**INTRODUCTION**

1. **INTRODUCTION**

Our project entitled “ HEALTH CARE ” is a paperless hospital service, which is to register and store patient details/doctor details and retrieve these details automatically whenever required. It facilitates to access the patients to book their appointments online. Our project paperless hospital service also provide computerized prescribing, and labs. After the doctor enters all medications, an electronic copy of the patient’s prescription is sent to lab.

**1.1 PURPOSE OF THE PROJECT**

Our project is a website for hospital, implemented in PHP. This project is to provide interaction between hospital and patients. It provided the information about hospital, department and doctors, also provide online booking facility for patients. The system includes Computerized prescribing, labs. There are 4 type of users namely admin, patient, doctor, lab assistant. They are each provided with a username and password to access his/her account. Each type of users will be allowed to perform various activities according to their user type. The admin can add the details of doctors, department, lab assistance etc. The patient can be first register into this website and then login. And also can view the doctors and departments. Then take the appointment of a doctor for a day.

**1.2 SCOPE** **OF THE PROJECT**

Health care, this system is more efficient. Easily take the appointment of doctors. And view the appointment status and view the prescription details. The doctor can view the patient details and appointment details. Then add the prescription including the lab test. If the doctor can add the lab test the view the lab report of the patients. Lab assistant can be login to this site and view the patient details, and doctor added prescription and add the test results.

**SYSTEM ANALYSIS**

1. **SYSTEM ANALYSIS**

In system analysis, the problem is identified, alternative solutions are evaluated and the most feasible solutions are recommended. An initial investigation is performed to identify the current problems and solutions for the smooth functioning of the organization. Each module thoroughly studied and all the recommends for the project that are needed in order to satisfy the goals of the project are gathered. The problem is split into modules and is viewed at various angles. This leads to the evolution of the project. The system design should be approached like a building block. The main system has different subsystem. The identification of different modules and subsystem are the main activity during the system design. It includes input and output requirements.

**2.1 EXISTING SYSTEM**

Firstly, a detailed study of existing system was performed. Hospital currently uses a manual system for the management and maintenance of the critical information. The current system requires numerous paper forms, the prescription details of the patient is stored hospital in file format. There is possibility to loss the records at any time. There is no online booking facility.

* + 1. **LIMITATIONS OF EXISTING SYSTEM**
* No effective interaction.
* Time consuming.
* Consumes large volume of paper works.
* Manual works.
  1. **PROPOSED SYSTEM**

The proposed system is a website. The proposed system is designed to overcome all drawbacks of the existing system.The primary aim is to computerized all this manual works. There was an extreme need of reducing the paper works in the hospital, it is possible through the system “HEALTH CARE”. Its utilities are very use full and quick in nature.

**2.2.1 ADVANTAGES OF PROPOSED SYSTEM**

* User friendly
* Interaction became easier.
* Provide security and privacy for the data.
* Reduce manual effort.
* Reducing the processing time.
* Online booking facility for registered users.
* Computerized prescribing, lab.

**FEASIBILITY STUDY**

**3. FEASIBILITY STUDY**

The feasibility study concerns with the considerations made to verify whether the system fit to be developed in all terms. Main objective of feasibility study is to test the technical, social and economic feasibility of developing a system. This is done before developing a system. This is done by investigating the existing system in the area under investigation and generating ideas about the new system. The feasibility study to be conducted for this project involves:

* Technical Feasibility
* Economic Feasibility
* Operational Feasibility
* Behavioral Feasibility
* Legal Feasibility

**3.1 TECHNICAL FEASIBILITY**

Technical feasibility is the complete study of the project in terms of input, process, output, fields, programs and procedures. The system must be evaluated from the technical view point first. The assessment of this feasibility must be based on an outline design of the system requirement in terms of input, output, programs, procedure and staff. Having identified the outline of the system, the investigation must go on to suggest the type of equipment, required method of developing the system, and the method of running the system.

**3.2 ECONOMIC FEASIBILITY**

Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system. More commonly known as cost/benefit analysis; the procedure is to determine the benefits and savings that are expected from the proposed system and compare them with costs. If benefits outweigh costs, a decision is taken to design and implement the system. This is an ongoing effort that improves in accuracy at each phase of the system life cycle.

**3.3 OPERATIONAL FEASIBILITY**

It is mainly related to human organizational and political aspects. This test of feasibility asks if the system will work when it is developed and installed. The points considered are:

* What changes will be brought with the system?
* What organizational structures are disturbed?
* What new skills will be required?

**3.4 BEHAVIORAL FEASIBILITY**

People are inherently resistant to change and computers have been known to facilitate changes. The new tool does not require any staff maintenance. It doesn’t need any extra training to customers. The interfaces designed are easy to use. The proposed system is very user friendly. It reduces the drawbacks of the existing system. All these reasons make the new system behaviorally feasible.

**3.5 LEGAL FEASIBILITY**

Legal feasibility is to know if the proposed project confirms the legal and ethical requirement. A determination of any infringement, violation, or liability that could result from the development of the system. It is an evaluation of alternative approaches to the development of the system or product.

**SOFTWARE**

**REQUIREMENT**

**ANALYSIS**

**4. SOFTWARE REQUIREMENTS ANALYSIS**

**4.1 HARDWARE REQUIREMENTS**

SYSTEM - PENTIUM IV 2.4 GHZ

HARD DISK - 40 GB

RAM - 512 Mb

DISPLAY - COLOR MONITOR

KEYBOARD - WINDOWS COMPATIBLE

MOUSE - WINDOWS COMPATIBLE

**4.2 SOFTWARE REQUIREMENTS**

Operating System : Windows 8 or later

Database :MYSQL  
Web server : Apache Servers

Browser : Google Chrome/Internet Explorer

Other : PHP

**4.3 ABOUT THE SOFTWARE**

The project is done using PHP and ANDROID as front-end language and My SQL as the back end.

**4.3.1 PHP**

PHP is a server-side scripting language designed for Web development but also used as a general-purpose programming language. PHP is now installed on more than 20 million Web sites and 1 million Web servers. While PHP originally stood for Personal Home Page, it is now said to stand for PHP: Hypertext Preprocessor, a recursive acronym.

PHP code is interpreted by a Web server with a PHP processor module which generates the resulting Web page: PHP commands can be embedded directly into an HTML source document rather than calling an external file to process data. PHP includes free and open source libraries with the core build. PHP is a fundamentally Internet-aware system with modules built in for accessing File Transfer Protocol (FTP) servers, many database servers, embedded SQL libraries such as embedded Postage SQL, MySQL, Microsoft SQL Server and SQLite, LDAP servers, and others. PHP is commonly used as the P in this bundle alongside Linux, Apache and MySQL, although the P may also refer to Python, Perl, or some mix of the three.

**4.3.2 SUBLIME TEXT**

Sublime Text is a proprietary cross-platform source code editor with a Python application programming interface (API). It natively supports many programming language and markup languages, functions can be added by users with plugins, typically community-built and maintained under free-software licenses.

**FEATURES**

The following is a list of features of sublime Text:

* “Goto Anything”,quick navigation to files, symbols, or lines.
* “Command palatte” uses adaptive matching for quick keyboard invocation of arbitrary commands.
* Simultaneous editing: simultaneously make the same interactive changes to multiple selected areas.
* Python-based plugin API
* Project-specific preferences.
* Extensive customizability via JSON settings files, including project-specific and platform-specific settings.
* Cross-platform (Windows, macOS, Linux) and Supportive Plugins for cross-platform.
* Compatible with many language grammars from TextMate.

**4.3.3 MySQL**

This is a powerful relational database management system containing modifiable source code. It is the most preferable option for Linux users due to its robust, swift, and reliable structure. This is also a very fast and reliable database management system. The vendor of this database is Microsoft, and the database provides fast access to developers who are working on a Windows platform. . The software is compatible with both Windows and Linux platforms. MySQL is a relational database management system that provides the facility to manage databases.

**FEATURES**

MySQL also provides the following features:

* Support for languages such as Perl, Python, and PHP
* Support for a thread-based memory allocation system (Therefore, it is quite fast)
* Support for fixed as well as variable length records
* Support for a host-based verification system that provides security through verifying
* Passwords that are encrypted during transit
* Support for large databases
* Support for different types of field data types
* Support for UNIX sockets, TCP/IP sockets, and Named Pipes for providing connectivity

**The Internet Information Server (IIS)**

Internet Information Services (IIS, formerly Internet Information Server) is an extensible [web server](http://en.wikipedia.org/wiki/Web_server) created by [Microsoft](http://en.wikipedia.org/wiki/Microsoft) for use with [Windows NT](http://en.wikipedia.org/wiki/Windows_NT) family. IIS supports [HTTP](http://en.wikipedia.org/wiki/HTTP),  [HTTPS](http://en.wikipedia.org/wiki/HTTPS),  [FTP](http://en.wikipedia.org/wiki/File_Transfer_Protocol),  [FTPS](http://en.wikipedia.org/wiki/FTPS),  [SMTP](http://en.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol) and [NNTP](http://en.wikipedia.org/wiki/Network_News_Transfer_Protocol). It has been an integral part of the Windows NT family since [Windows NT 4.0](http://en.wikipedia.org/wiki/Windows_NT_4.0), though it may be absent from some editions (e.g. Windows XP Home edition). IIS is not turned on by default when Windows is installed. The IIS Manager is accessed through the [Microsoft Management Console](http://en.wikipedia.org/wiki/Microsoft_Management_Console) or Administrative Tools in the Control Panel.

**4.3.3.1 FURTHER ENHANCEMENT**

The software developed in php which makes the system more reliable and compatible with the other environments. The application proves better extensibility and flexibility for future enhancements. Enhancement means modifying, adding or redeveloping the code to support changes in the specification. Any further requirement application is possible with the same features guaranteed. The design of this software is in such a way that the addition of any new module if necessary is possible without affecting the integrity of the present system.

As a further enhancement, the tool could be developed into a fully featured IDE supporting all the features as any other IDE supports for developing projects. Users can quickly perform their task with a single browser.

**SYSTEM DESIGN AND**

**DEVELOPMENT**

**5. SYSTEM DESIGN AND DEVELOPMENT**

In [system engineering](http://en.wikipedia.org/wiki/Systems_engineering), modular design — or "modularity in design" — is an approach that subdivides a system into smaller parts (modules) that can be independently created and then used in different systems to drive multiple functionalities. Besides reduction in cost (due to lesser customization, and less learning time), and flexibility in design, modularity offers other benefits such as augmentation (adding new solution by merely plugging in a new module), and exclusion.

Applications are getting more and more complicated, and they are increasingly assembled from pieces developed independently. But they still need to be reliable. Modular coding enables you to achieve and manage that complexity. Splitting an application into modules has benefits for software quality. Modular software limits the risk of creeping coupled nest by requiring that different components of the system interoperate through well-defined API contracts to a method will return only once.

Modularity is about the interaction between systems, rather than between small parts of subsystems. Modular applications are composed of modules. Modularity gives systems clearer design and control of module interdependencies; it also gives developers more flexibility in maintenance.

Modular design will have large benefits for the architecture of the entire application as it grows from its infancy. The real benefits of modular programming might not be apparent in the first version of an application. But they will become obvious later with the reduced cost of creating the 2.0 and 3.0 versions. Since modular programming does not add significant cost to creating the 1.0 version of an application, there is little reason not to use this approach on all projects. A downside to modularity (and this depends on the extent of modularity) is that modular systems are not optimized for performance. This is usually due to the cost of putting up interfaces between modules.

Design is the first step in the development phase for any engineered product or system. Design is a creative process; a good design is the key to effective system. The term "Design" is defined as "The process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization ". It may be defined as the process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used.

From a project management point of view, software design is conducted in two steps. Preliminary design is concerned with the transformation of requirements in to data and software architecture. Detail design focuses on refinement to the architectural representation that lead to detail algorithm data structure and representation of software.

In System Design, high-end decisions are taken regarding the basic system architecture, platforms and tools to be used. The system design transforms a logical representation of what a given system is required to be in the physical specification. Design starts with the System Requirement Specification and converts it to a physical reality during the development. Important design factors such as reliability response time, throughput of the system maintainability, expandability etc should be taken into account to storage device. This is the difference between logical and physical data.

**5.1 INPUT DESIGN**

The user interface design is very important for any application. The interface design describes how the software communicates within itself, to system that interpreted with it and with humans who use it. The interface is a packaging for computer software if the interface is easy to learn, simple to use. If the interface design is very good, the user will fall into an interactive software application.

The input design is the process of converting the user-oriented inputs in to the computer-based format. The data is fed into the system using simple interactive forms. The forms have been supplied with messages so that user can enter data without facing any difficulty. The data is validated wherever it requires in the project. This ensures that only the correct data have been incorporated into the system.

The goal of designing input data is to make the automation as easy and free from errors as possible. For providing a good input design for the application easy data input and selection features are adopted. The input design requirements such as user friendliness, consistent format and Interactive Dialogue for giving the right message and help for the user at right time are also considered for the development of this project.

**5.2 OUTPUT DESIGN**

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any systems results of processing are communicated to the user and to other systems through outputs. In the output design it is determined how the information is to be displayed for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship with the user and helps in decision-making. The objective of the output design is to convey the information of all the past activities, current status and to emphasize important events. The output generally refers to the results and information that is generated from the system. Outputs from computers are required primarily to communicate the results of processing to the users. They are also used to provide a permanent copy of these results for later consultation.

**5.3 MODULE DESCRIPTION**

Modules of Health Care System are:

* **Admin:** This module will maintain the information hospital. The admin can add the details of department, doctors, ambulance, lab assistant. The admin can view the details of patients, lab test details.
* **Doctor:**  Doctor can login to this website and view the patients appointment details and it approved or reject the request. The doctor can add the prescription and view the lab test details.
* **Lab assistant:** The lab assistant can be login to this website and view the patient prescription details and add the test details.
* **Patient:** The patient can be first register into the website and login them. The patient can take the appointment and view its status and view the prescription, test result.

**5.4 DATA FLOW DIAGRAM (DFD)**

A DFD is a network that describes the flow of data throughout a system, data stores, and the processes that change, or transform data flows. The DFD network is a formal, logical abstract of a system that may have many possible physical configurations. For this reason, a set of symbols that do not imply a physical form is used to represent data sources, data flows, data transformations and data storage.

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 data flows 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 labeled to identify an external entity that is a source or destination of a data flow.

There are four symbols that are used in the drawing of Data Flow Diagrams:

* **Entities**

External entities represent the source of data that enter the system or the recipients of data that leave the system.

* **Process**

Processes represent activities in which data is manipulated by being stored or retrieved or transformed in some way. A circle represents it. The process will show the data transformation or change.

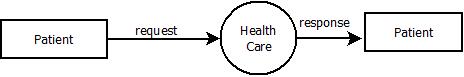
* **Databases**

Databases represent storage of data within the system.

* **Data flow**

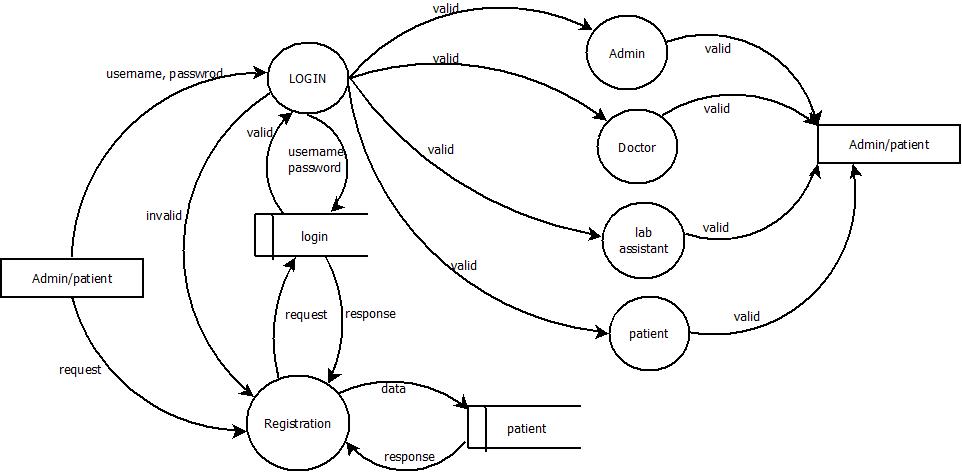
A data flow shows the flow of information from its source to its destination. A line represents a data flow, with arrowheads showing the direction of flow.

LEVEL 0



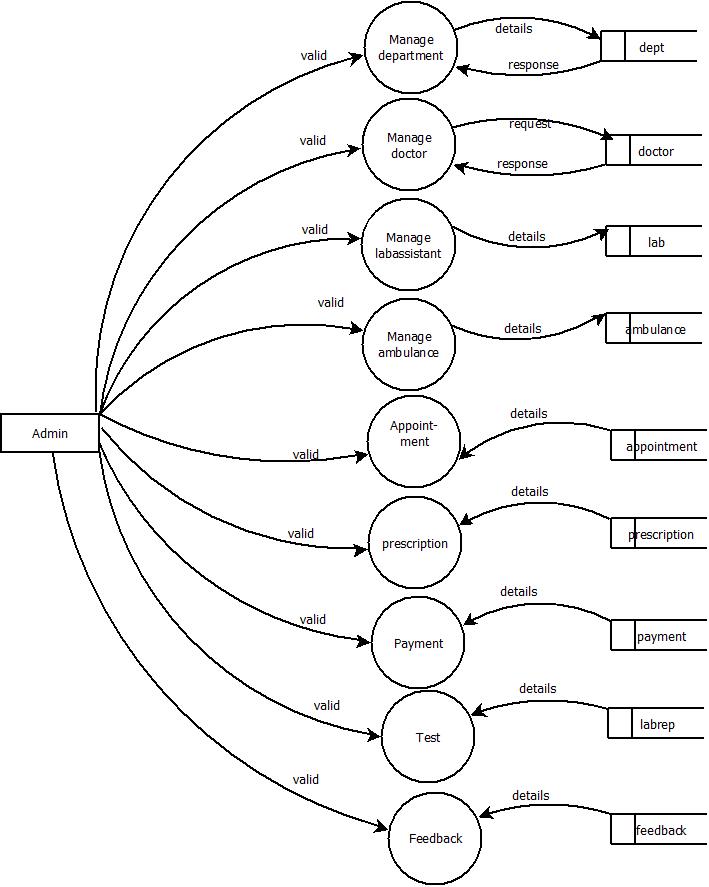
LEVEL 1

LOGIN



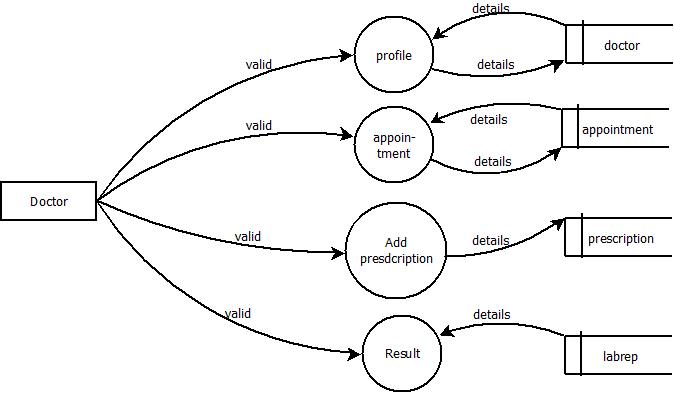
LEVEL 2

ADMIN



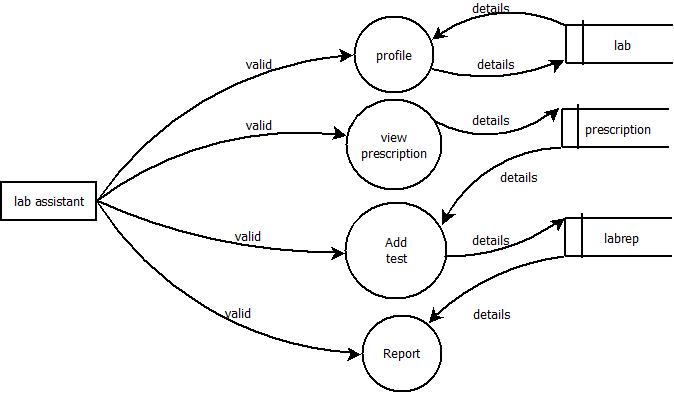
LEVEL 3

DOCTOR

****

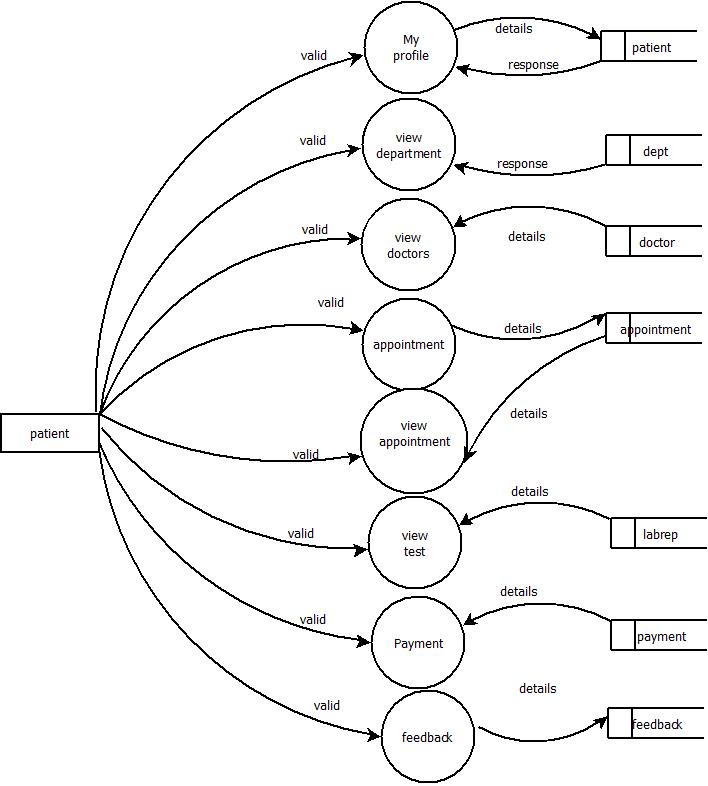
LEVEL 4

LAB ASSISTANT



LEVEL 5

PATIENT



**5.5 DATABASE DESIGN**

The most important aspect of building an application is the design of database. The data they store must be according to the user requirement. A well-designed database is essential for the good performance of the system. A database table known as a relation provides information related to specify entity.

**5.5.1 DATA NORMALIZATION**

The most important aspect of building an application is the design of database. The data they store must be according to the user requirement. A well-designed database is essential for the good performance of the system. A database table known as a relation provides information related to specify entity. The basic functions involved in a database system related to the information required by the user. It helps in,

* Minimization of duplication of data
* Enabling the model to be translated to database design
* All relations in a relational database are required to satisfy the following conditions

**Data in First Normal Form**

* Remove repeating data from table
* From the removed data, create one or more tables and relationships

**Data in Second Normal Form**

* Identify tables and relationships with one or more than one key
* Remove data that depends on only one part of the key
* From the removed data, create one or more tables and relationships

**Data in Third Normal Form**

* Remove that depends on other hand in the table or relationships
* From the removed data, create one or more tables and relationships

**Advantages of Normalization are:**

* Helps in complexity of maintaining data relationships
* It reduces inconsistency of data
* Eliminate the repeating fields
* Create a row for each occurrence of a repeated field

The second normal form has the characteristics of the first normal form and all the attributes must fully be depend on the primary key. The proposed system is using second normal form as it is found most suitable.

**5.5.2 TABLE DESIGN**

This is one of the major tasks in designing the database. It is important to realize that the design of the system is totally interrelated and so table design cannot really be considered in isolation from inputs, outputs, procedures, codes and security requirements.

**Database Name:** hmc

**Table 1:** login

**Primary Key:** uid

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **CONSTRAINTS** | **DESCRIPTION** |
| Uid | int | 11 | PRIMARY KEY | User id |
| Uname | Varchar | 30 | NOT NULL | User name |
| Upass | Varchar | 20 | NOT NULL | User password |
| Utype | Varchar | 30 | NOT NULL | User type |

**Table 2:** doctor

**Primary Key:**  id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **CONSTRAINTS** | **DESCRIPTION** |
| Id | int | 11 | PRIMARY KEY | Doctor id |
| Name | varchar | 30 | NOT NULL | Doctor name |
| Specialization | Varchar | 50 | NOT NULL | Specialization |
| Address | Varchar | 50 | NOT NULL | Address |
| Phno | Varchar | 12 | NOT NULL | Phone number |
| Email | Varchar | 40 | NOT NULL | Email id |
| Gender | Varchar | 10 | NOT NULL | Gender |
| Qualification | Varchar | 10 | NOT NULL | Qualification |
| Time | Varchar | 10 | NOT NULL | Time |
| Img | Varchar | 50 | NOT NULL | Images |
| Da | Varchar | 30 | NOT NULL | Day |

**Table 3:** patient

**Primary Key:** id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **CONSTRAINTS** | **DESCRIPTION** |
| Id | int | 11 | PRIMARY KEY | Patient id |
| Name | varchar | 50 | NOT NULL | Name |
| Address | Varchar | 50 | NOT NULL | Address |
| Phno | Varchar | 12 | NOT NULL | Phone number |
| Email | Varchar | 40 | NOT NULL | Email id |
| Place | Varchar | 40 | NOT NULL | Place |

**Table 4**: lab

**Primary Key:** lid

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **CONSTRAINTS** | **DESCRIPTION** |
| Lid | int | 11 | PRIMARY KEY | Lab assistant is |
| Name | Varchar | 30 | NOT NULL | Name |
| Qlftn | varchar | 20 | NOT NULL | Qualification |
| Addr | Varchar | 50 | NOT NULL | Address |
| Phno | Varchar | 12 | NOT NULL | Phone number |
| Email | Varchar | 30 | NOT NULL | Email |
| Gen | Varchar | 10 | NOT NULL | Gender |
| Exp | Varchar | 10 | NOT NULL | Experience |

**Table 5**: dept

**Primary Key:** dept id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **CONSTRAINTS** | **DESCRIPTION** |
| Deptid | int | 11 | PRIMARY KEY | Department id |
| Deptn | varchar | 50 | NOT NULL | Department |

**Table 6**: appointment

**Primary Key:** apid

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **CONSTRAINTS** | **DESCRIPTION** |
| Apid | int | 10 | PRIMARY KEY | Appointment id |
| Vid | int | 10 | NOT NULL | User id |
| Pname | varchar | 30 | NOT NULL | Patient name |
| Age | int | 4 | NOT NULL | Age |
| Gender | Varchar | 10 | NOT NULL | Gender |
| Doctrname | Varchar | 30 | NOT NULL | Doctor name |
| Time | Varchar | 10 | NOT NULL | Time |
| Phno | Varchar | 12 | NOT NULL | Phone number |
| Date | Date | 10 | NOT NULL | Date |
| Fee | Int | 10 | NOT NULL | Fees |
| Docid | Int | 10 | NOT NULL | Doctor id |
| Status | Varchar | 30 | NOT NULL | Status |
| Da | Varchar | 30 | NOT NULL | Day |

**Table 7**: ambulance

**Primary Key:** aid

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **CONSTRAINTS** | **DESCRIPTION** |
| Aid | int | 11 | PRIMARY KEY | Ambulance id |
| Namb | Int | 10 | NOT NULL | No of ambulance |
| Dname | Varchar | 50 | NOT NULL | Driver name |
| Facility | Varchar | 50 | NOT NULL | Facility |

**Table 8:** prescription

**Primary Key:** idp

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **CONSTRAINTS** | **DESCRIPTION** |
| Idp | int | 11 | PRIMARY KEY | Prescription id |
| Apid | Int | 10 | NOT NULL | Appointment id |
| Pname | Varchar | 50 | NOT NULL | Patient name |
| Prescription | Varchar | 30 | NOT NULL | Prescription |
| Docid | int | 11 | NOT NULL | Doctor id |
| Pid | int | 11 | NOT NULL | Patient id |
| Date | Varchar | 10 | NOT NULL | Date |
| Ltest | Varchar | 50 | NOT NULL | Lab test |

**Table 9**: labrep

**Primary Key:** tid

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **CONSTRAINTS** | **DESCRIPT`ION** |
| Tid | int | 11 | PRIMARY KEY | Test id |
| Pid | Int | 11 | NOT NULL | Patient id |
| Pname | Varchar | 30 | NOT NULL | Patient name |
| Docid | Varchar | 10 | NOT NULL | Doctor id |
| Tname | Varchar | 30 | NOT NULL | Test name |
| Result | Varchar | 30 | NOT NULL | Result |
| Resultfile | Varchar | 50 | NOT NULL | Result file |

**Table 10**: payment

**Primary Key:** id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **CONSTRAINTS** | **DESCRIPTION** |
| Id | int | 11 | PRIMARY KEY | Payment id |
| Chn | varchar | 50 | NOT NULL | Card holder name |
| Cvv | varchar | 10 | NOT NULL | C v v number |
| Cardno | Varchar | 20 | NOT NULL | Card number |
| Date | Date | 10 | NOT NULL | Date |
| Amount | Int | 5 | NOT NULL | Amount |
| Vid | Int | 10 | NOT NULL | User id |
| Pstatus | Varchar | 30 | NOT NULL | Payment status |

**Table 11**: feedback

**Primary Key:** fid

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **CONSTRAINTS** | **DESCRIPTION** |
| Fid | int | 11 | PRIMARY KEY | Feedback id |
| Vid | Int | 10 | NOT NULL | User id |
| Name | Varchar | 30 | NOT NULL | Name |
| Fed | Varchar | 30 | NOT NULL | Feedback |

**CODING**

**6. CODING**

Coding refers to the process of translating representations of the software into a form that can be understood by the computer. The coding process transforms detail design into a programming language. Programming language characteristics and coding style can profoundly affect quality and maintainability.

**6.1 SELECTION OF PROGRAMMING LANGUAGE**

A general set of engineering characteristics for a programming language can be established. They include ease of translation from design to code, computer efficiency, source code portability, availability of development tools and maintainability.

The art of choosing a language is to start with the problem, decide what it requires and their relative importance since it will probably be impossible to satisfy all of them equally. Available language should be measured against a lot of requirements. The criteria that are applied during an evaluation of available language are:

* General application Area
* Algorithmic and Computational Capability
* Environment in which Software will execute
* Cost effectiveness
* Ease of Use
* Open Source

All the above requirements are considered while choosing the programming language for Virtual University. It is developed using tools such as .net The Powerful database MYSQL is used as the back end. Both come under the category of open sources software’s development tools.

**6.2 STANDARD CODING PRACTICES**

Coding Practice (or style) encompasses a coding philosophy that stresses simplicity and clarity. The elements of style include internal documentation that is source code level, methods for data declaration, an approach to statement construction and techniques for input and output. Internal documentation of source begins with the visual organization selection of identifier names continues with placement and composition of commenting and concludes with the visual organization of the program. The selection of meaning full identifier names and order of data declaration are crucial understanding comments and code during maintenance.

Some Remarkable Conventions\ Styles followed are listed below

* Variable & methods and classes are named with standard naming conventions
* All the controls placed in the webpage are given meaningful names
* Exception handling is done to avoid runtime errors
* Object Oriented Style is used in coding to fully utilize the OO features.
* Access specifies are used appropriately in classes to restrict access to attributes

**6.3 CODE VALIDATION AND OPTIMIZATIONS**

Efficiency of the proposed system depends on how many rules are applied to validate the code, the way in which the rules are formulated and codified and whether the soundness of the rules are checked in all possible execution paths of the program considered .Unlike in code optimization, validation is done based on rules which restrict the analysis to perform as a qualifier, indicating whether the rule tested is passed or failed which leads to error detection. A code validation and optimization technique assisting the embedded system software debugging to make it more effective at revealing errors and redundancy is proposed. Since the method adopts a static analysis, the tool developed has the merits and demerits of static analysis. Since the analysis is done on machine code this work has got the advantages and disadvantages of machine code analysis.

**6.4 CODING**

**DatabaseCon.php**

**<?php**

**class DatabaseCon {**

**private $dbhost = 'localhost:3306';**

**private $dbuser = 'root';**

**private $dbpass = '';**

**private $conn;**

**function \_\_construct()**

**{**

**// echo "inside constructor";**

**$this->conn = mysql\_connect($this->dbhost, $this->dbuser, $this->dbpass);**

**if(!$this->conn )**

**{**

**die('Could not connect: ' . mysql\_error());**

**}**

**//echo 'Connected successfully';**

**mysql\_select\_db( 'hcm' );**

**}**

**function insertQuery($qry) {**

**echo $qry;**

**$retval = mysql\_query( $qry, $this->conn );**

**if(!$retval )**

**{**

**die('Could not enter data: ' . mysql\_error());**

**}**

**//echo "Entered data successfully\n";**

**}**

**function selectQuery($str) {**

**// echo $str;**

**$f=0;**

**$retval=mysql\_query($str,$this->conn);**

**$num\_rows= mysql\_num\_rows(mysql\_query($str));**

**//echo $num\_rows;**

**//if(!$retval)**

**if($num\_rows>0)**

**{**

**$f=1;**

**}**

**return $f;**

**}**

**function selectData($str)**

**{**

**$retval=mysql\_query($str,$this->conn);**

**return $retval;**

**}**

**function updateQuery($str)**

**{**

**$retval=mysql\_query($str,$this->conn);**

**if(!$retval)**

**{**

**die('Could not update data!'.mysql\_error());**

**}**

**}**

**function closeDatabase()**

**{**

**mysql\_close($this->conn);**

**}**

**}**

**?>**

**TESTING**

**7. TESTING**

The different levels of testing done on the system are: -

* Unit Testing
* Integration Testing
* Final/ System testing
* Validation Testing
* Output Testing
* User Acceptance Testing
* Black box testing
* White box testing

**7.1 UNIT TESTING**

Here each module is test individually and integrate the overall system. Unit testing focuses verification efforts even in the smallest unit of software design in each module. This is known as “Module Testing”. The modules of the system are tested separately. This testing is carried out in the programming style itself. In this testing each module is focused to work satisfactorily as regard to expected output from the module. There are some validation checks for the fields.

**7.2 INTEGRATION TESTING**

Data can be lost across an interface, one module can have an adverse effect on the other sub-functions, when combined may not produce the desired functions. Integrated testing is the systematic testing to uncover the errors within the interface. This testing is done with simple data and the developed system has run successfully with this simple data. The need for integrated system is to find the overall system performance.

**7.3 SYSTEM TESTING**

When a system is developed, it is hoped that it performs properly. In practice, however, some errors always occur. The main purpose of testing an information system is to find the errors and correct them. A successful test is one, which finds an error.

The main objectives of system testing are

* To ensure during operation the system will perform as per specification.
* To make sure that the system meets user’s requirements during operation.
* To verify that the controls incorporated in the system function as intended.
* To see that when correct inputs are fed to the system the outputs are correct.
* To make sure that during operation, incorrect input and output will be deleted.

The scope of a system test should include both manual operations and computerized.

**7.4 VALIDATION TESTING**

At the culmination of black box testing, software is completely assembled as a package. Interfacing errors have been uncovered and correct and final series of test, i.e., validation test begins. Validation test van is defined with a simple definition that succeeds when the software functions in a manner that can be reasonably accepted by the customer.

**7.5 OUTPUT TESTING**

After performing validation testing, the next step is output testing of the proposed system. Since the system cannot be useful if it does not produce the required output. Asking the user about the format in which the system is required tests the output displayed or generated by the system under consideration. Here the output format is considered in two ways. One is on screen format and other one is printed format. The output format on the screen is found to be corrected as the format was designed in the system phase according to the user needs. As for hard copy the output comes according to the specification requested by the user. Here the output testing does not result in any correction in the system.

Taking various kinds of data plays a vital role in system testing. After preparing the test data, system under study is tested using the tested data. While testing, errors are again uncovered and corrected by using the above steps and corrections are also noted for future use. The system has been verified and validated by running test data and live data. First the system is tested with some sample test data are generated with the knowledge of possible range of values that are required to hold by the fields. The system runs successfully for the given test data and for live data

**7.6 USER ACCEPTANCE TESTING**

User acceptance testing of the system is the key factor for the success of any system. The system under consideration is tested for the user acceptance by constantly keeping in touch with perspective system at the time of development and making change whenever required. This is done with regard to the input screen design and output screen design.

**7.7 BLACK BOX TESTING**

Knowing the specific function that a product has been designed to perform, test can be conducted that each function is fully operational. Black Box Testing is carried to test that input to a function is probably accepted and output is correctly produced. A black box testing examines some aspects of a system with little regards for the internal logical structure of the software. Errors in the following categories were found through black box testing

* Incorrect or missing function
* Interface errors
* Errors in data structures or external database access
* Performance errors
* Initialization and termination errors

**7.8 WHITE BOX TESTING**

White Box Testing of software is predicated on a close examination of procedural details. The status of a program may be tested at various points. Things to determine whether asserted status is corresponding to the actual status. Using the following test case can be derived.

* Exercise all logical conditions on their true and false side
* Exercise all loops within their boundaries and their operational bounds

**SYSTEM**

**IMPLEMENTATION**

**AND MAINTENANCE**

**8. SYSTEM IMPLEMENTATION AND**

**MAINTENANCE**

Implementation is the stage of the project when the theoretical design is turned into a working system. The implementation stage is a systems project in its own right. It includes careful planning, investigation of current system and its constraints on implementation, design of methods to achieve the changeover, training of the staff in the changeover procedure and evaluation of changeover method.

The first task in implementation is planning- deciding on the methods and time-scale to be adopted. Once the planning has been completed, the major effort is to ensure that the programs in the system are working properly. At the same time concentrate on training the staff. When the staffs have been trained, the complete system, involving both computer and user can be executed effectively.

When the Manager’s system is linked to terminals on remote sites, the telecommunication network and tests of the network along with the system are also included under implementation. Depending upon the nature of the system, extensive user training may be required. Programming itself is a design work. The initial parameters of the management information system should be modified as a result of programming efforts; programming provides a Reality test for the assumptions made by the analyst.

System testing check the readiness and accuracy of the system access update and retrieve data from new files. Once the program becomes available, the test data are read into the computer and processed. In this system, conventional Parallel Run was conducted to establish the efficiency of the system.

Implementation is used here to mean the process of converting a new or a revised system design into an operational one. Conversion is one aspect of Implementation. Conversion means changing from one system to another. The objective is to put the tested system into operation while holding costs, risks and personal irritation to a minimum.

Changeover is the process of adopting the new system. The new system has to be introduced however. This is done after the system has been developed and tested completely. There is a set of methods like Direct Changeover, Parallel Changeover, Pilot running etc. Pilot running is intended here.

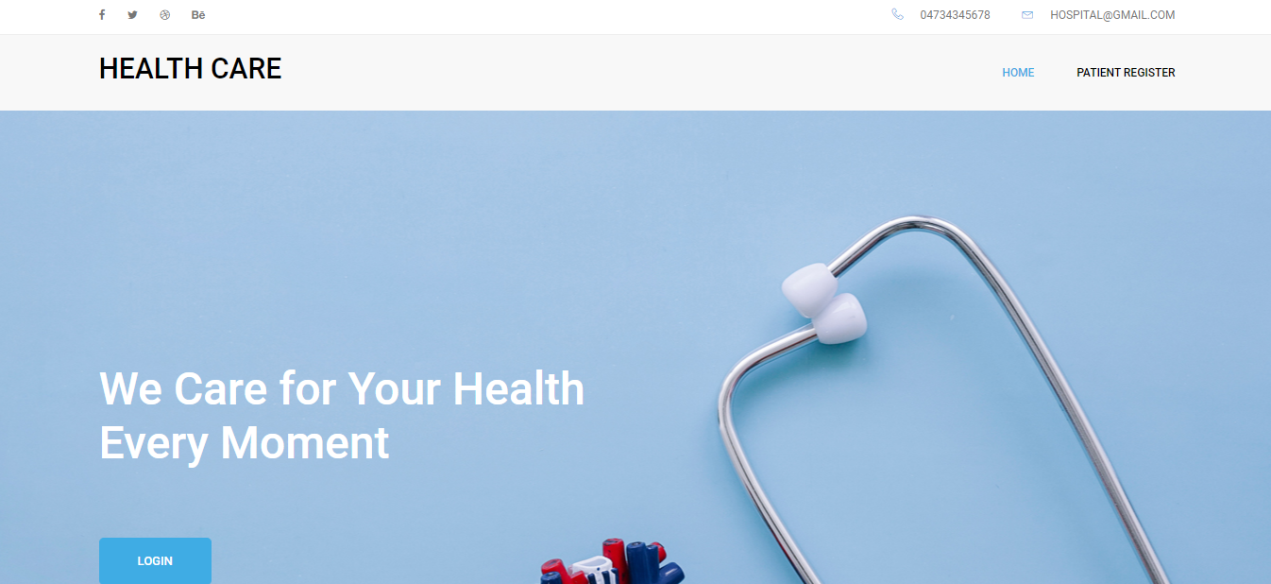
Data from one or more previous periods for the whole or part of the system is run on the new system after results have been obtained from the old system and both are compared. It is performed till the completion of one system life cycle.

When the changeover has taken place there will be a need for amendment to correct or improve the new system. When the user wants to add any new records, some fields will automatically get their default values. If the user desires to change these default values he can do it.

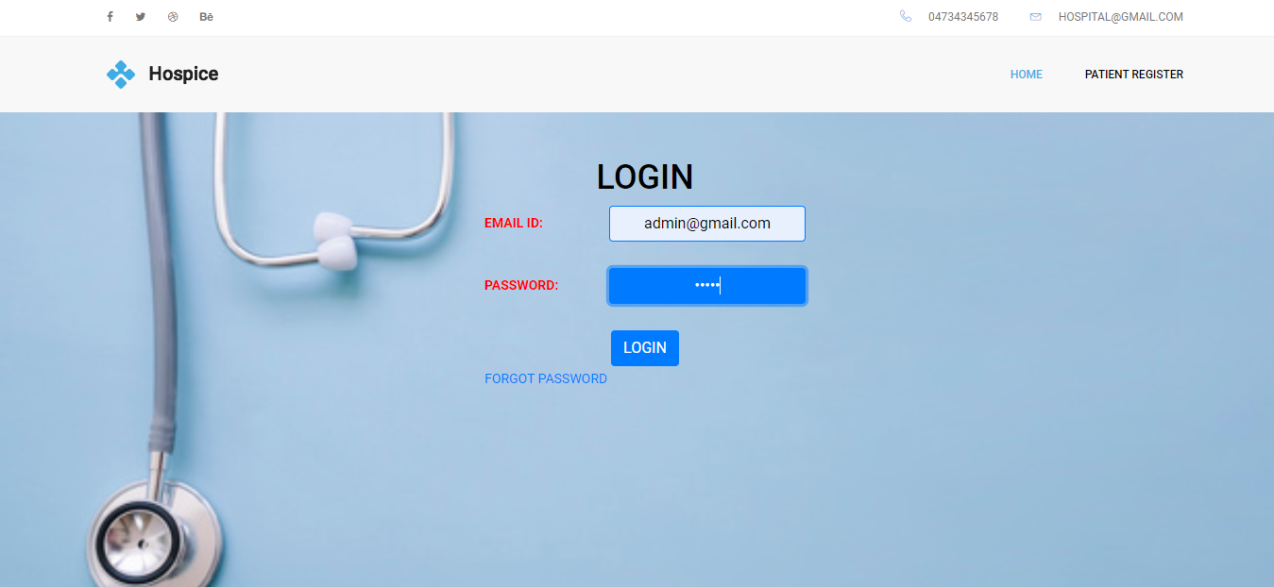
**SCREENSHOT**

**9. SCREENSHOT**

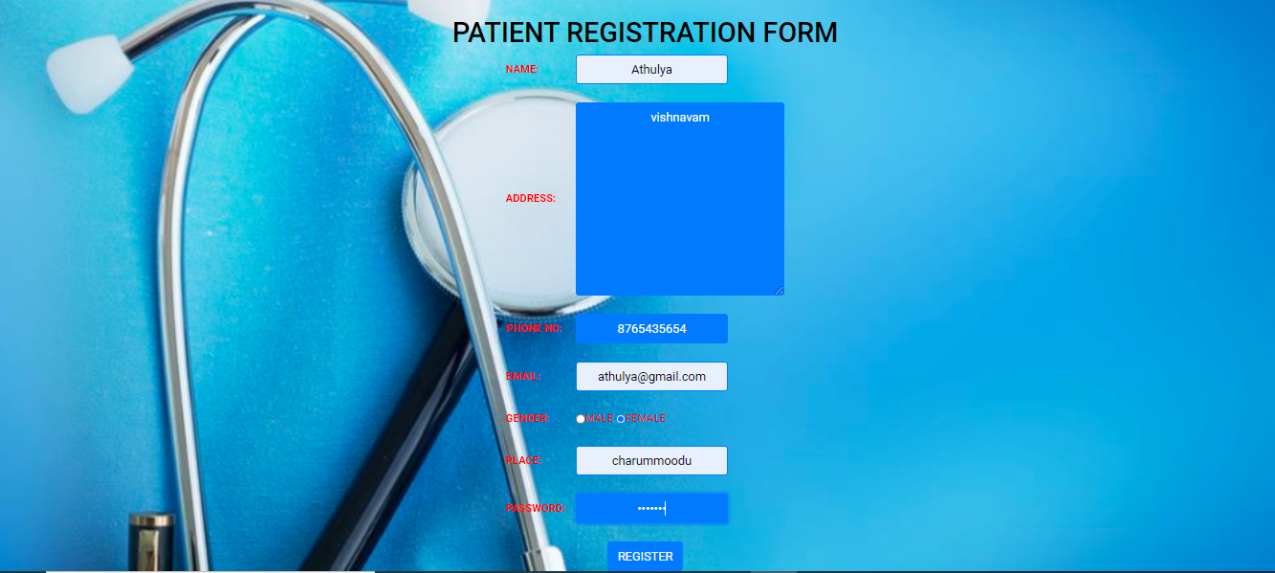
**HOME PAGE**

****

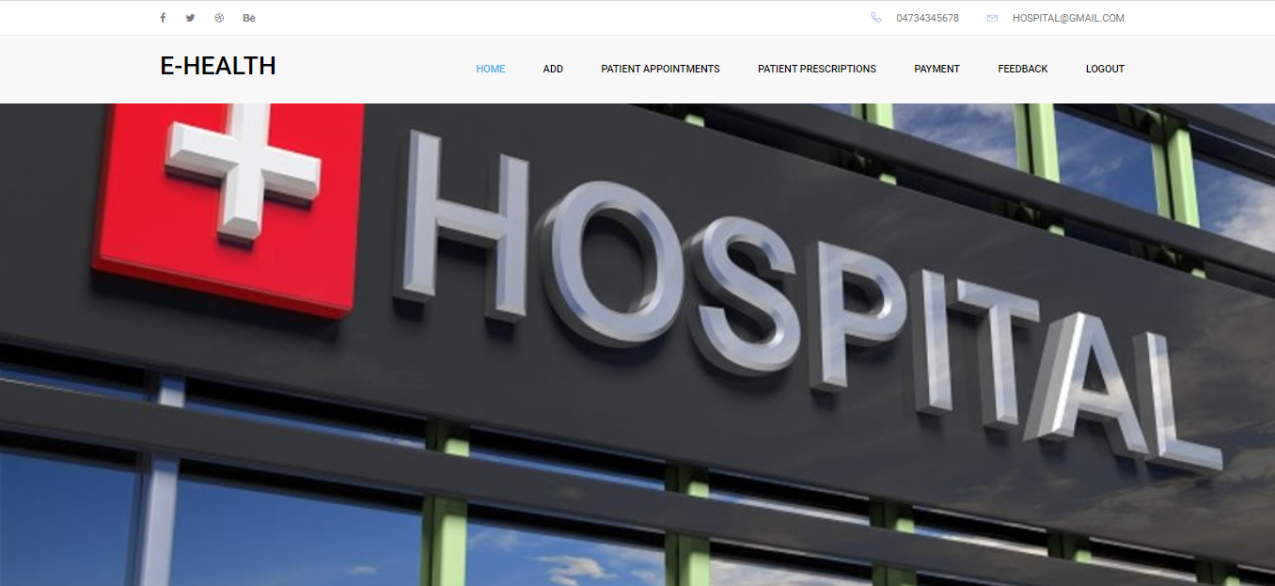
**LOGIN**

****

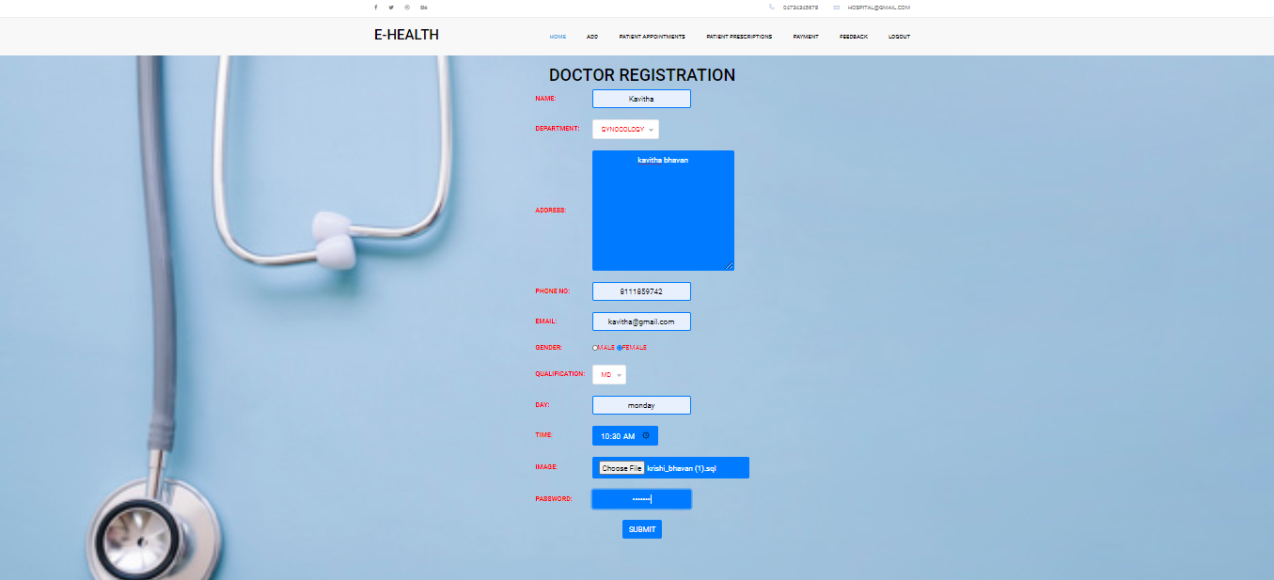
**PATIENT REGSITARTION**

****

**ADMIN HOME**

****

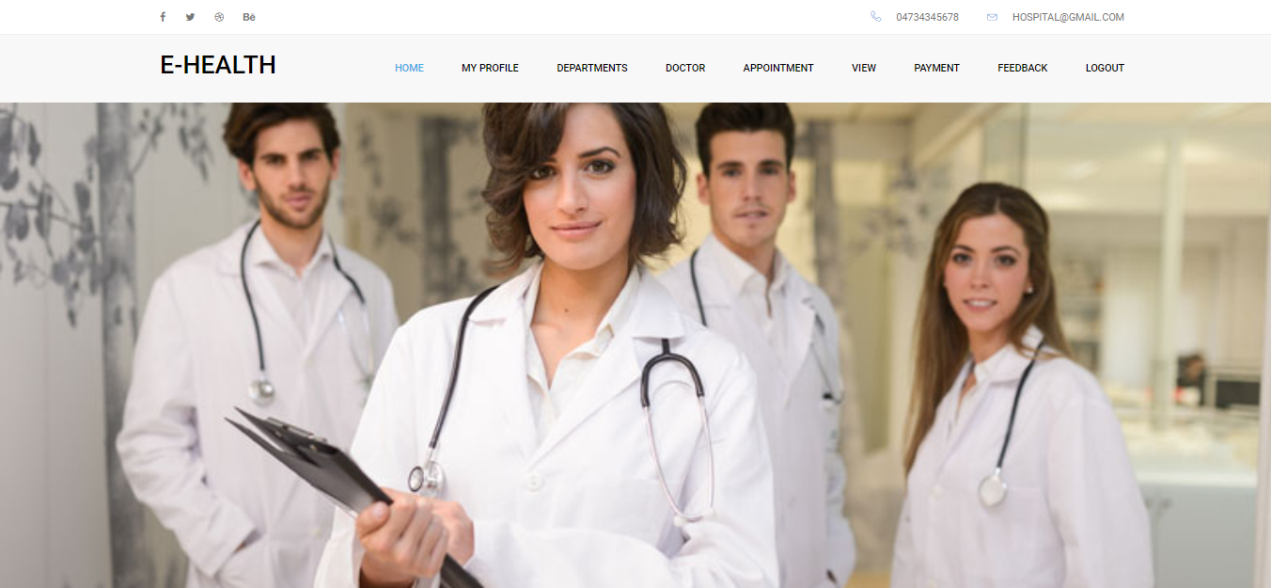
**DOCTOR**

****

**DEPARTMENT**

****

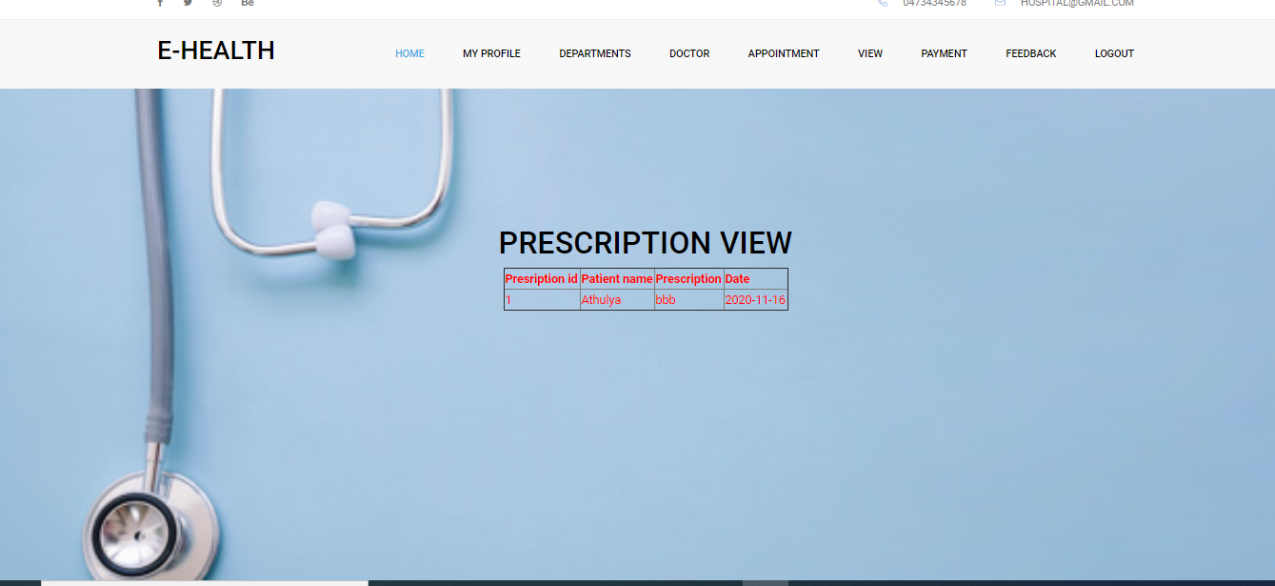
**PATIENT HOME PAGE**

****

**APPOINTMENT**

****

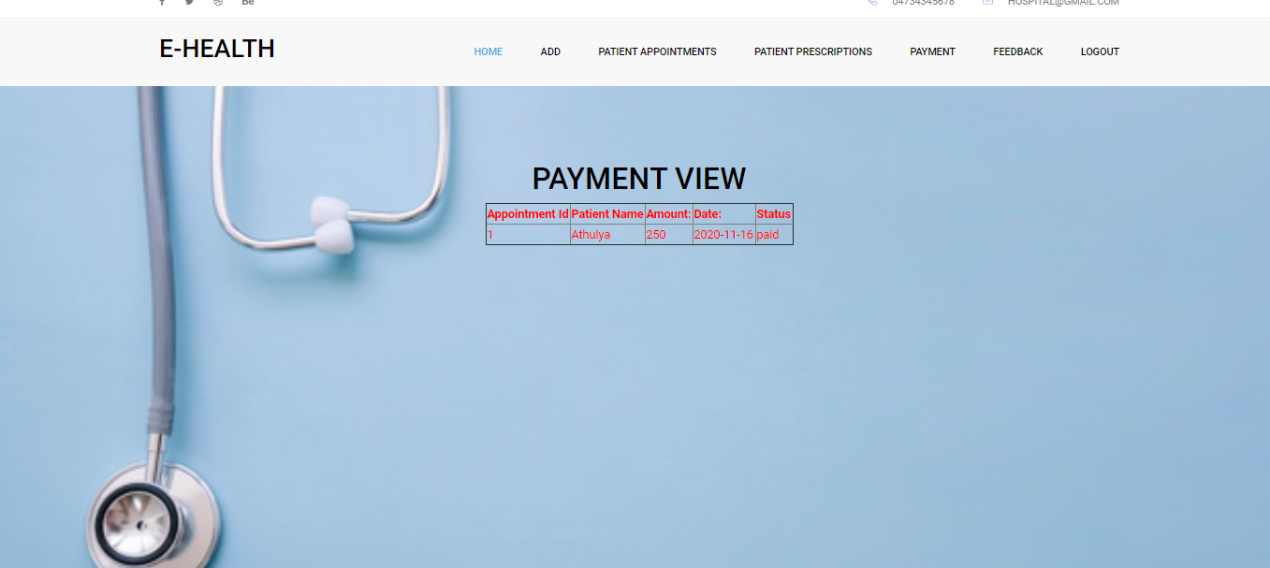
**VIEW PRESCRIPTION**

****

**VIEW PATIENT**

****

**VIEW PAYMENT**

****

**FUTURE**

**ENHANCEMENT**

**10. FUTURE ENHANCEMENT**

There are various possibilities for enhancing this project. Add the link of various different publishers for getting more feeds. This simply requires the re-use of certain code modules which were used earlier. We have to simply specify the URL of the new publisher’s feed location at the appropriate place. The output design should also be changed. Bigger updates can be implemented. Each user can have his own space, where the data they like can be stored for future viewing. According to logic of a person so many modifications can be done, it depends on he who doing it.

**CONCLUSION**

**11. CONCLUSION**

System based application titled “**HEALTH CARE**” that is easily accessible, informative and helpful. It has been designed in such a way that it is easy to modify, can be updated efficiently and accurately. The forms are designed user friendly by providing messages and captions whenever necessary, so that user has no problem to overcome difficulties in data entry, validation, searching etc. Administrator could refer the data stored in this software for further reference. This system is more efficient. Easily register the birth and death certificates. Check complaints status in online.

**BIBLIOGRAPHY**

**12. BIBLIOGRAPHY**

1. Software Engineering: Fundamentals of Software Engineering Fourth Edited for a mall (2014 edition)..
2. PHP Mysql website programming: Problem-Design-Solution by Chris Lea, Mike Buzzard,DilipThomas,Jessy White-Cinis
3. Mysql/PHP Database Application,2nd Edition by Brand Bulger
4. Elias. M. Awad, “System*Analysis and* Design”, Galgoeia publications – Second Edition2005.
5. Henry Korth,” Data*Base Design Concept*”, Mc grew hill –Fifth Edition 2001. Roger.S.Pressman,” Software*Engineering* ”, Mc grew hill Fifth Edition 2006.
6. www.stackoverflow.com
7. www.vbforums.com
8. www.xtemevbtalk.com