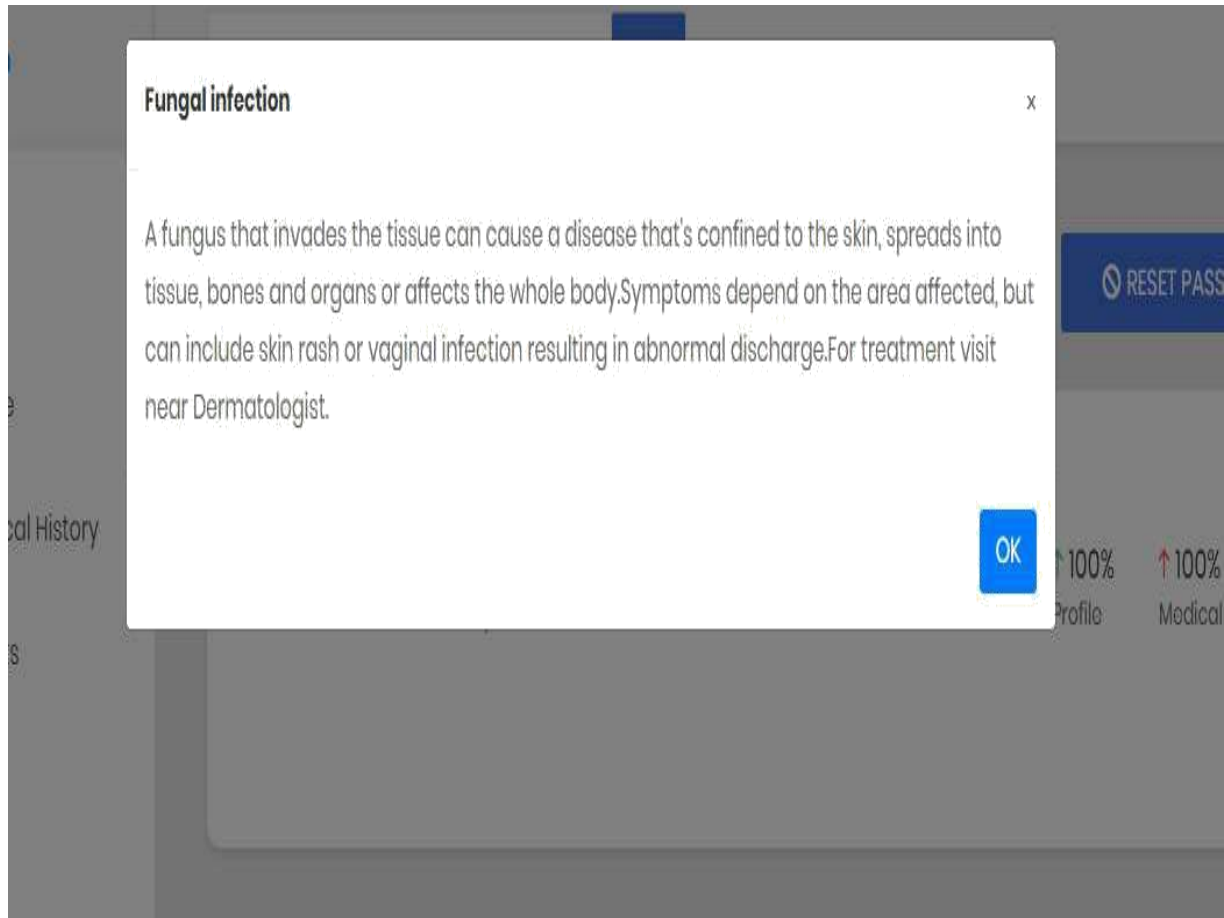







Fig 4.10




Welcome **Dr. Devesh,**

-  Dashboard
-  Update Profile
-  Appointments
-  Add Patient
-  Logout

Hi **Dr. Devesh,**

 RESET PASSWORD

Next Appointments

NAME	EMAIL	SYMPTOMS	DATE			
Devesh Kashmiri	kashmiridevesh@gmail.com	symptoms	2019-05-28			
Devesh Kashmiri	kashmiridevesh@gmail.com	symptoms	2019-05-31			

Previous Appointments

Name	Email	Symptoms
------	-------	----------

Fig 4.11

Welcome **Dr. Devesh,**

 Dashboard

Update Profile

 Appointments

+ Add Patient

 Logout

Kashmiri	kashmiridevesh@gmail.com	symptoms	28			
----------	--------------------------	----------	----	---	---	---

Devesh Kashmiri	kashmiridevesh@gmail.com	symptoms	2019-05-31			
--------------------	--------------------------	----------	------------	---	---	---

Previous Appointments

Name	Email	Symptoms
Joma Tharish	joma.j@infosys.com	['itching', 'skin_rash', 'vomiting']
Devesh Kashmiri	kashmiridevesh@gmail.com	['itching', 'continuous_sneezing', 'shivering']
Devesh Kashmiri	kashmiridevesh@gmail.com	symptoms
Devesh Kashmiri	kashmiridevesh@gmail.com	symptoms

Fig 4.12



- Dashboard
- Update Profile
- Appointments
- + Add Patient
- Logout

Profile

First Name

Devesh

Last Name

Kashmiri

Age

22

Street

Hootagalli

City

Mysuru

Postal Code

Visiting Hours Start

10:00 AM

Visiting Hours End

05:00 PM

Fee Structure

1000

Doctor Specialization

Years od Experience

Years of Experience

Fig 4.13

Welcome **Dr. Devesh,**

 Dashboard

 Update Profile

 Appointments

 Add Patient

 Logout

All Appointments

Patient Name	Patient Email	Patient Symptoms
Devesh Kashmiri	kashmiridevesh@gmail.com	symptoms
Devesh Kashmiri	kashmiridevesh@gmail.com	symptoms
Jama Tharish	jama.j@infosys.com	['itching', 'skin_rash', 'vomiting']
Devesh Kashmiri	kashmiridevesh@gmail.com	['itching', 'continuous_sneezing', 'shivering']
Devesh Kashmiri	kashmiridevesh@gmail.com	symptoms
Devesh Kashmiri	kashmiridevesh@gmail.com	symptoms

Fig 4.14

Welcome **Dr. Devesh,**

- Dashboard
- Update Profile
- Appointments
- + Add Patient
- Logout

Profile

First Name

Last Name

Age

E-Mail

Gender
 Male Female Other

History of Illness

History of Medication

Fig 4.15

Patient Name	History of Illness	History of medication
Devesh Kashmiri	Yayyy this is working fine	this is also working

[Go Back](#)

Fig 4.16

5.1 Conclusions

The primary point of the proposed project is to build up a web application which can be valuable for managing doctor-patient activity. We have gotten a new system prepared in the wake of recognizing issues in existing manual framework. In this, a simple to utilize GUI is proposed which can be used by all sorts of doctors and patients. In the project, we made an endeavor to successfully present the idea of Finding cure for patients registering on our website, and for doctors to grow their work. We portray the proposed framework and clarify the features executed by our proposed framework. We likewise give a concise review of the technologies utilized amid the development of our proposed framework. This venture can be additionally refined and reached out by presenting new and increasingly creative features.

The project will assist the particular appointments to be managed and automated. It will diminish the human exertion and make the task of patient, and doctor simpler. It is productive to utilize and simple to work on. Thus remembering the focal points and applications; we are building up an Find your Cure application which has absolute management control of patient and doctor and particular administration of various appointments.

The project was successfully implemented and tested for quality, accuracy and performance. All the objectives and requirements were accomplished and all the needs are met.

5.2 Future Scope

It can be extended to reach to a lot of the people on their mobile handsets, by using android application and enhanced features.

It can have an additional functionality where doctors can register themselves and add or, update or delete their personal thinking of symptoms according to the availability of existing symptoms.

Its performance can be increased by making it perform even during the peak loads.

Logs can be maintained for the changes being done by the user. The timestamp and person responsible should be saved for both of them. More offers and deals can be presented on the website to attract more customers and increase its scalability.

The site can be inculcated with payment options.

5.3 Goals Achieved

Reduced manual work

Easy information retrieval

**Reduced errors due to human
intervention User friendly**

Flexible for further enhancement

All doctors and patients on one platform

Easy, convenient and efficient for event organizers