**HEALTHWAVE**

#### A MINOR PROJECT REPORT

**Submitted in partial fulfilment of the requirement for the award of Degree of Master of Computer Applications**

**Submitted to**



**RAJIV GANDHI PRODYOGIKI VISHWAVIDYALAYA, BHOPAL (M.P.)**

**Submitted by:**

**Mr. Rishabh Shukla Under the Supervision of**

**Enrollment No 0827CA16DD1 Prof. Jayant Bihare Mr. Animesh Chourey**

**Enrollment No 0827CA16DD05**



**MASTER OF COMPUTER APPLICATIONS**

**ACROPOLIS INSTITUTE OF TECHONOLOGY & RESEARCH, INDORE SESSION 2019-2020**

**BONA FIDE CERTIFICATE**

This is certified that this project report **“HEALTHWAVE”** is the bona fide work of **“Rishabh Shukla(0827CA16DD15) and Animesh Chourey (0827CA16DD07)”** who carried out the project under my supervision.

Signature of HOD Signature of Supervisor

Prof. Geeta Santhosh Prof. Jayant Bihare

#### HEAD OF THE DEPARTMENT Project Guide

**MCA (Associate Professor)**

**MCA**

**Project Coordinator External**

Prof. Jayant Bihare

Signature of Coordinator Signature of External

## EXECUTIVE SUMMARY

“HEALTHWAVE is a hospital website which will allow user access to all facility of the hospital and user can access form everywhere in the world. Our project HEALTHWAVE include appointment of the patient and storing the patient detail into the system. The all data is accessible by the administrator. Only they can add database of doctors and department. The data can be retrieved easily. The interface is user-friendly. The data are well protected for personal use and make the data processing very fast

**Language and Tools**

* Processor: CORE i3, i5
* Hard Disk:126 GB
* RAM :8 GB D. Software Requirements
* Operating system: Windows XP/8/10
* Database: SQLite
* Database connectivity: Django
* Server: Apache Tomcat 5/6
* Scripting: Python, Html, Css.
* Server Side: Python Django
* Python version: Python **3.6**

**Software Development Life Cycle**

* 1. **Planning:** The planning is the foremost phase in the systems advancement process and it determines that whether there is a need for the new system or not. So we planned to build an application for Hospital Online Website named HealthWave.
  2. **System Analysis and Requirement:** In this phase we determine the need of the business, responsible persons of the project and the deadline of the project
  3. **System Design:** The system design is the third phase of SDLC and this explains the required specifications, features of operations of the proposed system. The hardware components, software components, and network capabilities are considered in this phase.
  4. **Development:** In the fourth phase of SDLC we did all the backend work like programming and database connectivity.
  5. **Integration and Testing:** In this phase, we test the proposed system and the same time the system’s integration and system’s testing will be done at this stage.
  6. **Implementation:** In the sixth phase of SDLC, the major codes of the programs are written and the proposed system will be installed.
  7. **Maintenance:** This is the final phase of SDLC and this involves the maintenance and time to time updating of the system.

## Requirement Analysis

#### Requirement specification

Functional Requirements Specification It maintains two level of user: -

1. Administrator level
2. User level The Software include

* To increase efficiency of managing the Hospital, Doctors.
* It tracks all the information of Doctors, Employee.
* Manage information of doctors.
* Manage the details about appointments.
* Keeps tracks of the donations.

#### Non-Functional Requirement:

Reliability: The capability to maintain the specified level of performance is called reliability. This application is a web-based application that runs on any device that has a browser. Availability: The application will be available for 99% of the time and Security: The business logic is hidden from the users and is much safer and thus avoids unauthorized or illegal access or database corruption. Security of the user’s information is also safe as there is a login facility. Maintainability:

Maintenance is typically done after the software development has been completed. As the time evolves, so do the requirements and needs.

* 1. **Security Requirement:** The project provides post facility only to admin other than admin other user only read post no one can add or change the post. Only registered user can get access of **“HealthWave”**.
  2. **Safety Requirement**: All the information of user should be handled properly on the client side as well as server side**.**

**2.2 Performance Requirement:** The User Interface of application should be fast and responsive. The application should be run on fast server so user don’t wait for any action perform in application.

## Design Constraints

1. **User Interface Constraints:** Using this system is fairly simple and intuitive. A user familiar with basic navigation skills should be able to understand all functionality provided by the system.
2. **Software Constraints:** The system will be intended to run on Python 3.6 and above.
3. **Assumptions and dependencies:** Most of hospital websites have a lot of redundant features which are rarely used. Our new system focuses on the features which are most important to the patient who wants to view our website and the can also book the appointment to visit the doctor.
4. **Hardware Constraints:** The system should have 2gb or above RAM.
5. **System Design Constraints:** The system shall be implemented in Django.

## Project Plan

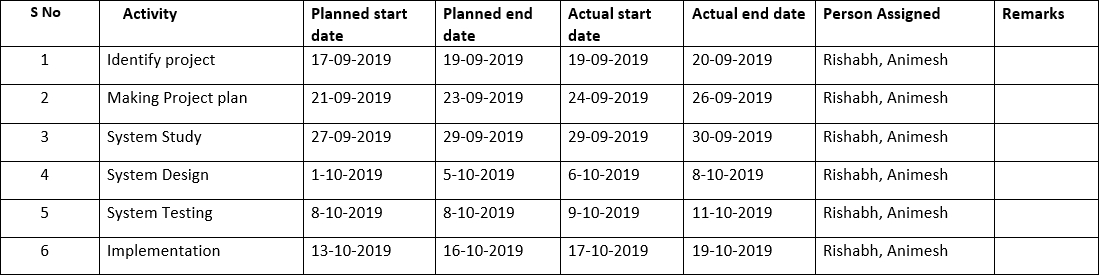
1. **1.Scope management**
2. 1. HEALTHWAVE will provide secure access to documents. It uses authenticity services provided by User ID’s.
3. 2. HEALTHWAVE aimed at eliminating the many facility of the hospitals. We can view the complete facility of hospital from everywhere in the world.
4. 3. HEALTHWAVE provides a dedicated personal e-storage space to users.
5. 4. HEALTHWAVE reduce the administrative overheads of hospital. It is also easy for users to receive the services available at hospital and viewing every information of hospital.
6. 5. To sign up f or your HEALTHWAVE, you need your Login details which you fill when you register in HEALTHWAVE.

#### People Management:

All users will be easily managed by HealthWave. Every

user can book the appointment and get access to departments. Admin has access of every user he can manage all users easily.

# 2.People management



1. **Risk Management:** Five main risk impact areas
   * New, unproven technologies
   * User and functional requirements
   * Application and system architecture
   * Performance

#### Communication Plan:

There’s no single right way to communicate on a project. In fact, our communication plan can and should include a variety of communication methods. Here are a few to consider:

* + Email
  + Meetings (in-person, phone, or video chat)
  + Discussion boards
  + Status reports

#### Time Management Plan:

After we have identified and prioritize the tasks it is time to estimate how much time it will take to complete them. We include all constraints and other factors in our estimation. Most of the time things will not go as planned, so include buffer times for unforeseen events into our project schedule.

After all the planning and estimating, we create a project schedule which includes all tasks and their duration.

**Feasibility Analysis**

The following are the various types of feasibility studies:

1. **Technical Feasibility:** This is concerned with specifying the equipments and the software to satisfy the user requirements. The technical needs of the system vary considerably but include:
   * The facility to produce outputs in a given time.
   * Response time under certain conditions.
   * Ability to process a certain volume of transactions at a specified speed.
2. **Economical Feasibility:** Economic analysis is the most frequently used method for evaluating the effectiveness of a new system. More commonly known as cost/benefit analysis, the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system.
3. **Operational Feasibility:** We show a proper error message when any mistakes are made by any message. We provide help and a guideline menu to help the user.

#### Feasibility Study Report

The result of the Feasibility Study provides us with the following facts:

* + The automated system would increase the efficiency of the system.
  + The automated system would increase customer's satisfaction.
  + The automated system has many requirements such as Efficiency cost effectiveness, prompt service, Reliability.
  + The automated system should be simple to use, incorporate all necessary services and maintainable.

**Methodology Adopted**

A methodology is a model, which we employ for the design, planning, implementation and achievement of their project objectives. We adopted Software Development Life Cycle (SDLC) for this software development.

**SDLC:** This is a conceptual model used in software development project. SDLC also heavily emphasizes on the use of documentation and has strict guidelines on it. we used all the phases of SDLC to develop our project.

**Phases of SDLC:**

* + Requirement gathering and analysis
  + Design
  + Implementation or coding
  + Testing
  + Deployment
  + Maintenance

## System Design

**System design** is the process of designing the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system.

Here are some elements of system

* **Architecture** - This is the conceptual model that defines the structure, behavior and more views of a system. We used flowcharts to describe and illustrate our system.
* **Modules -** This are components that handle one specific task in a system. A combination of the modules makes up the system.

Some modules of our system are:

#### Login Module

1. **Appointment module**
2. **Donate Module**
3. **Contact Module**

* **Components -** This provides a particular function or group of related functions. They are made up of modules.
* **Interfaces -** This is the shared boundary across which the components of the system exchange information and relate.

## Coding Methods

We used modular programming method for our software development rather than using top down or structured programming method.

**Modular Programming:** Modular programming is the process of subdividing a computer program into separate sub-programs. A module is a separate software component. It can often be used in a variety of applications and functions with other components of the system.

* + Each sub-module contains something necessary to execute only one aspect of the desired functionality.
  + Modular programming emphasis on breaking of large programs into small problems to increase the maintainability, readability of the code and to make the program handy to make any changes in future or to correct the errors.

Advantages of Using Modular Programming Approach –

* + **Ease of Use:** This approach allows simplicity, as rather than focusing on the entire thousands and millions of lines code in one go we can access it in the form of modules. This allows ease in debugging the code and prone to less error.
  + **Reusability:** It allows the user to reuse the functionality with a different interface without typing the whole program again.
  + **Ease of Maintenance:** It helps in less collision at the time of working on modules, helping a team to work with proper collaboration while working on a large application.

## Testing

Testing is a process of executing a program with the aim of finding error. To make our software perform well it should be error free. If testing is done successfully it will remove all the errors from the software.

* + - All the test should meet the customer requirements
    - To make our software testing should be performed by third party
    - Exhaustive testing is not possible. As we need the optimal amount of testing based on the risk assessment of the application.
    - All the test to be conducted should be planned before implementing it

**Steps:** Software is tested from two different perspectives:

* 1. Internal program logic is exercised using ―White box‖ test case design Techniques.
  2. Software requirements are exercised using ―black box‖ test case Design techniques. In both cases, the intent is to find the maximum number of errors with the Minimum amount of effort and time.

#### Testing Methodologies:

A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high-level tests that validate major system functions against customer requirements. A strategy must provide guidance for the practitioner and a set of milestones for the manager. Because the steps of the test strategy occur at a time when

deadline pressure begins to rise, progress must be measurable and problems must surface as early as possible. Following testing techniques are well known and the same strategy is adopted during this project testing.

#### Types of Testing: -

* + 1. **Unit Testing**

It focuses on smallest unit of software design. In this we test an individual unit or groups of inter related units. It is often done by programmer by using sample input and observing its corresponding outputs.

Ex

* + - 1. In a program we are checking if loop, method or function is working fine
      2. Misunderstood or incorrect, arithmetic precedence.
      3. Incorrect initialization

#### Integration Testing

The objective is to take unit tested components and build a program structure that has been dictated by design. Integration testing is testing in which a group of components are combined to produce output.

Integration testing are of two types:

1. Top Down
2. Bottom up
3. Black Box testing: - It is used for validation. In this we ignore internal working mechanism and

#### Focuses on what is the output?

1. White Box testing: - It is used for verification. In this we focus on internal mechanism i.e.

#### How the output is achieved?

* + 1. **Test Cases:**

A test case is a specification of the inputs, execution conditions, testing procedure, and expected results that define a single test to be executed to achieve a particular [software testing](https://en.wikipedia.org/wiki/Software_testing) objective, such as to exercise a particular program path or to verify compliance with a specific requirement , a test case is a set of conditions or variables under which a tester determines whether the software satisfies requirements and functions properly.

A test case has pre-requisites, input values and expected results in a documented form which cover the different test scenarios. Test cases are the set of positive and negative executable steps of a test scenario which has a set of pre-conditions, test data, expected result, post-conditions and actual results.

Test cases have a great impact on the testing phase. Writing test cases is almost as important as the testing process itself. The activity of writing test cases helps you think through the details and ensures you’re approaching the tests from as many angles as possible. The value of having test cases long-term is that anyone can go in and retest using the test case. Test cases are powerful artifacts that are beneficial to future teammates, as well as a good source of truth for how a system and particular feature work.

## Maintenance

Software maintenance is widely accepted as a part of SDLC nowadays. It stands for all the modifications and updates done after the delivery of software product.

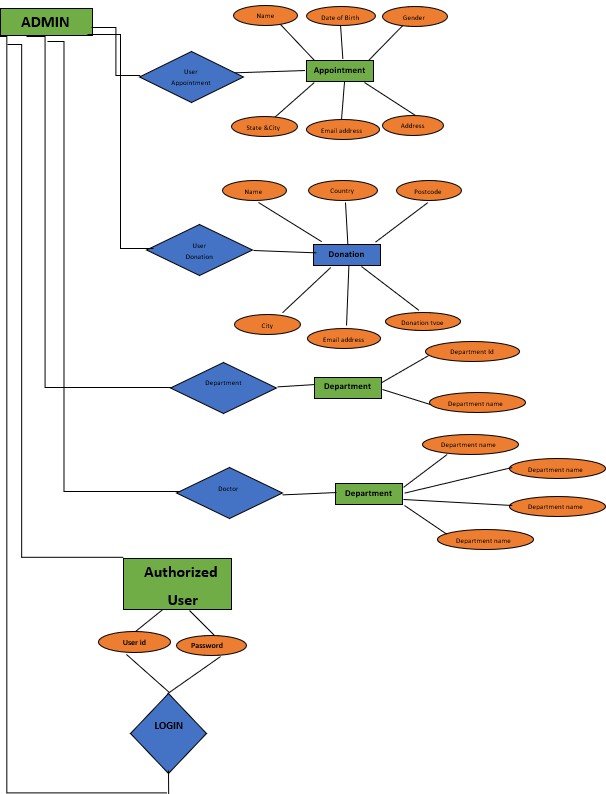
Following are some types of maintenance based on their characteristics:

* + - * **Corrective Maintenance** - This includes modifications and updates done in order to correct or fix problems, which are either discovered by user or concluded by user error reports.
      * **Adaptive Maintenance** - This includes modifications and updates applied to keep the software product up-to date and tuned to the ever changing world of technology and business environment.
      * **Perfective Maintenance** - This includes modifications and updates done in order to keep the software usable over long period of time. It includes new features, new user requirements for refining the software and improve its reliability and performance.
      * **Preventive Maintenance** - This includes modifications and updates to prevent future problems of the software. It aims to attend problems, which are not significant at this moment but may cause serious issues in future.

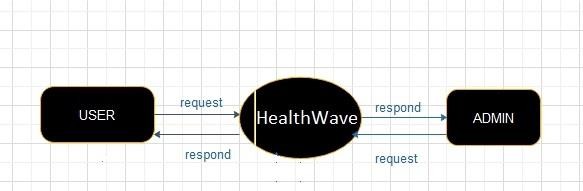
## List of Tables:

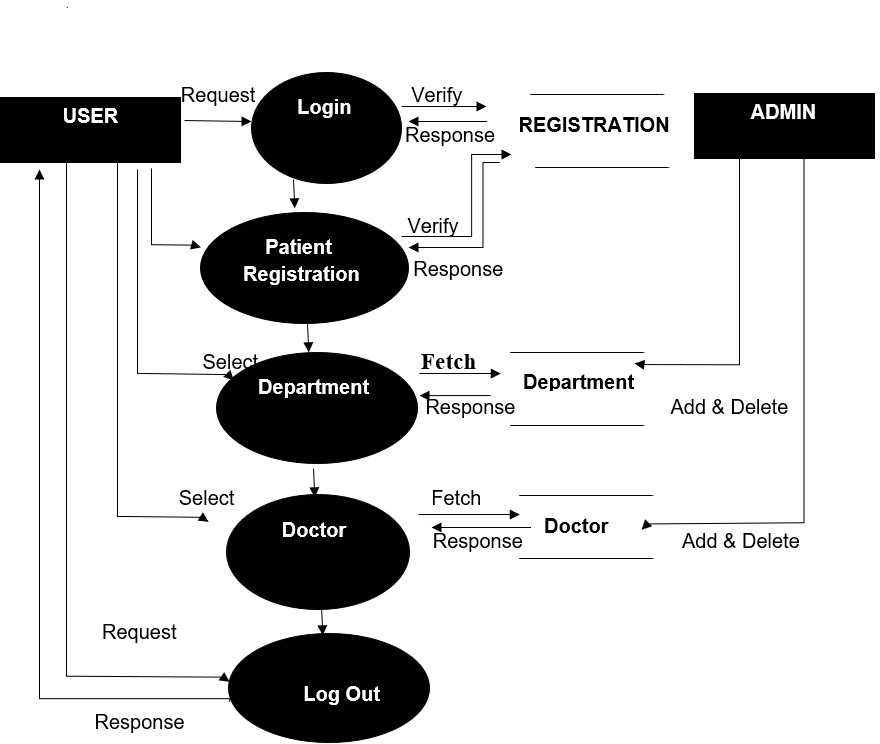
1. Appointment
2. Department
3. Donation
4. Login
5. Contact
6. Admin Info
7. Doctor

**ER-Diagram**



**Data Flow Diagram**



Level-1 DFD

## Screenshots of User Side

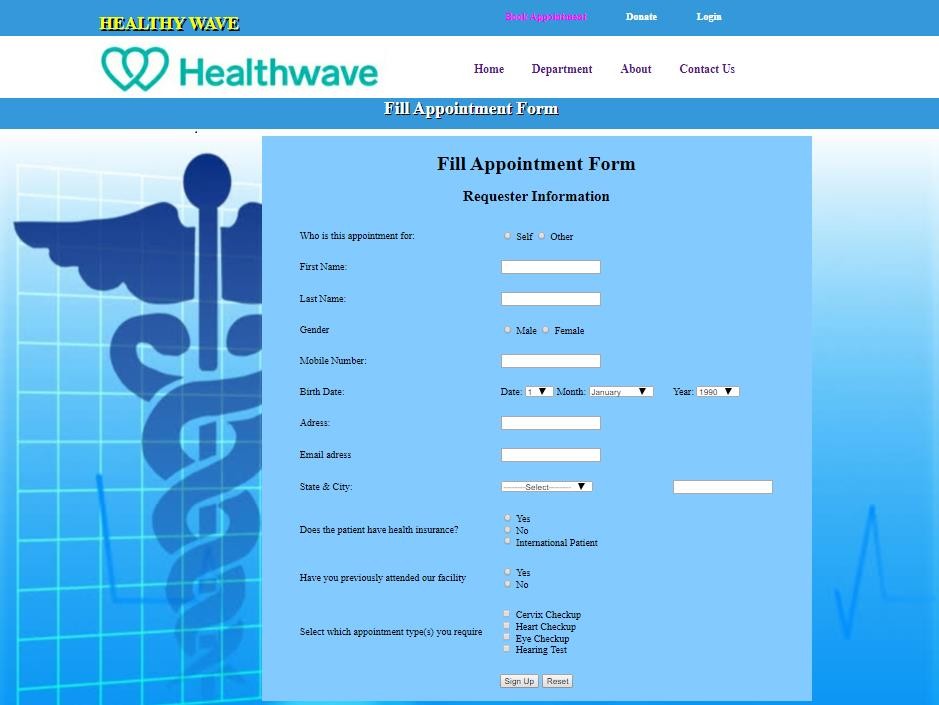
**Home page**



**Department page**



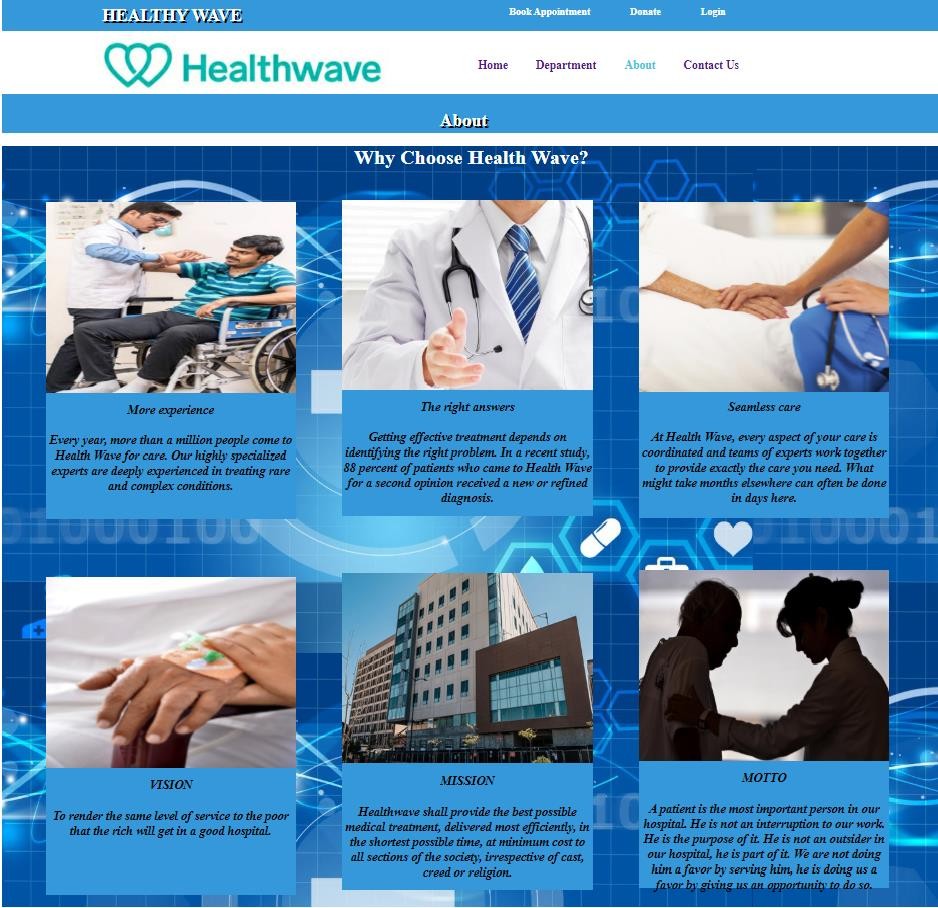
**Appointment page**



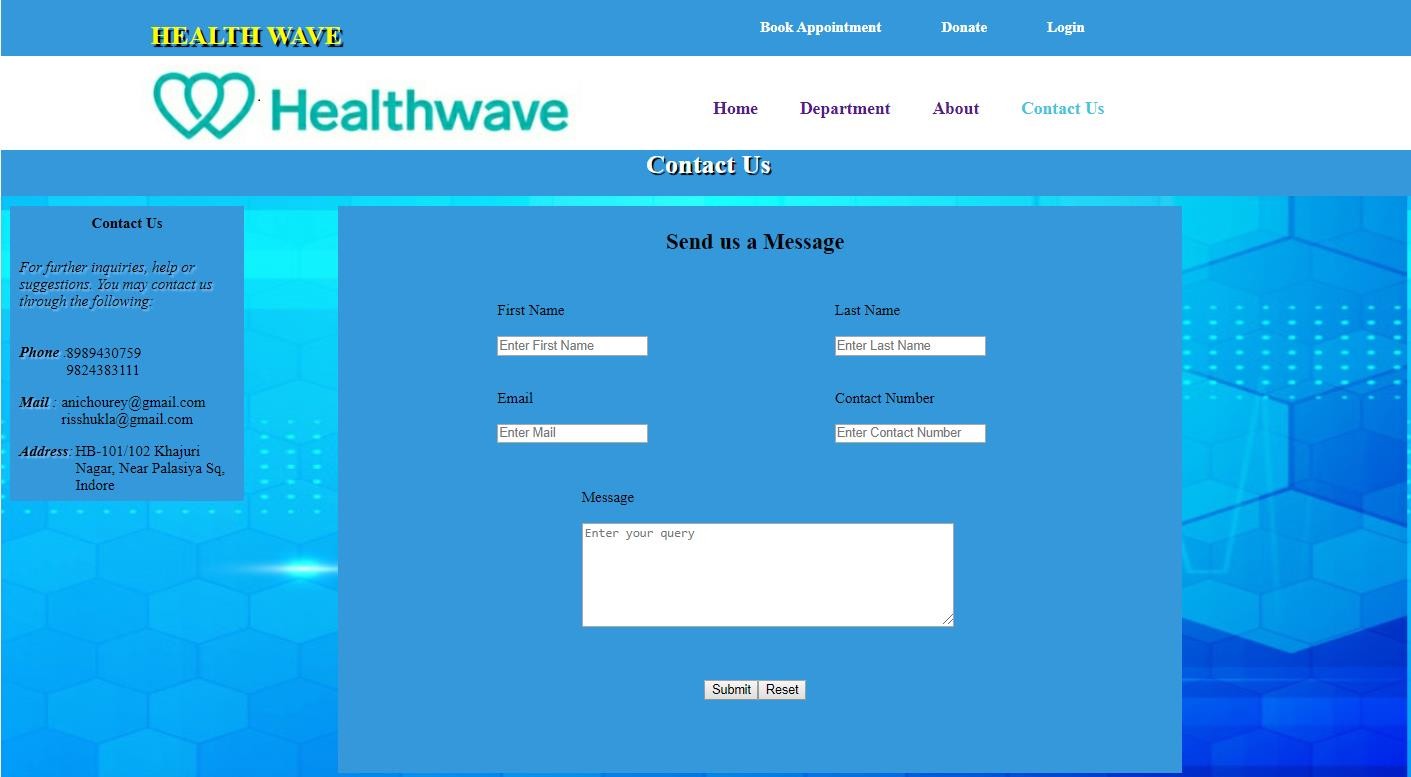
**Donation Page**



**About Page**



**Contact Page**

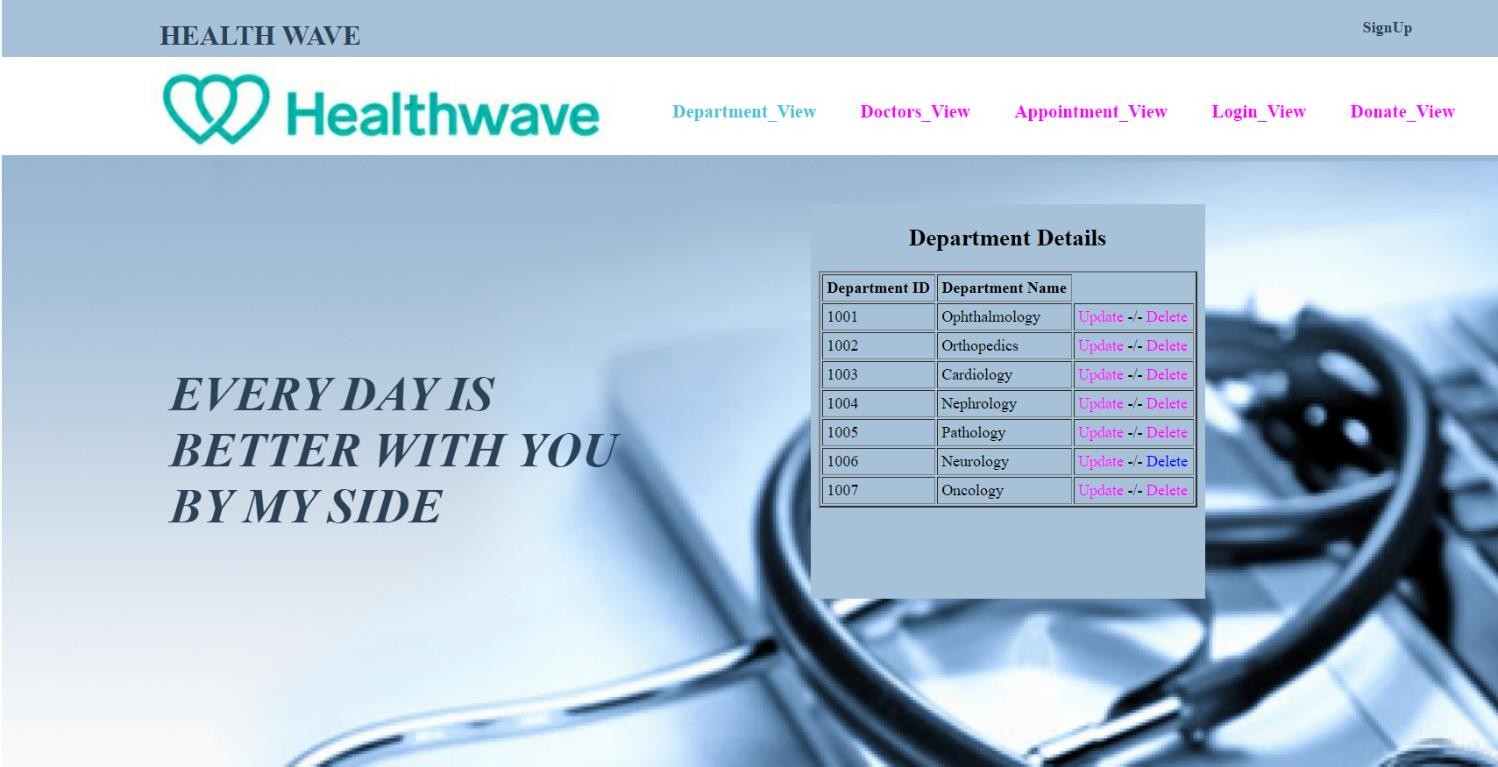


**Screenshots of Admin Side**

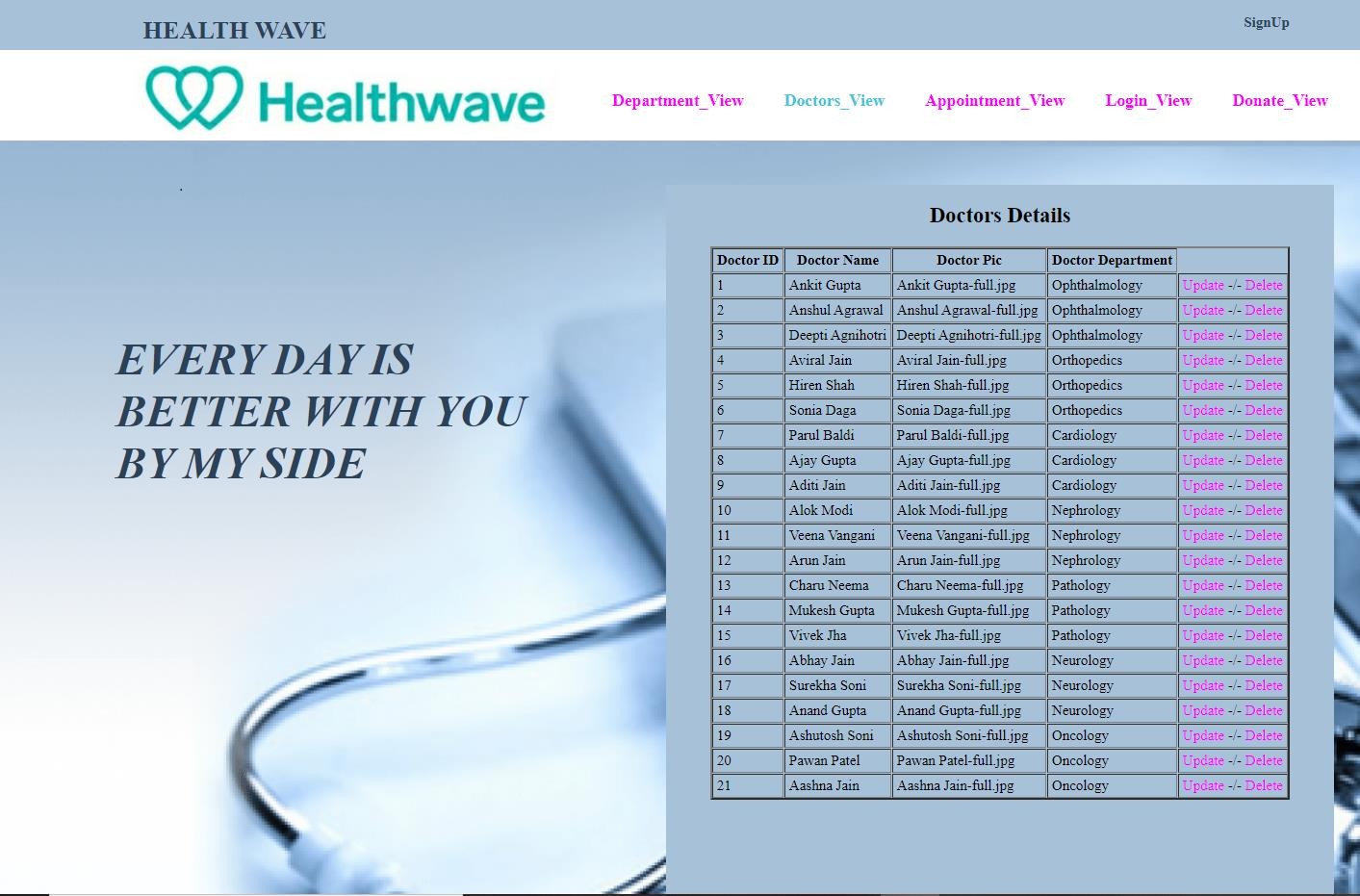
**Admin Home Page**



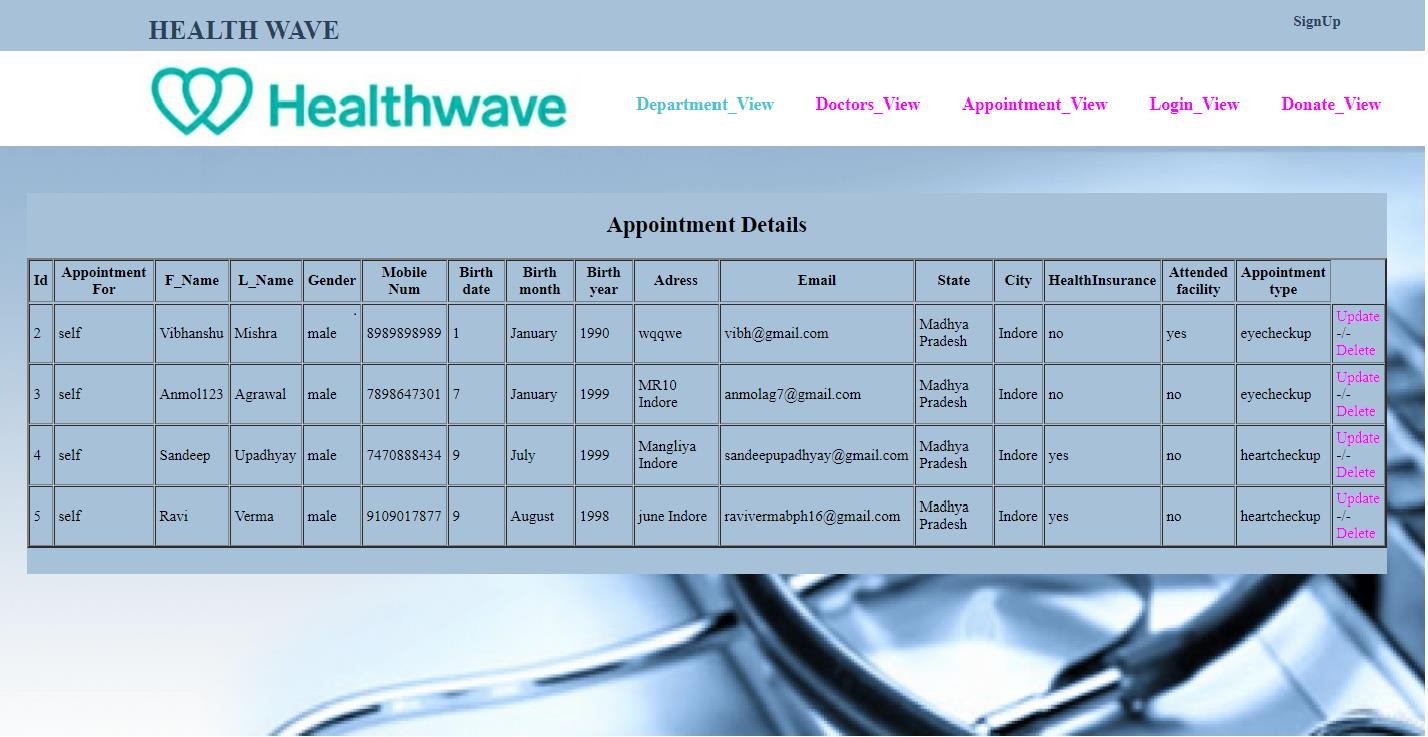
**Admin Department Page**



**Admin Doctor Info Page**



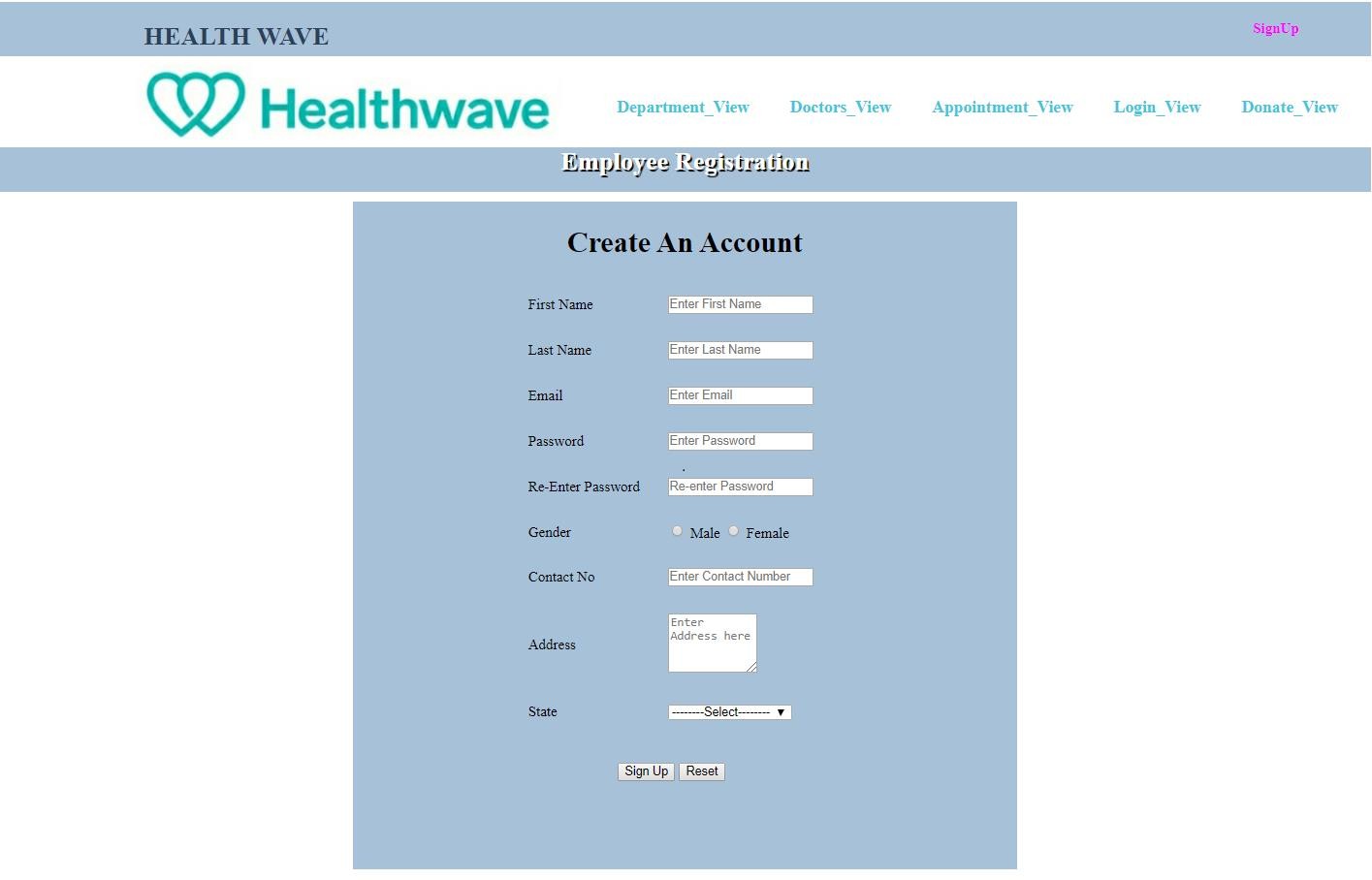
**Admin Appointment Info Page**



**Admin Donation Info Page**



**Admin Signup Page**



**References**

### Elmasri and Navathe, “Fundamentals of Database Systems”, 3/e, Addison- Wesley, 2001.

* [www.healthwave.com](http://www.healthwave.com/)
* Thomas M. Connolly, Carolyn E. Begg, “Database Systems Practical Approach to Design Implementation and Management”, 4/e, Addison- Wesley, 2005.
* [http://www.uml-diagrams.or](http://www.uml-diagrams.or/)