ABSTRACT

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

A significant part of the operation of any hospital involves the acquisition, management and timely retrieval of great volumes of information. This information typically involves; patient personal information and medical history, staff information, room and ward scheduling, staff scheduling, operating theater scheduling and various facilities waiting lists. All of this information must be managed in an efficient and cost wise fashion so that an institution's resources may be effectively utilized Hospitality System will automate the management of the hospital making it more efficient and error free. It aims at standardizing data, consolidating data ensuring data integrity and reducing inconsistencies.

Hospitality System provides the benefits of streamlined operations, enhanced administration & control, superior patient care, strict cost control and improved profitability. Hospitality System is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to hospitals. More importantly it is backed by reliable and dependable support.

The project ‘Hospitality System’ is based on the database, object oriented techniques. As there are many areas where we keep the records in database for which we are using MY SQL software which is one of the best and the easiest software to keep our information. This project uses PHP, HTML as the front-end Language which is an Object Oriented Programming and has connectivity with MY SQL.

Hospitality System is custom built to meet the specific requirement of the mid and large size hospitals across the globe. All the required modules and features have been particularly built to just fit in to your requirement. This package has been widely accepted by the clients in India and overseas. Not stopping only to this but they are highly satisfied and appreciating. Entire application is web based and built on 3 tier architecture using the latest technologies. The sound database of the application makes it more users friendly and expandable. The package is highly customizable and can be modified as per the needs and requirements of our clients. Prolonged study of the functionalities of the hospital and its specific requirement has given it a wonderful shape both technically and usability wise. It covers all the required modules right from Patient Registration, Doctor, Admin, Patient appointment, bill payment, record modification, discharge details etc. A unique feature that we have added is the SMS system. Whenever the patient books an appointment he/she gets a SMS regarding confirmation of the appointment and the same SMS is sent to the Doctor as well, telling him/her the patient name and age.

CHAPTER – 1

INTRODUCTION

Our project Hospitality System includes registration of patients, storing their details into the system and also computerized billing in the pharmacy, and labs. Our software has the facility to give a unique id for every patient and stores the details of every patient and the staff automatically. It includes a search facility to know the current status of each room. User can search availability of a doctor and the details of a patient using the id. The Hospitality System can be entered using a username and password. It is accessible by administrator only. Only he can add data into the database. The data can be retrieved easily. The interface is very user-friendly. The data are well protected for personal use and makes the data processing very fast.

The main aim of this project is to provide essential medical services online to everyone hardly matters we live in metro or a remotely located village. Users can connect through their home internet or approach any nearby kiosk to get these services. What motivate to build this system are:

1) Very few or no doctors at remote locations
2) Limited hour services and lack of sophisticated medical equipments
3) No patients history/lab data management

The Software is for the automation of Hospital Management. It maintains two level of user:-

- Administrator Level
- User Level

The Software includes:-

- Maintaining Patient and Doctor Details.
- Providing and maintaining all kinds of test for a patient
- providing prescription and precautions
- Billing and Report generation

(i) FUNCTIONALITY FOR THE VISITORS:

1. The visitors of the website are provided the facility to register themselves to the website.
2. Visitors can view our About Us webpage.
3. Visitors can contact us through telephone no.

.

(ii) FUNCTIONALITY FOR THE REGISTERED USERS:

1. New User (Patient/Doctor) who is not registered has to get registered them by filling entries of the registration form with unique user-id.
2. Patient can take appointment from Doctor.
3. Patient can fill feedback about treatment.
4. Patient can fill medicine error form in case of wrong medicine or any side effects by taking medicine.
5. Patient can mail to doctor.
6. Patient can change password.
7. Doctor can give appointment to patient.
8. Doctor can view patient history.
9. Doctor can prescribe medicine to patient.
10. Doctor can change password.

(iii) FUNCTIONALITY FOR THE ADMINISTRATOR:

1. Administrator is assigned to the website at the time of deployment of the site, with user Id and password; he is the overall manager of the site.
2. Administrator can view doctor details.
3. Administrator can view patient feedback.
4. Administrator can view medicine error.
5. Administrator can delete Patient/Doctor.

CHAPTER - 2

PROBLEM IDENTIFICATION

Problems with conventional system:

(i) Lack of immediate retrievals: The information is very difficult to retrieve and to find particular information like- E.g. - To find out about the patient's history, the user has to go through various registers. This results in inconvenience and wastage of time.

(ii) Lack of immediate information storage: The information generated by various transactions takes time and efforts to be stored at right place.

(iii) Lack of prompt updating: Various changes to information like patient details or immunization details of child are difficult to make as paper work is involved.

(iv) Error prone manual calculation: Manual calculations are error prone and take a lot of time this may result in incorrect information. For example: calculation of patient's bill based on various treatments.

(v) Preparation of accurate and prompt reports: This becomes a difficult task as information is difficult to collect from various registers.

Since HOSPITAL is associated with the lives of common people and their day-to-day routines so I decided to work on this project. The manual handling of the record is time consuming and highly prone to error. The purpose of this project is to automate or make online, the process of day-to-day activities like Room activities, Admission of New Patient, Discharge of Patient, Assign a Doctor, and finally compute the bill etc.

I have tried my best to make the complicated process **Hospitality System** as simple as possible using Structured & Modular technique & Menu oriented interface. I have tried to design the software in such a way that user may not have any difficulty in using this package & further expansion is possible without much effort.

Even though I cannot claim that this work to be entirely exhaustive, the main purpose of my exercise is perform each Hospital's activity in computerized way rather than manually which is time consuming. I am confident that this software package can be readily used by non-programming personal avoiding human handled chance of error.

CHAPTER – 3

SOFTWARE MODEL USED

The software model that we used for developing this project is the Waterfall model. The Waterfall Model was first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases. Waterfall model is the earliest SDLC approach that was used for software development. The waterfall Model illustrates the software development process in a linear sequential flow; hence it is also referred to as a linear-sequential life cycle model. This means that any phase in the development process begins only if the previous phase is complete. In waterfall model phases do not overlap.

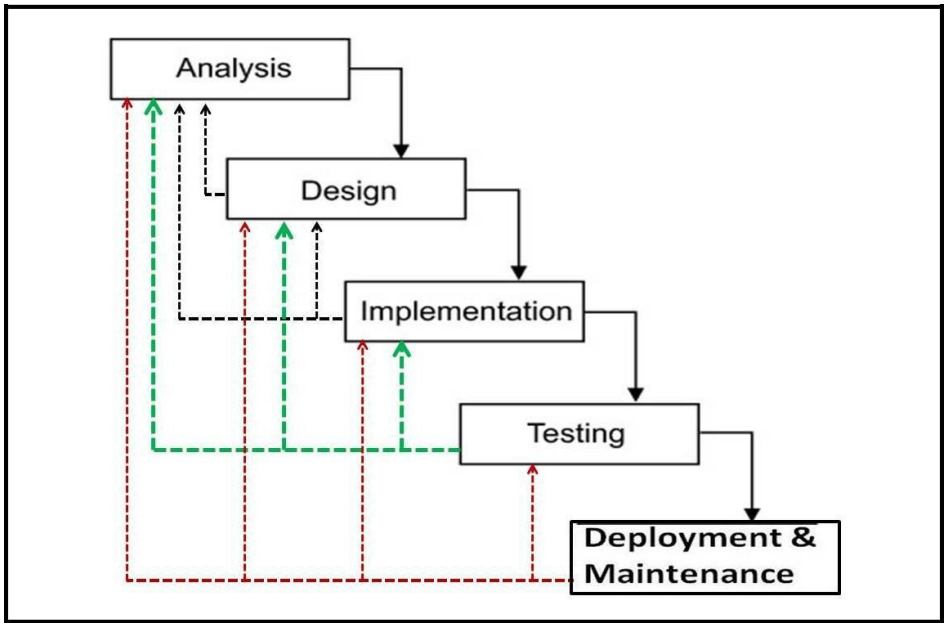


Fig. - 1: Illustration of Waterfall Model of SDLC.

(i) ANALYSIS: All possible requirements of the system to be developed are stated in this phase. Requirements are a set of functions and constraints that the end user expects from the system. The requirements are gathered from the end user, and are analyzed for their validity and the possibility of incorporating them. Finally, a requirement specification document is created which serves the purpose of guideline for next phase of the model.

(ii) DESIGN: Before starting the actual coding phase, it is highly important to understand the requirements of the end user and also have an idea of how the end product should look like. The requirement specifications from the first phase are studied here, and a system design is prepared. The design helps in specifying hardware and system requirements, and also helps in defining the overall system architecture. The system design specifications serve as an input for the next phase of the model.

(iii)IMPLEMENTATION: On receiving system design documents, the work is divided in modules/units and actual coding starts. The system is first elaborated into small programs called units, which are integrated in the next phase.

(iv) TESTING: The units are now integrated to form a complete system during the integration phase and tested to check if all modules/units coordinate with each other and the system as a whole behaves as per the specifications. After successfully testing the software, it is delivered to the customers.

(v) DEPLOYMENT AND MAINTAINENCE: This phase of the model is virtually never-ending. Generally, problems with the system (which are not found during the development cycle) come up after its practical use starts, so the issues related to the system are solved after its deployment. Not all the problems come into picture directly, but they arise from time to time and need to be solved; hence this process is referred to as maintenance, even though it is still pretty much in the testing phase.

The waterfall model has the following advantages for building this project:

- There is clear compartmentalization of work and control. That makes it easier to set a schedule for the tasks to be completed within a specified period.
- Only after the work for a particular phase is over, does the other phase start, due to which there is no overlapping of phases, or the product does not have to go through different iterative steps.
- This model is much easier to envision. Since the processes are all carried out in a linear trickle-down manner, the cost of resources is reduced to a large extent, which in turn helps in reducing the cost of the project considerably.

CHAPTER – 4

IMPLEMENTATION

This project will be a desktop application to be developed in PHP having MYSQL as backend.

- Database Design (MYSQL)
- Form Design (PHP)
- Coding (PHP)
- Testing (PHP)

Project is related to Hospital Management System. The project maintains two levels of users:-

- Administrator Level-Doctor
- User Level-Data Entry Operator

Main facilities available in this project are:-

- Maintaining records of indoor/outdoor patients.
- Maintaining patients diagnosis details, advised tests to be done.
- Providing different test facilities to a doctor for diagnosis of patients.
    - Gastroenterology, Oncology, Radiotherapy and Hematology
    - Nephrology
    - Physiotherapy
    - Dermatology
    - Pulmonologist
    - Endocrinology
    - General Physician
    - Neurology

- Ophthalmology
- Vascular and Endovascular
- Pediatric
- ENT
- Urology and Transplant

- Maintaining records of indoor/outdoor patients.
- Maintaining patients diagnosis details, advised tests to be done.
- Providing different test facilities to a doctor for diagnosis of
- Maintaining patient's injection entry records.
- Maintaining patient's prescription, medicine and diet advice details.
- Providing billing details for indoor/outdoor patients.
- Maintaining backup of data as per user requirements (between mentioned dates).
- Related test reports, patient details report, prescription and billing reports can be generated as per user requirements.
- User or Administrator can search a patient's record by his/her name or their registration date.

4.1 SYSTEM IMPLEMENTATION MAINTENANCE AND REVIEW:

As we know, creating software is one thing and the implementation of the created software is another. The process of implementing software is much difficult as compared to the task of creating the project. First we have to implement the software on a small scale for removing the bugs and other errors in the project and after removing them we can implement the software on a large scale. Before we think in terms of implementing the Software on a large basis, we must consider the Hardware requirements.

Whenever we develop software or project a certain hardware and software is being used by the programmer for developing the project.

The hardware and software to be used by the programmer for developing the project should be such that it would result in the development of a project, which would satisfy all the basic needs for which the project has been created by the programmer. The Hardware should be such that cost constraints of the Client should also be taken into account without affecting the performance.

4.2 HARDWARE EVALUATION FACTORS:

When we evaluate computer hardware, we should first investigate specific *physical and performance* characteristics for each hardware component to be acquired. These specific questions must be answered concerning many important factors. These *hardware evaluation factors* questions are summarized in the below figure.

Notice that there is much more to evaluating hardware than determining the fastest and cheapest computing device. For e.g. the question of possible obsolescence must be addressed by making a technology evaluation. The factor of *ergonomics* is also very important. Ergonomics is the science and technology that tries to ensure that computer and other technologies are "user-friendly", that is safe, comfortable and easy to use. *Connectivity is* another important evaluation factor, since so many computer systems are now interconnected within wide area or local area telecommunications networks.

Hardware evaluation factors are as follows:

1) Performance

2) Cost

3) Reliability

4) Availability

5) Compatibility

6) Modularity

7) Technology

8) Connectivity

9) Environmental requirements

10) Software

11) Support

4.3 SOFTWARE EVALUATION FACTORS:

Software can be evaluated according to many factors similar to the hardware evaluation. Thus the factors of *performance, cost, reliability, compatibility, modularity, technology, ergonomics, and support* should be used to evaluate proposed software acquisitions. In addition, however, *the software evaluation factors* are summarized in below figure. For e.g. some software packages require too much memory capacity and are notoriously slow, hard to use, or poorly documented. They are not a good selection for most end users, even if offered at attractive prices.

1) Efficiency: Is the software a well-written system of computer instructions that does not use much memory capacity or CPU time?

2) Flexibility: Can it handle its processing assignments easily without major modifications?

3) Security: Does it provide control procedures for errors, malfunctions and improper use?

4) Language: Do our computer programmers and users write it in a programming language that is used?

5) Documentation: Is the s/w well documented? Does it include helpful user instructions?

6) Hardware: Does existing hardware have the features required to best use this software?

7) Other characteristics of hardware such as its performance, what about the cost, how much is reliable and etc.

4.4 USE CASE DIAGRAM

A use case diagram is a graphic depiction of the interactions among the elements of a system. In this project we have administrator who takes appropriate measures. A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

As depicted in the use case diagram below, the actor is the user (administrator), who can perform various actions like the various updates required, management of backend services. The user may visit the contact us page and may get the required details. The user may then communicate for the details of a particular doctor and can take appointment. The user and the doctor will then receive a confirmation message of appointment on their mobile phones.

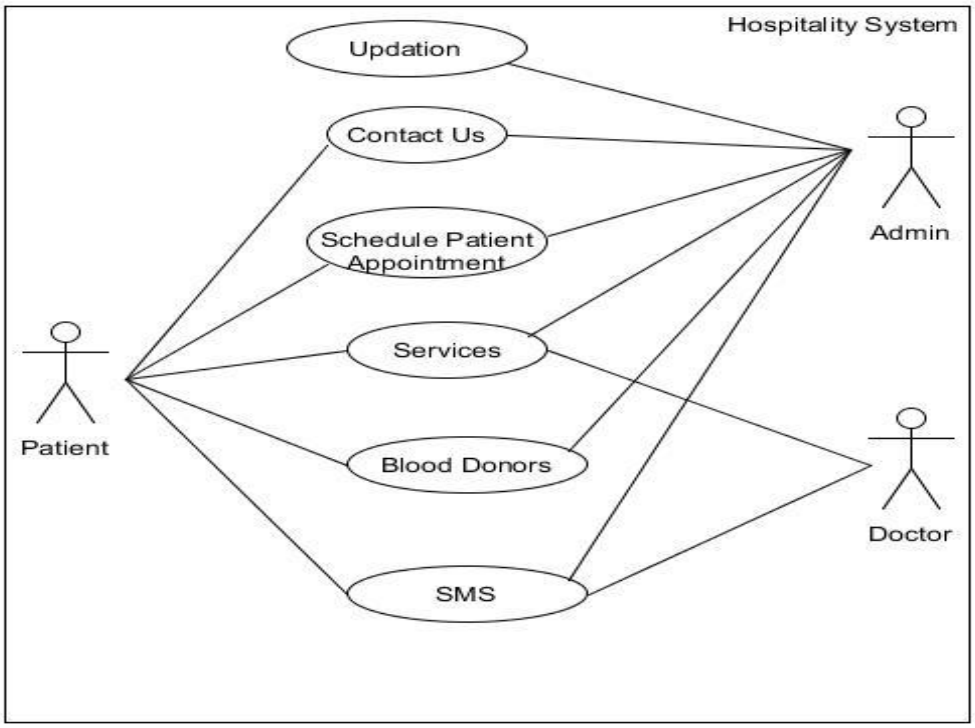


Fig. -2: Use Case Diagram

4.5 SEQUENCE DIAGRAM

A Sequence diagram is an interaction diagram that shows how processes operate with one another and what is their order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. Here in our project we intend to take our first step as the interaction between patient and the receptionist. The receptionist checks the doctor's availability and then confirms the appointment. Then, the confirmation message is sent to the respective doctor as well as the patient that gives a record of their appointment details. The patient receives the consultation from the doctor and then pays the consultation fees to the receptionist.

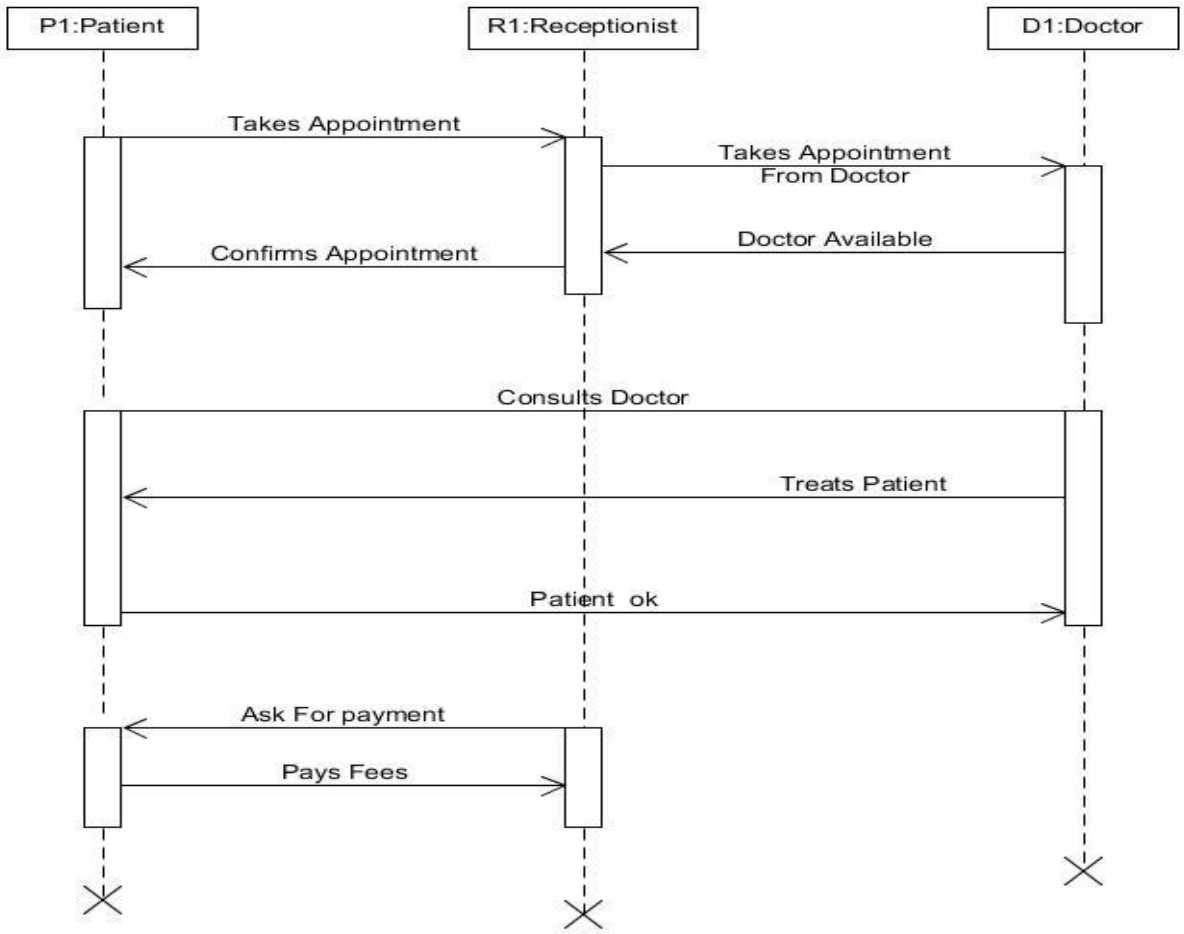
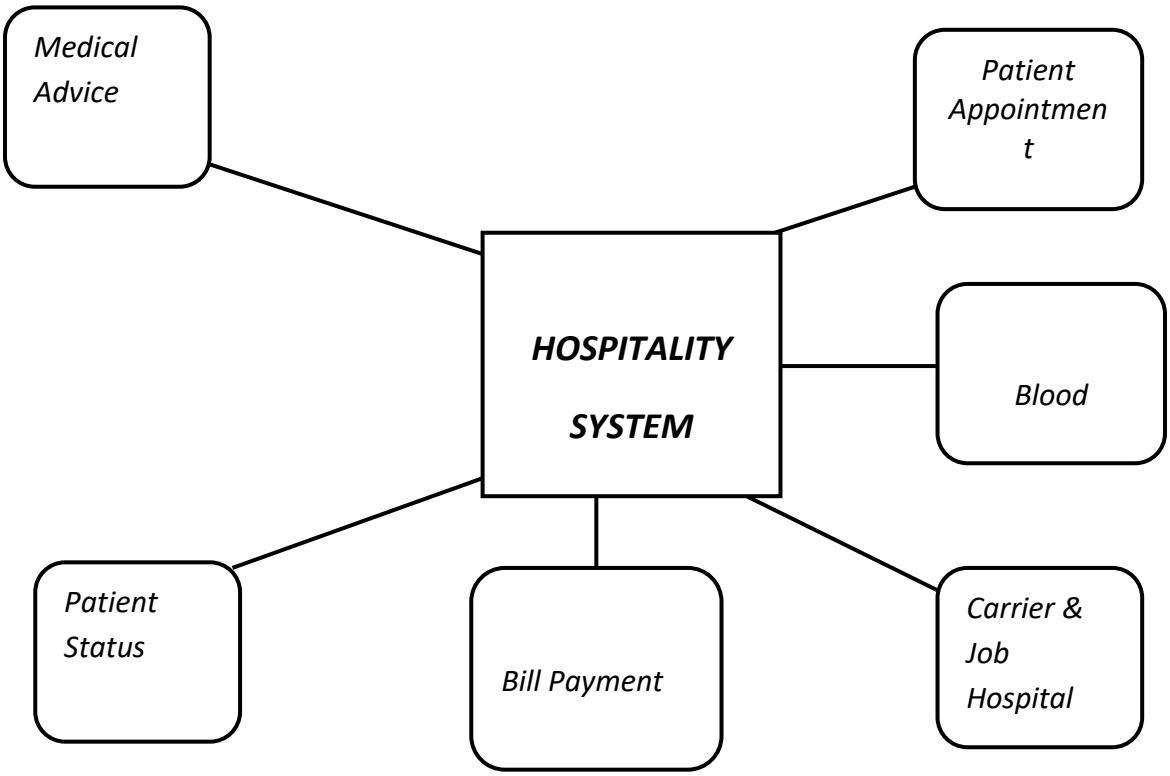


Fig. -3: Sequence Diagram

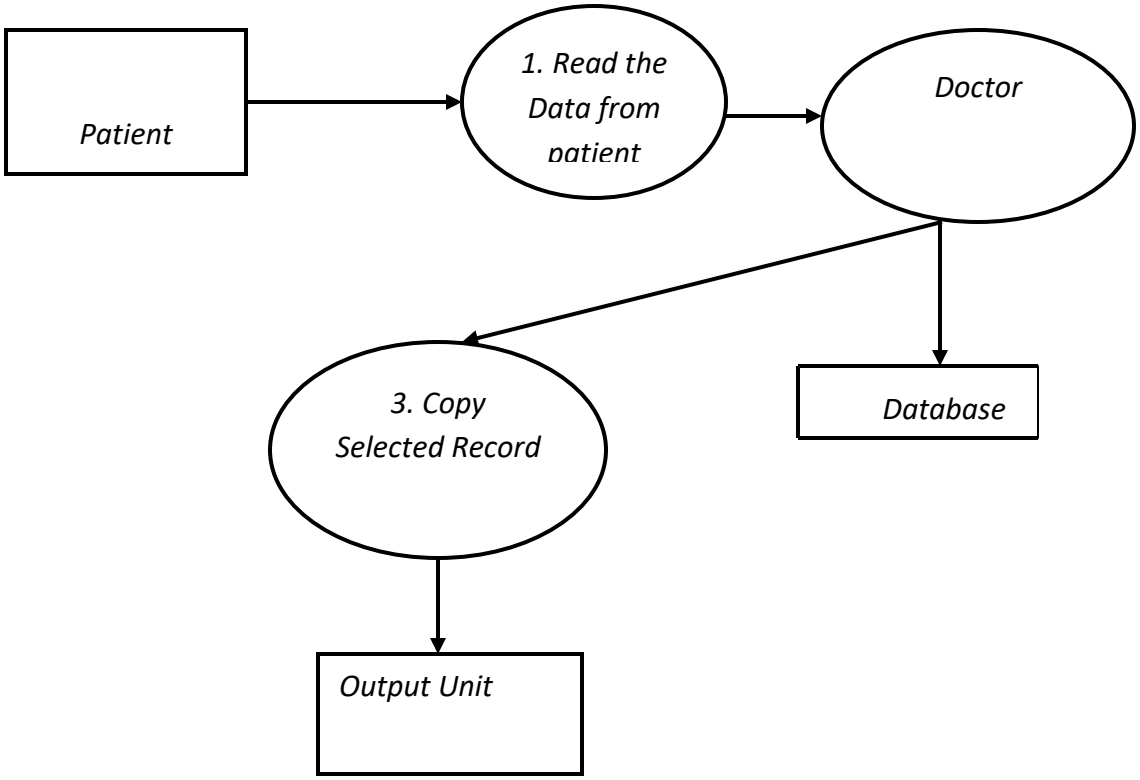
4.6 DATA FLOW DIAGRAM

Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. In our project first step is the user interaction with the website. Then, the user acquires the information about various services and doctor's specialization. The admin controls the user activities and look after the backend activities. The admin can add a new user and update the user details. The flow of information is depicted in the following data flow diagrams.

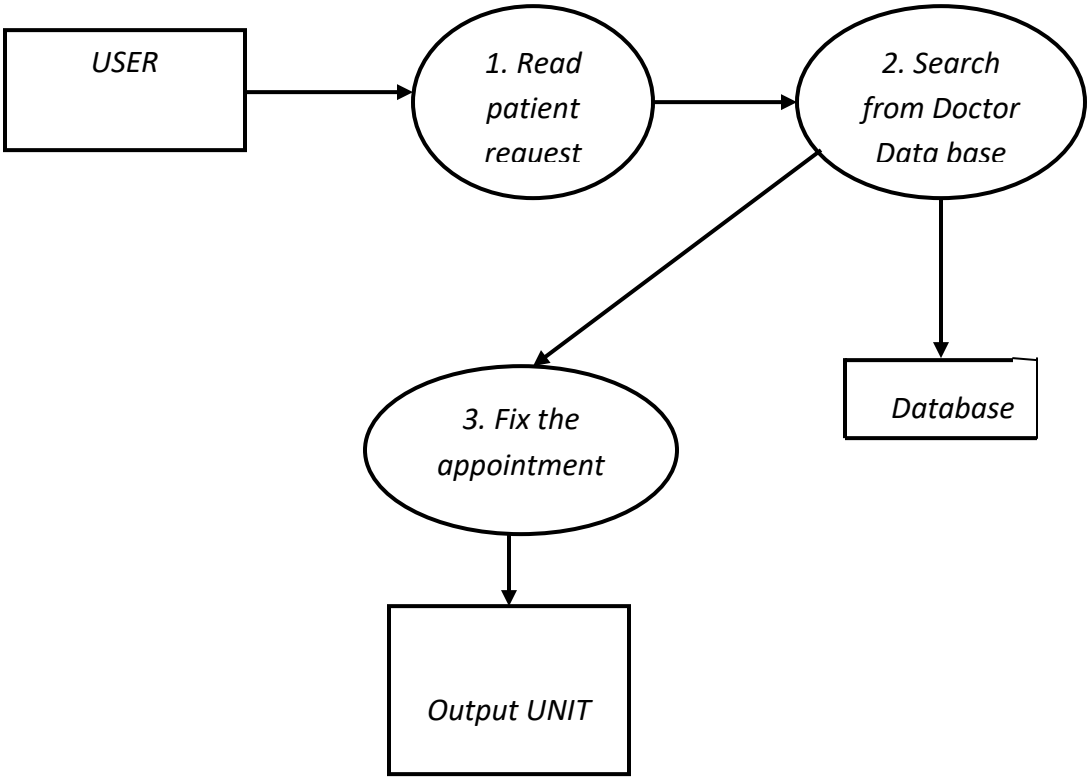
Fig. -4: Data Flow Diagrams



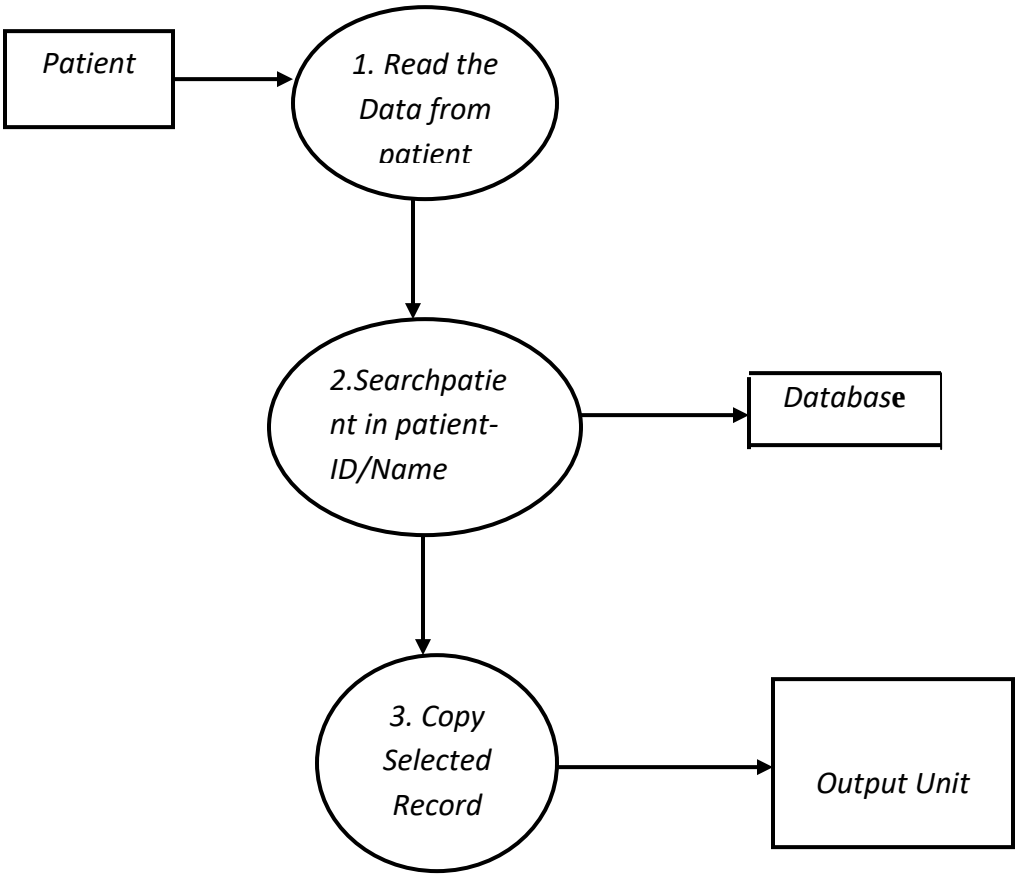
Context Level DFD



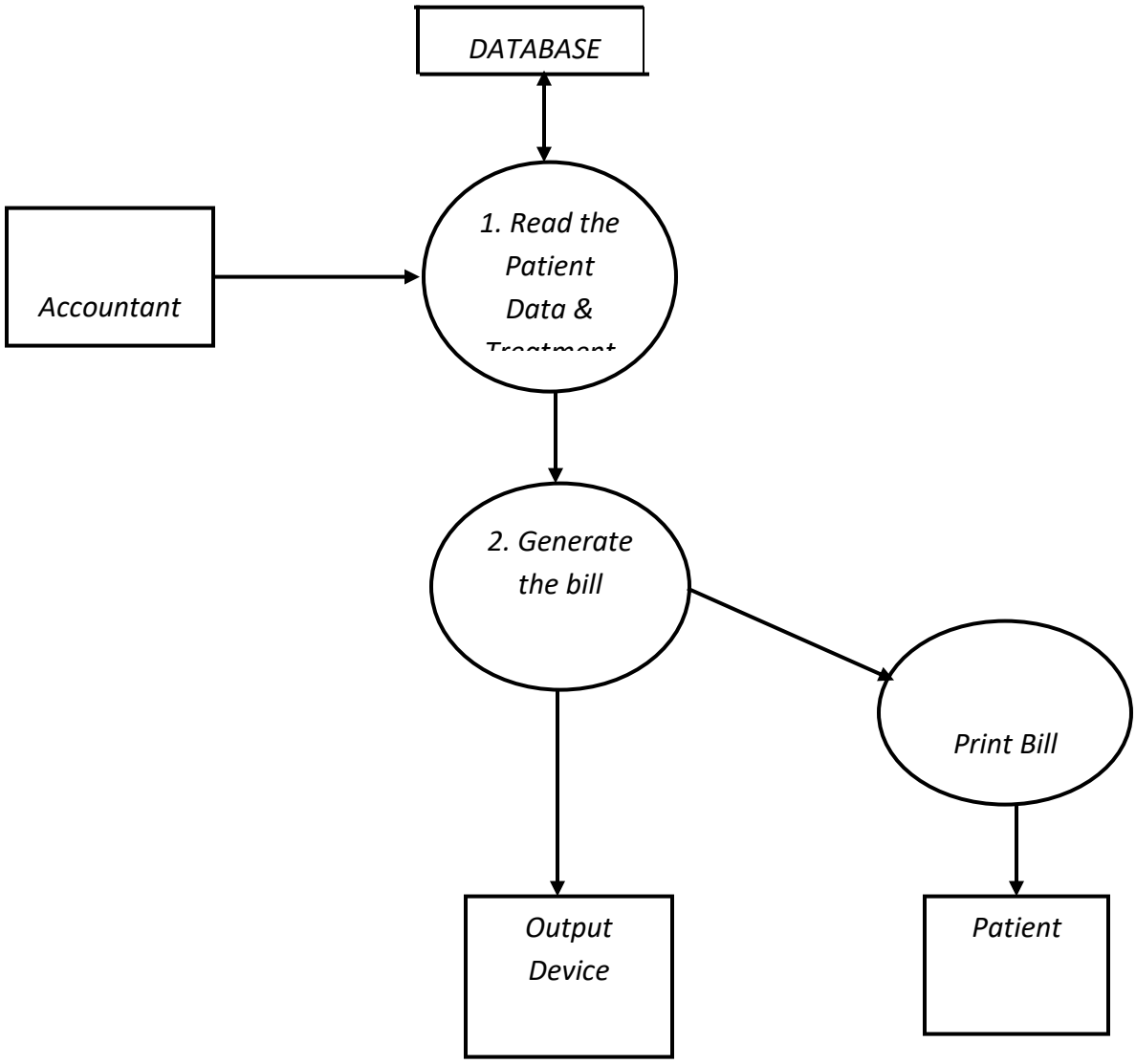
DFD for Medical Advice



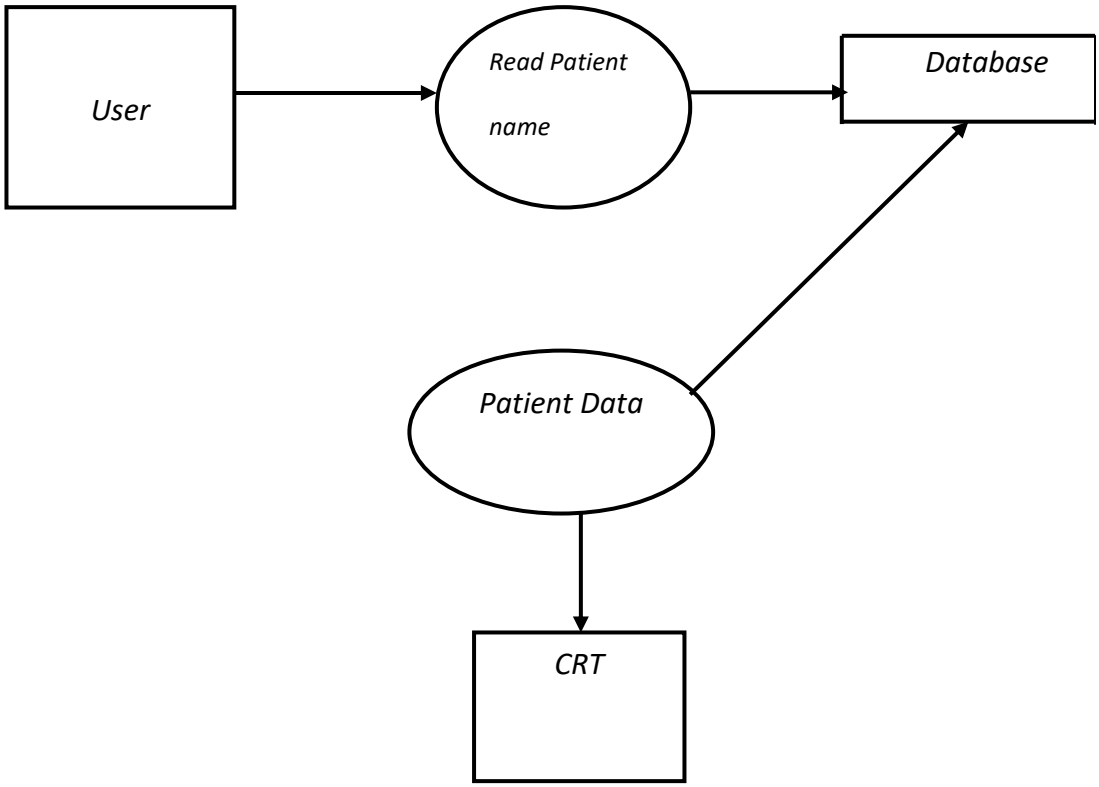
DFD for patient Appointment



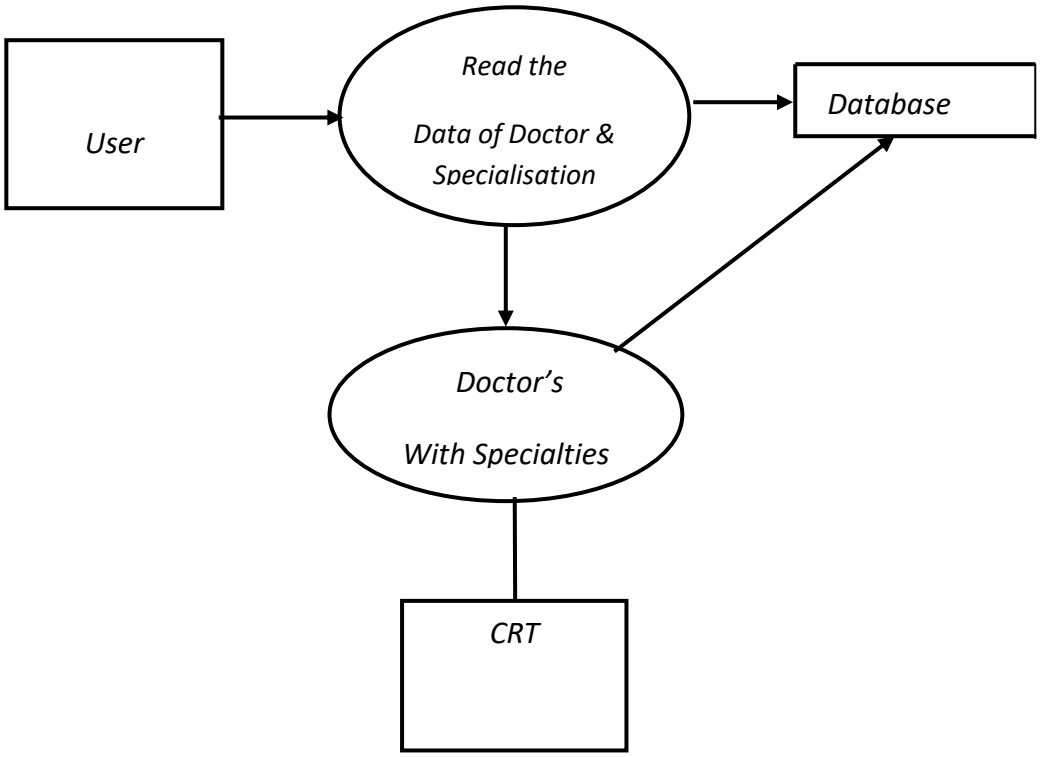
DFD for Patient Search



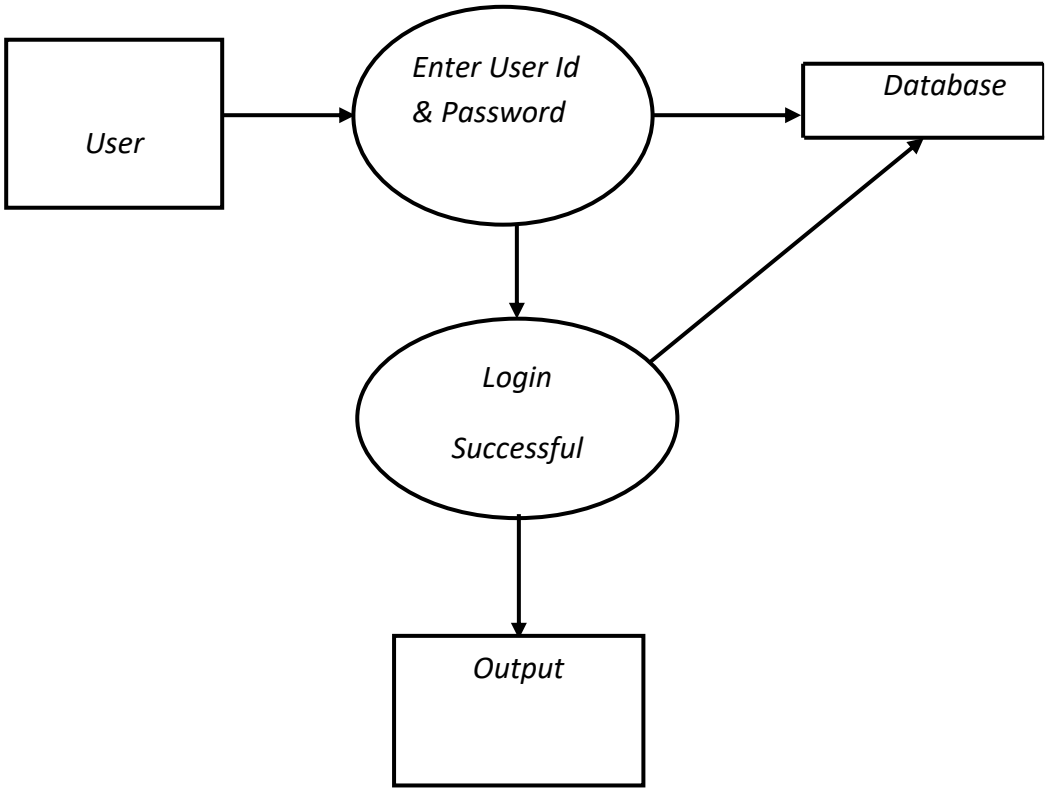
DFD For Bill Payment



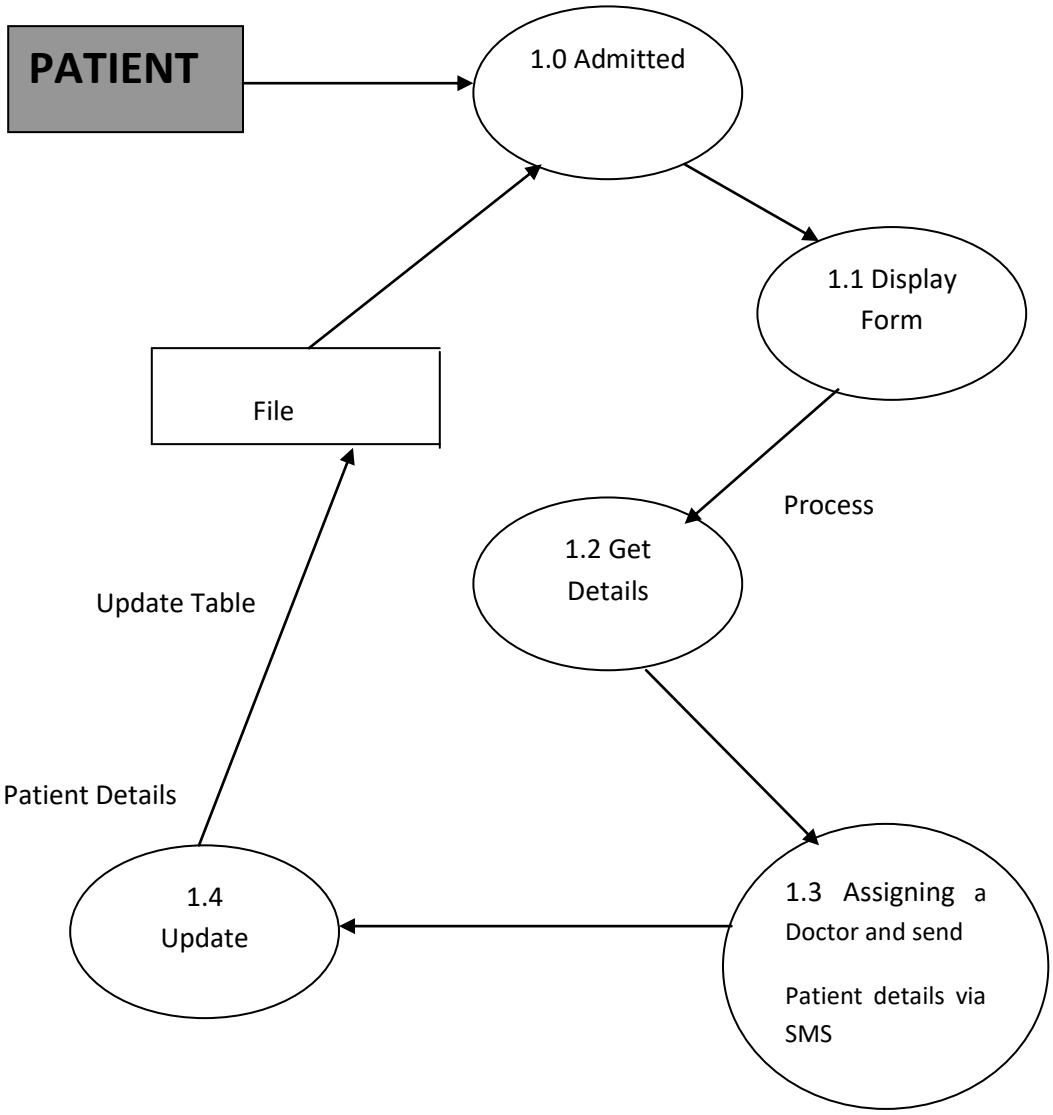
DFD For Online Searching For Patient



DFD For Searching a Doctors

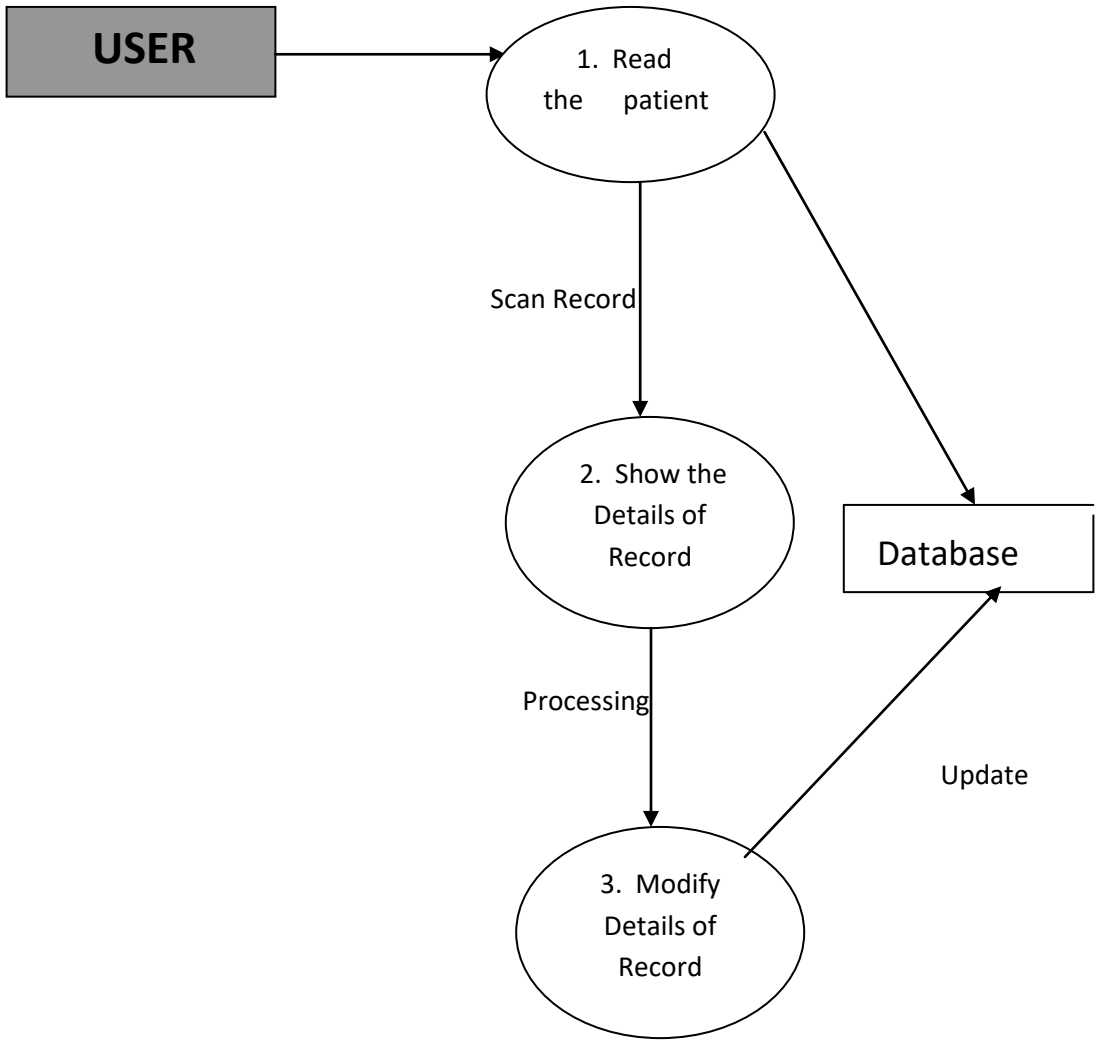


DFD For Login Of User/Admin



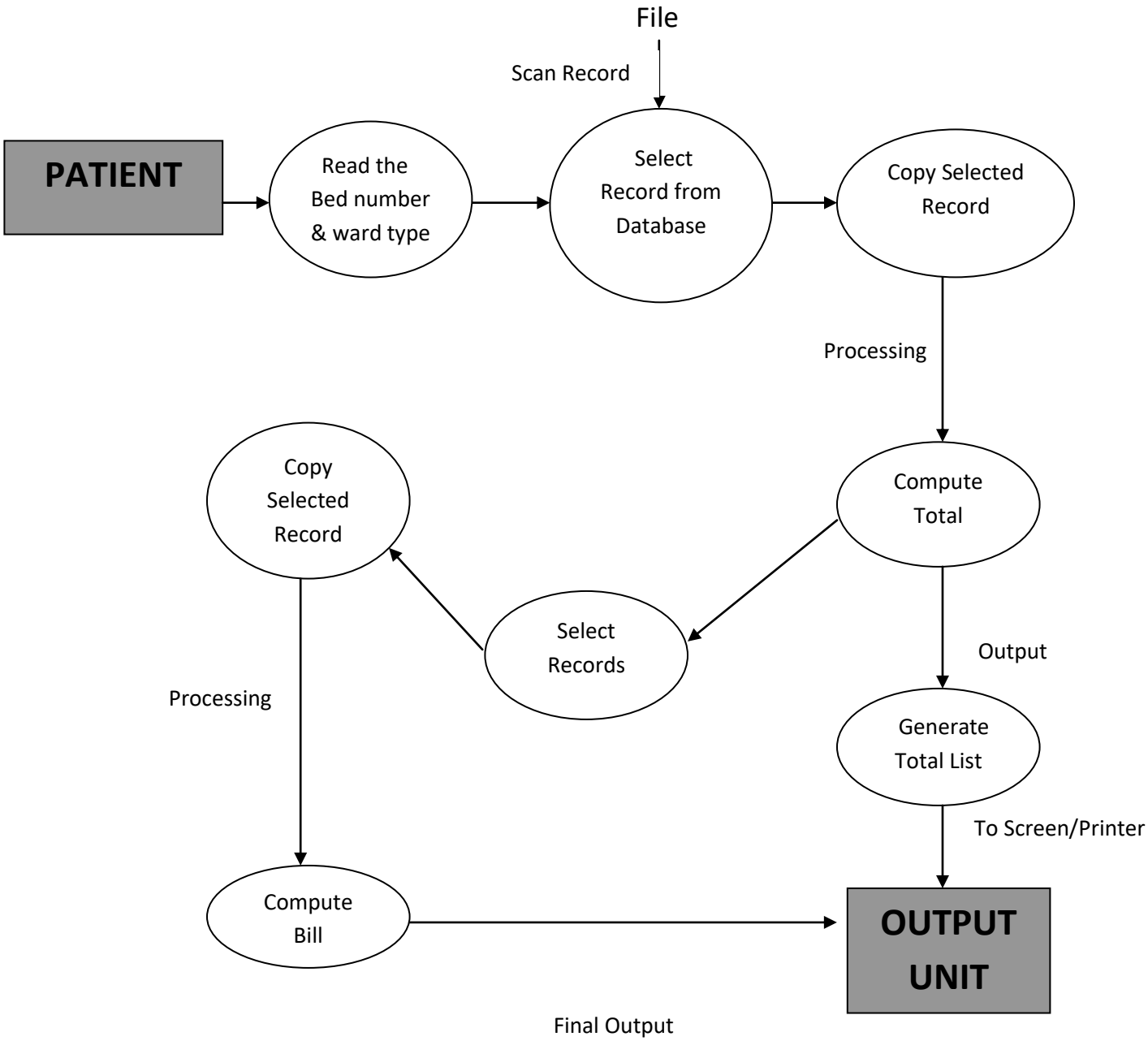
DATA FLOW DIAGRAM

ADMISSION OF A NEW PATIENT



DATA FLOW DIAGRAM

RECORD MODIFICATION



DATA FLOW DIAGRAM

LISTING OF PATIENTS

CHAPTER - 5

TESTING

5.1 TESTING

Testing is a process of executing a program with the intent of finding an error. Testing is a crucial element of software quality assurance and presents ultimate review of specification, design and coding.

System Testing is an important phase. Testing represents an interesting anomaly for the software. Thus a series of testing are performed for the proposed system before the system is ready for user acceptance testing.

A good test case is one that has a high probability of finding an as undiscovered error. A successful test is one that uncovers an as undiscovered error.

Testing Objectives:

1. Testing is a process of executing a program with the intent of finding an error
2. A good test case is one that has a probability of finding an as yet undiscovered error
3. A successful test is one that uncovers an undiscovered error

Testing Principles:

- All tests should be traceable to end user requirements
- Tests should be planned long before testing begins
- Testing should begin on a small scale and progress towards testing in large
- Exhaustive testing is not possible
- To be most effective testing should be conducted by a independent third party.

The primary objective for test case design is to derive a set of tests that has the highest livelihood for uncovering defects in software. To accomplish this objective two different categories of test case design techniques are used. They are

- White box testing.
- Black box testing.

White-box testing:

White box testing focus on the program control structure. Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been executed.

Block-box testing:

Black box testing is designed to validate functional requirements without regard to the internal workings of a program. Black box testing mainly focuses on the information domain of the software, deriving test cases by partitioning input and output in a manner that provides through test coverage. Incorrect and missing functions, interface errors, errors in data structures, error in functional logic are the errors falling in this category.

Testing strategies:

- A strategy for software testing must accommodate low-level tests that are necessary to verify that all small source code segments has been correctly implemented as well as high-level tests that validate major system functions against customer requirements.

- Data outside of the specified input range should be tested to check the robustness of the program
- Boundary cases should be tested (top and bottom of specified range) to make sure the highest and lowest allowable inputs produce proper output
- The number zero should be tested when numerical data is to be input.
- Stress testing should be performed (try to overload the program with inputs to see where it reaches its maximum capacity), especially with real time systems.

- Crash testing should be performed to see what it takes to bring the system down.
- Test monitoring tools should be used whenever possible to track which tests have already been performed and the outputs of these tests to avoid repetition and to aid in the software maintenance.
- Other functional testing techniques include: transaction testing, syntax testing, domain testing, logic testing, and state testing.
- Finite state machine models can be used as a guide to design functional tests.

Testing fundamentals:

Testing is a process of executing program with the intent of finding error. A good test case is one that has high probability of finding an undiscovered error. If testing is conducted successfully it uncovers the errors in the software. Testing cannot show the absence of defects, it can only show that software defects present.

Test Cases:

Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been executed.

Using White-Box testing methods, the software engineer can drive test cases that

- Guarantee that logical decisions on their true and false sides.
- Exercise all logical decisions on their true and false sides.
- Execute all loops at their boundaries and with in their operational bounds.
- Exercise internal data structure to assure their validity.

The test case specification for system testing has to be submitted for review before system testing commences.

5.2 Testing Techniques:

Testing consists of the following steps:

- Program Testing
- System Testing
- Performance Testing
- Configuration Testing

(i) Program Testing:

Test Objective:	In this testing I ensure that program has no syntax and logical error.
Technique:	Compiled the program and check the error.
Completion Criteria:	1. All planned tests have been executed. 2. All syntax error is removed.

(ii) System Testing:

Test Objective:	Ensured that it meets the fundamental requirement laid out in the requirement specification.
Technique:	Run the program and actual program is compared with the expected one. In case of discrepancy, the sequences of instruction are traced to determine the problem.
Completion Criteria:	1. All planed test have been executed 2. All identified defects have been addressed

Note: This type of testing is based on black-box technique that are verifying that the application conforms to the requirements by interacting with the application via the user interface and matching the actual output with the expected output

(iii) Configuration Testing:

Test Objective:	Validation and verification done by me to know that the system functions properly on all supported software & hardware configuration, described in requirement specification.
Technique:	• For each supported version of OS, run the system for difference services released. • Testing is done for coexistence with other popular pc application.
Completion Criteria:	All test completed with expected results on various combinations of hardware and software combination detailed in the requirement specification.

Note: Configuration Testing verifies the system on different software and hardware specifications.

(iv) Performance Testing:

Test Objective:	Validate system response time for designate d business functions for the targeted no of current users.
Technique:	Used different clients each run test scripts to place a load on the system.
Completion Criteria:	Successful completion of test script with expected results with expected response time.

Note: The goal of Performance Testing is to verify and validate the performance and scalability requirements.

Black box testing takes an external perspective of the test object to derive test cases. These tests can be functional or non-functional, though usually functional. The test designer

selects valid and invalid input and determines the correct output. There is no knowledge of the test object's internal structure.

This method of test design is applicable to all levels of software testing: unit, integration, functional testing, system and acceptance. The higher the level, and hence the bigger and more complex the box, the more one is forced to use black box testing to simplify. While this method can uncover unimplemented parts of the specification, one cannot be sure that all existent paths are tested.

5.3 Test Cases Analysis:

Following are some test cases are used for Hospitality System:

Test Case: 1

1.	*Form Name*	:	Admin Panel
2.	*Input for Testing*	:	(i) Testing applied on each and every data field for validation (ii) Validation while inserting data into the database.
3.	*Testing Apply*	:	(i) All fields must be entered valid data otherwise <"all fields are required to fill otherwise it gives message"> (ii) Check Login ID. If login id is correct the user must login successfully otherwise it should say go to unauthorized user.
4.	*Test Result*	:	Ok
5.	*Remedy*	:	Is Not Required.
6.	*Alpha testing again*	:	Tested Success Successfully.

Test Case: 2

1.	*Form Name*	:	Register
2.	*Input for Testing*	:	(i) Testing applied on each and every data field for validation (ii) Validation while inserting data into the database. (iii) If integer is entered in place of String it should give an exception.
3.	*Testing Apply*	:	Input data and click on save button Add save properly
4.	*Outcome*	:	It gives Exception.
5.	*Test Result*	:	Ok
6.	*Remedy*	:	Is Not required
7.	*Alpha testing again*	:	Tested Success Successfully

Test Case: 3

1.	*Form Name*	:	Patient Form
2.	*Input for Testing*	:	It should insert field in database.
3.	*Testing Apply*	:	Insert and Save the data in data base
4.	*Outcome*	:	Data has been properly saved in user data base
5.	*Test Result*	:	Ok
6.	*Remedy*	:	Not Required
7.	*Alpha Testing*	:	Tested Successfully

Test Case: 4

1.	*Form Name*	:	Doctor
2.	*Input for Testing*	:	Values for each data field in user database
3.	*Testing Apply*	:	Feedback should be submitted. Admin can view feedback.
4.	*Outcome*	:	Submitted and Admin is able to view feedback form.
5.	*Test Result*	:	Ok
6.	*Remedy*	:	Not Required
7.	*Alpha Testing*	:	Test successfully

CHAPTER-6

RESULT ANALYSIS

The Hospitality System is the software which reduces the manual work of the hospital and keeps the patient records and details updated. The database saves all the important detailed records for reference purpose.

Here we have given step by step procedure to use the software and explained its working with the help of the snapshots of the software. After going through these steps one can easily use this software.

Below are the snapshots of our Hospitality System and its working:

1. Screen Shot of the Main Interface of My Hospital:

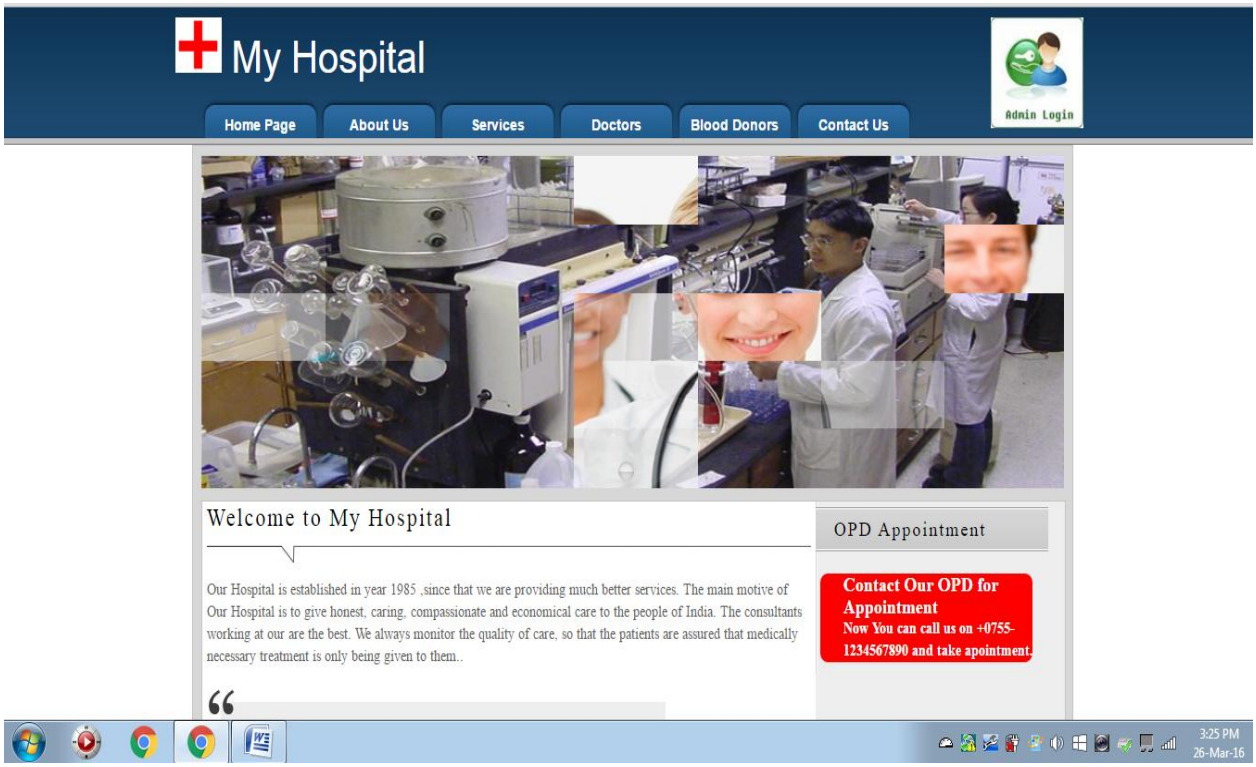


Fig. -5: Home Page

2. Screenshot of the details on front page of the website which includes the contact details for the appointment.

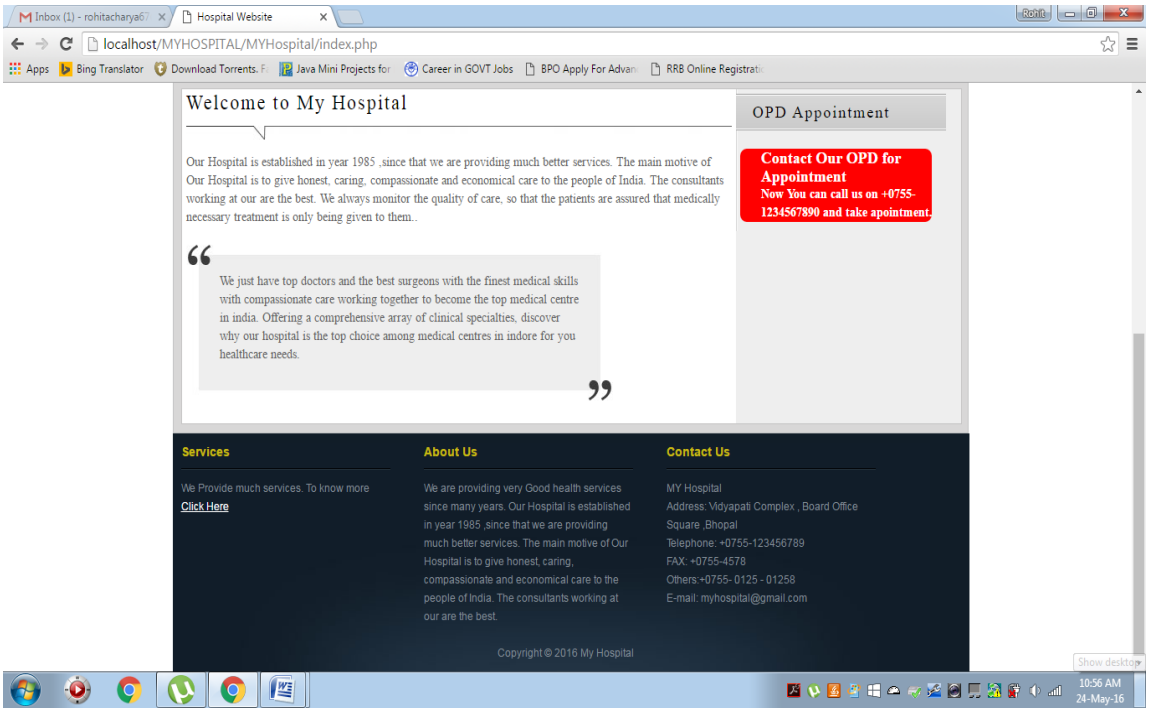
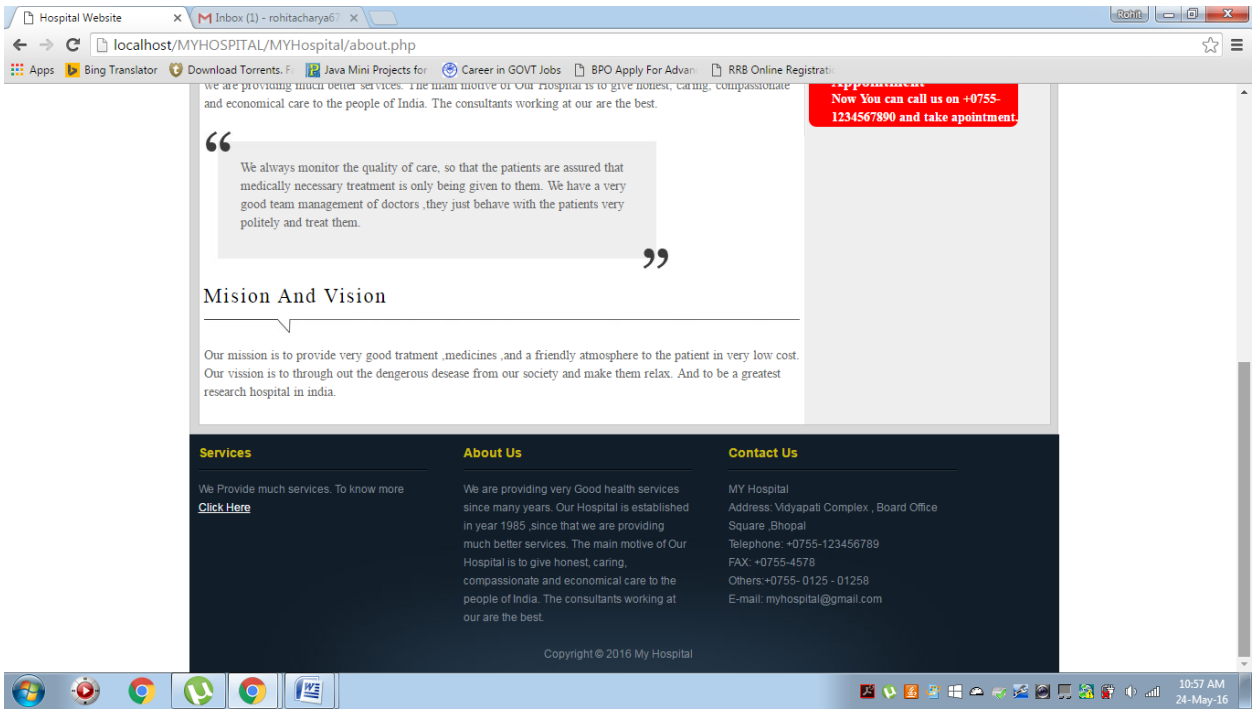


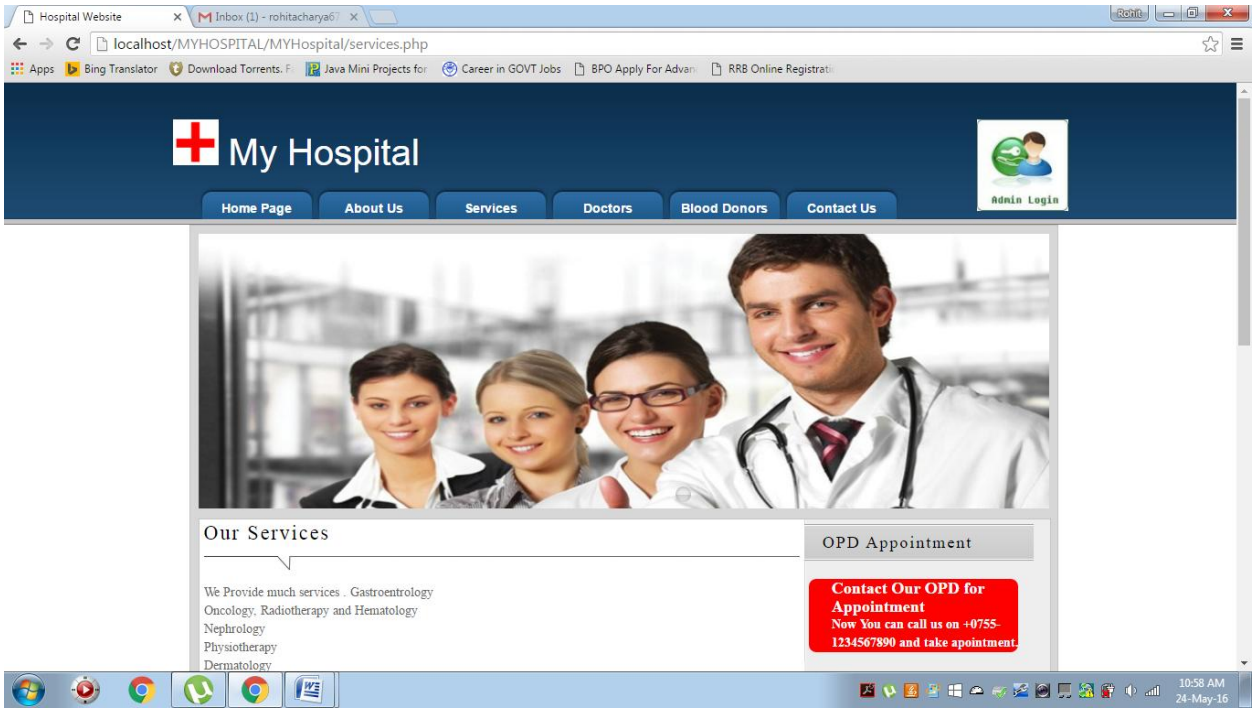
Fig. -6: Details of the website

3. Details About the hospital.

Fig. -7: About Us



4. Screenshot showing various services provided by the hospital.



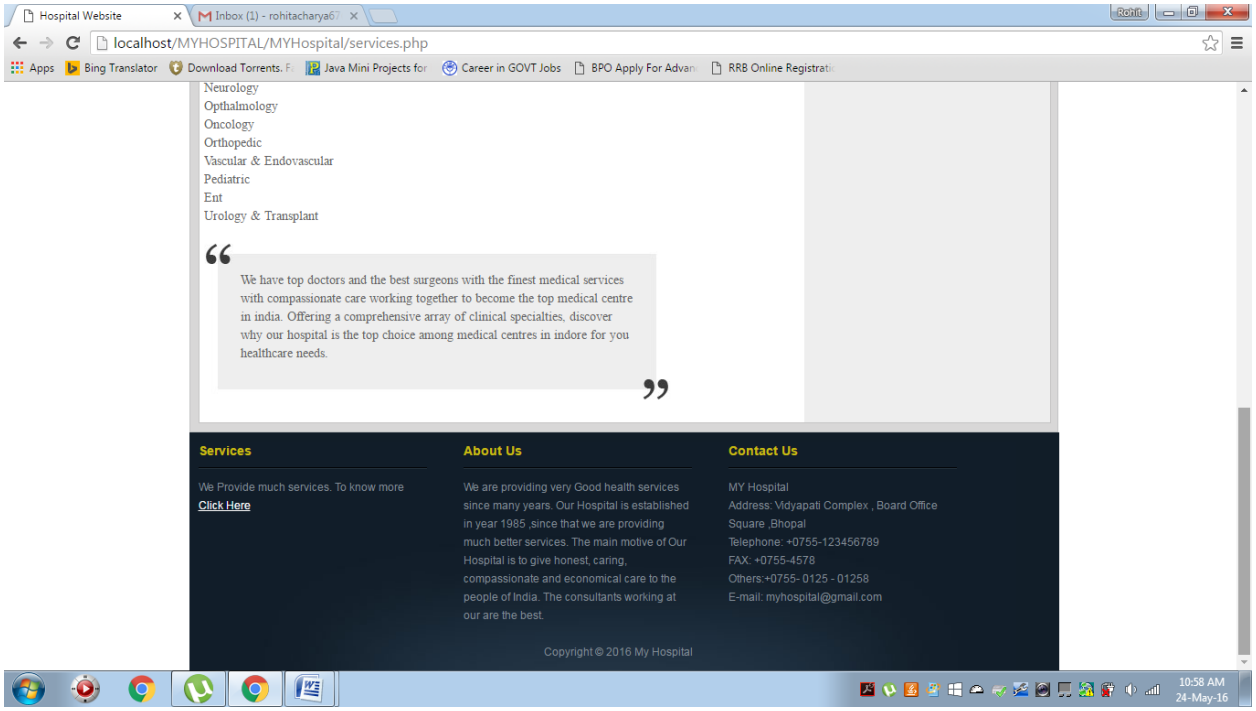
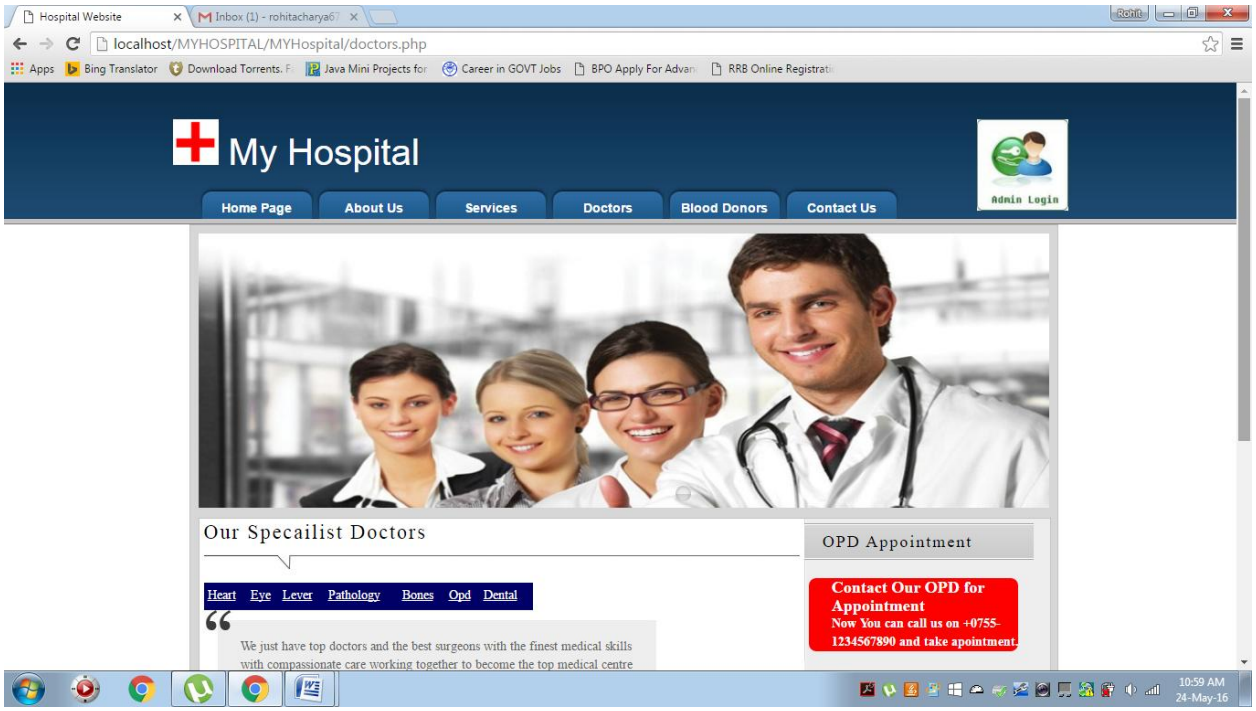


Fig. -8: Various Services Offered

5. Screenshot showing the details of the specialist doctors available at the hospital.



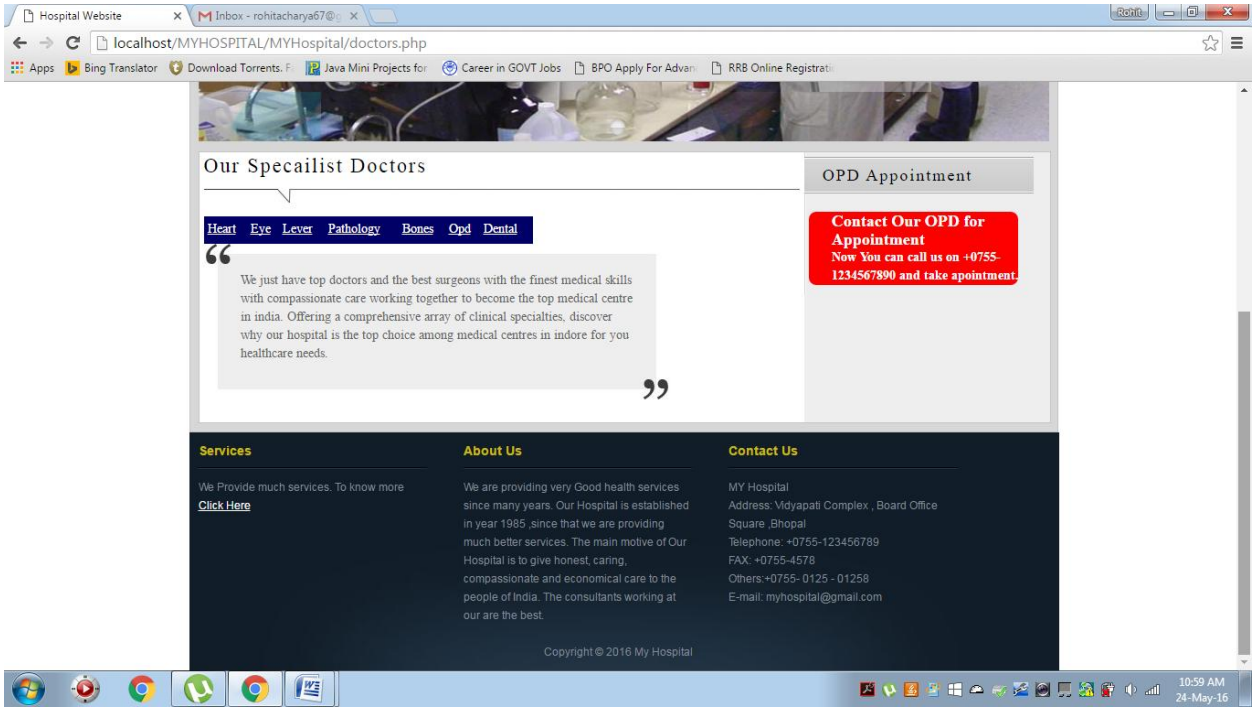


Fig. -9: Our Specialist Doctors

6. Record of Blood Donors available in the Blood Bank of the hospital.

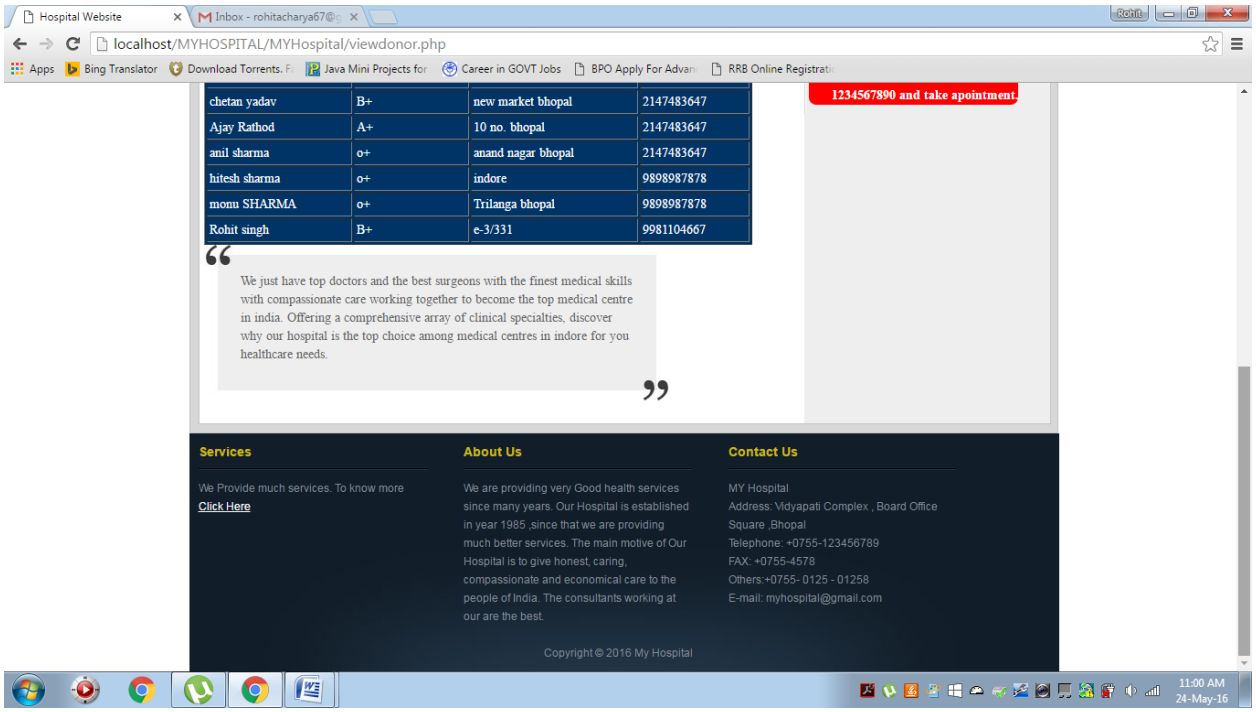


Fig. -10: Blood Donors

7. Contact details for any further enquiries and queries.

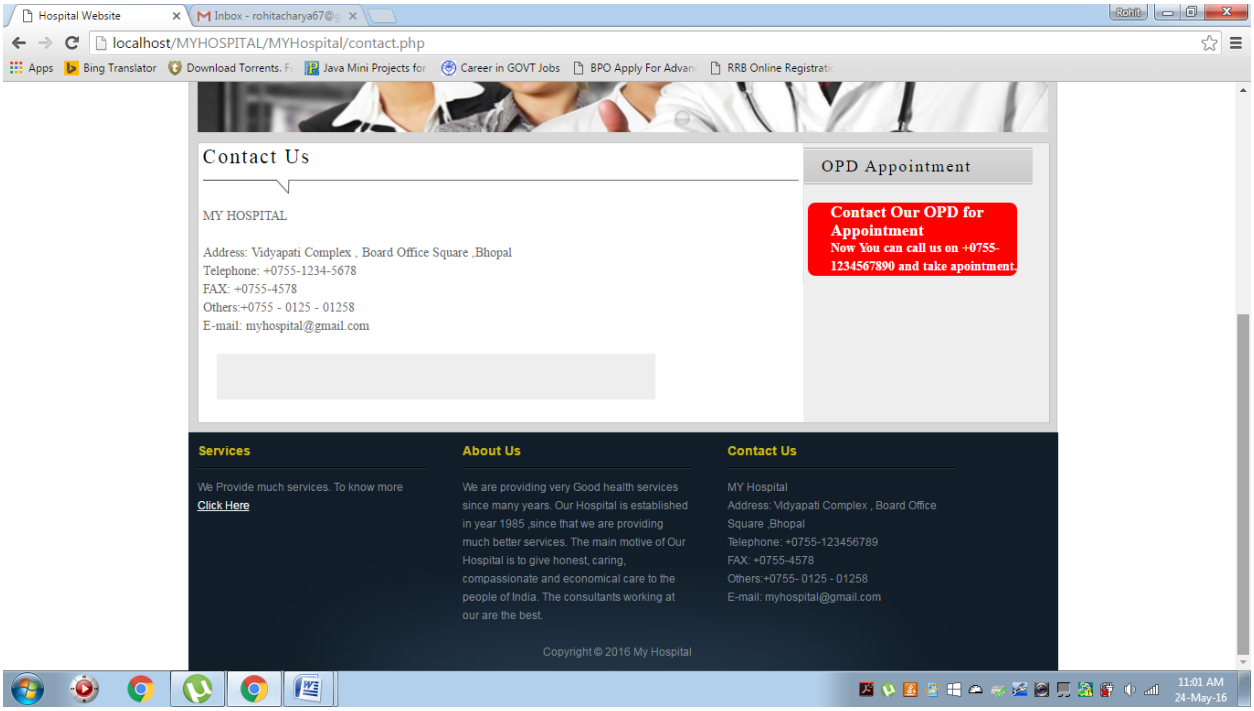
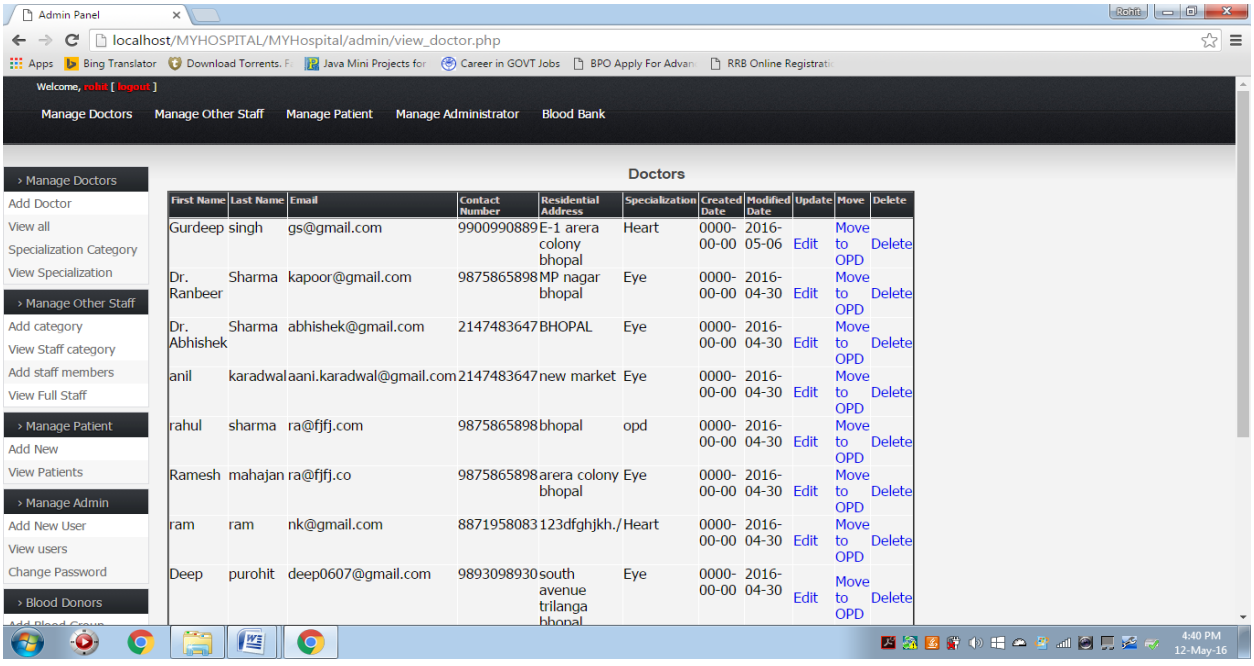


Fig. -11: Contact Us

8.

Fig. -12: Admin Login Page

9. Backend Screenshot of all the doctors of the hospital with their respective details like Name, Email Id, and specialization.



First Name	Last Name	Email	Contact Number	Residential Address	Specialization	Created Date	Modified Date	Update	Move	Delete
Gurdeep	singh	gs@gmail.com	9900990889	E-1 arera colony bhopal	Heart	0000-00-00	2016-05-06	Edit	Move to OPD	Delete
Dr. Ranbeer	Sharma	kapoor@gmail.com	9875865898	MP nagar bhopal	Eye	0000-00-00	2016-04-30	Edit	Move to OPD	Delete
Dr. Abhishek	Sharma	abhishek@gmail.com	2147483647	BHOPAL	Eye	0000-00-00	2016-04-30	Edit	Move to OPD	Delete
anil	karadwal	aani.karadwal@gmail.com	2147483647	new market	Eye	0000-00-00	2016-04-30	Edit	Move to OPD	Delete
rahul	sharma	ra@fjfj.com	9875865898	bhopal	opd	0000-00-00	2016-04-30	Edit	Move to OPD	Delete
Ramesh	mahajan	ra@fjfj.co	9875865898	arera colony bhopal	Eye	0000-00-00	2016-04-30	Edit	Move to OPD	Delete
ram	ram	nk@gmail.com	8871958083	123dfghjkh./	Heart	0000-00-00	2016-04-30	Edit	Move to OPD	Delete
Deep	purohit	deep0607@gmail.com	9893098930	south avenue trilanga bhopal	Eye	0000-00-00	2016-04-30	Edit	Move to OPD	Delete

Fig. -13: List of all Doctors

10.

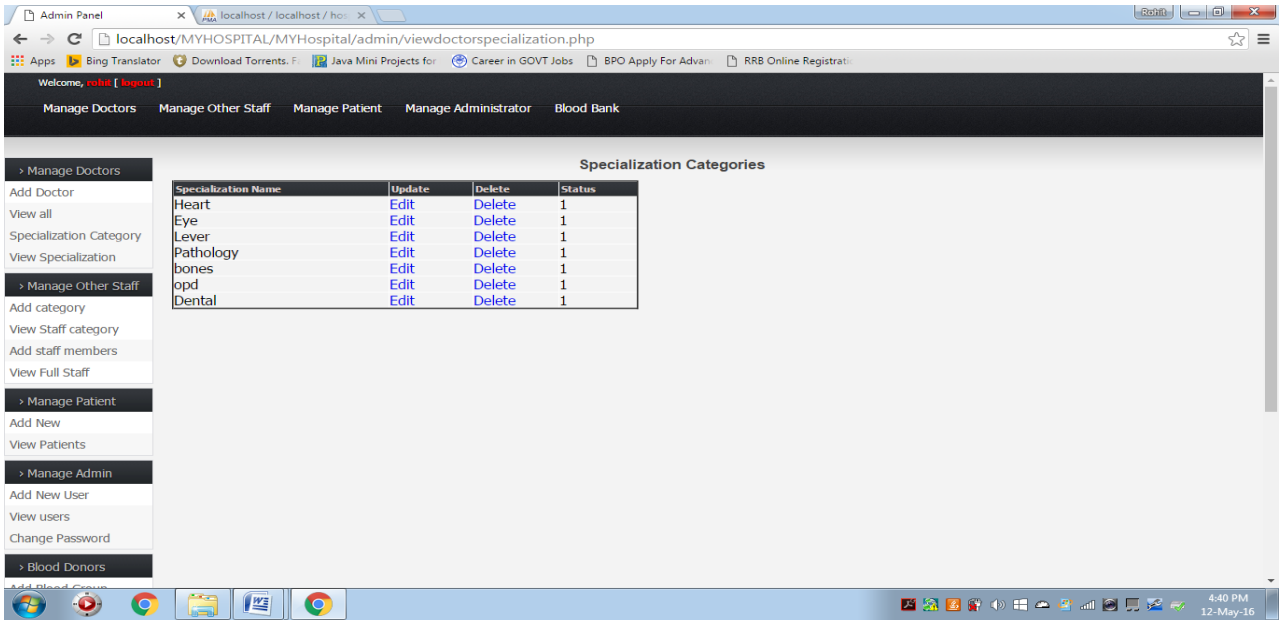


Fig. -14: Doctor's Specialization

11. Category of other staff members.

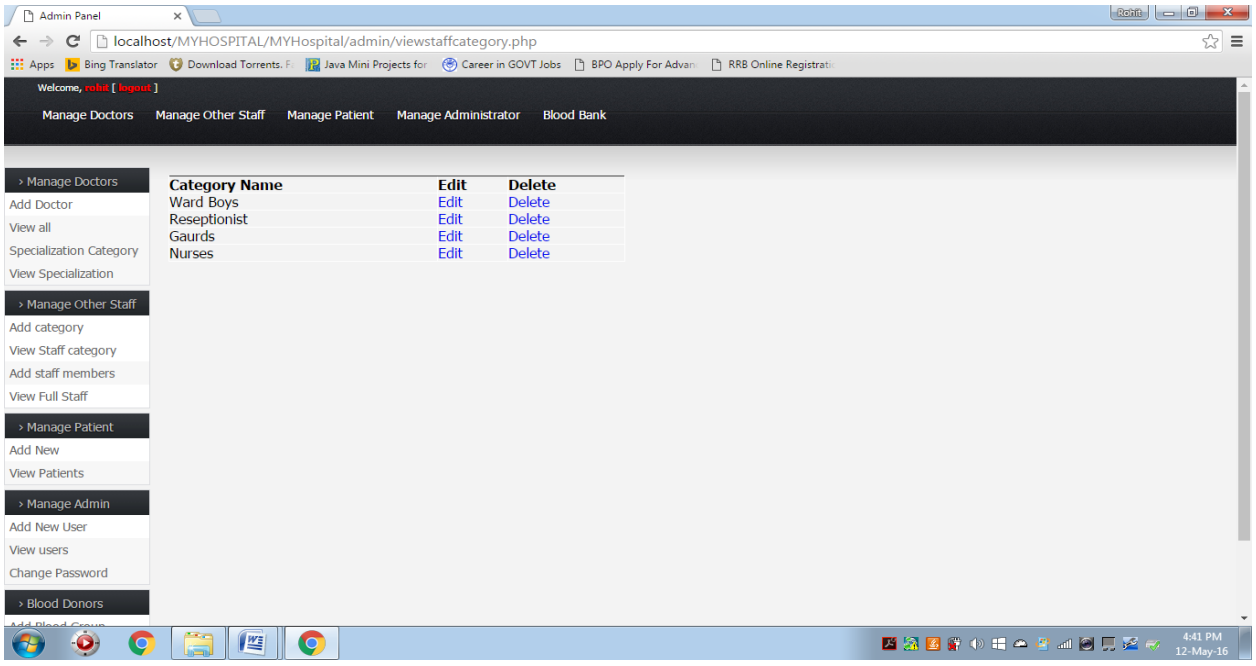


Fig. -15: Other Staff

12. The Admin can add a new member to the list of staff.

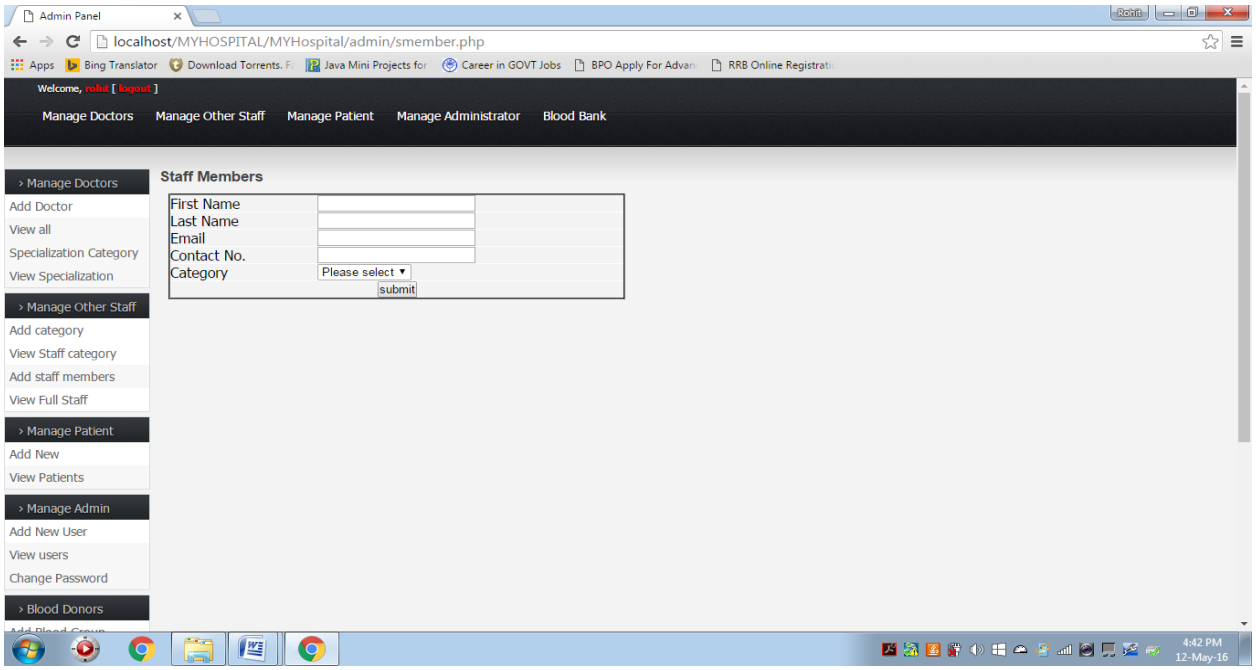


Fig. -16: Add Staff Member

13. Registration of a new patient.

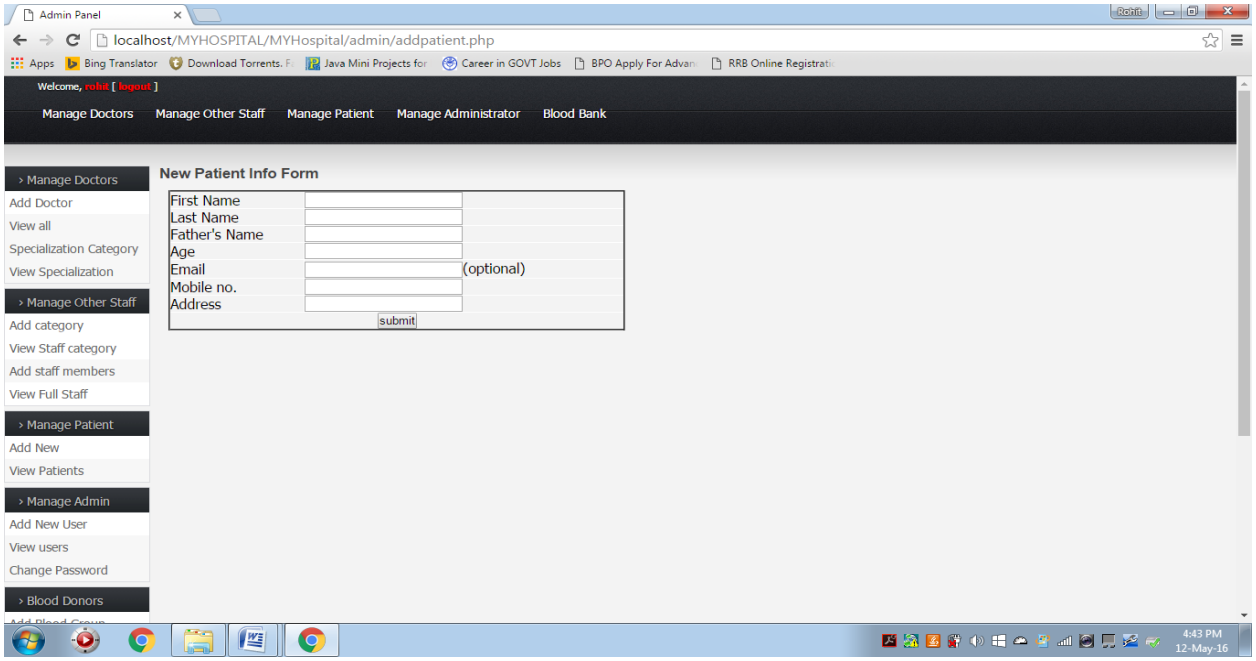


Fig. -17: Add New Patient

14. Detailed information of admitted patient's to the hospital.

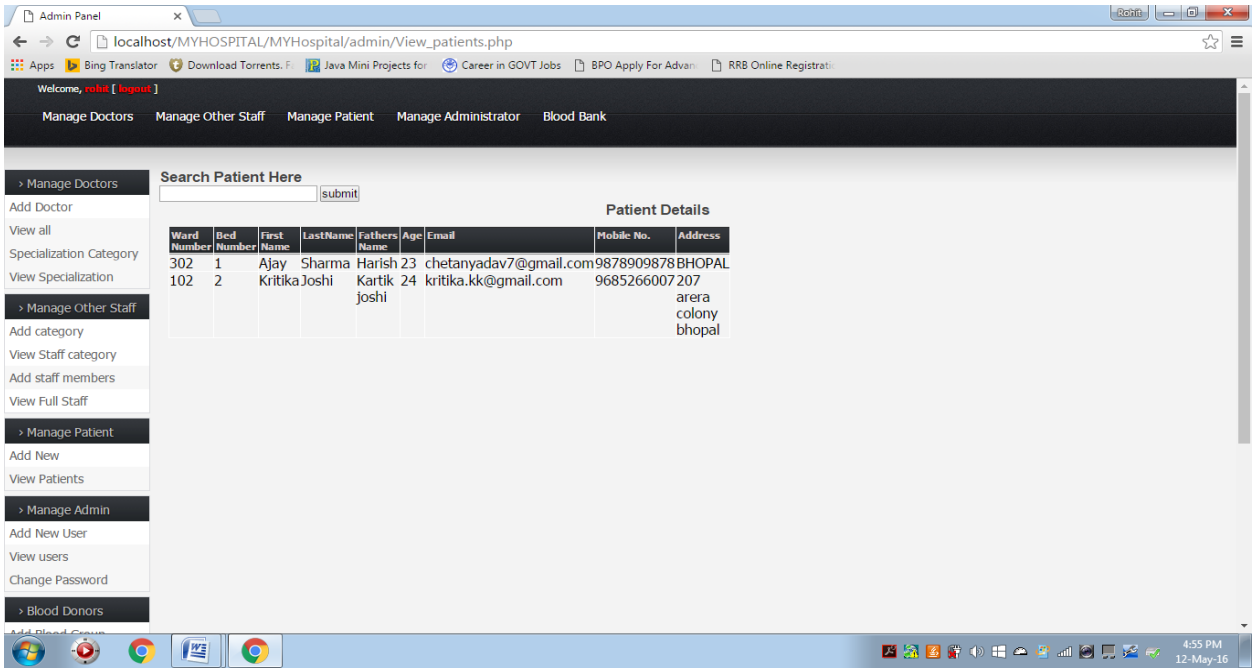


Fig. -18: Admitted Patient Details

15. Addition of new administrator/receptionist.

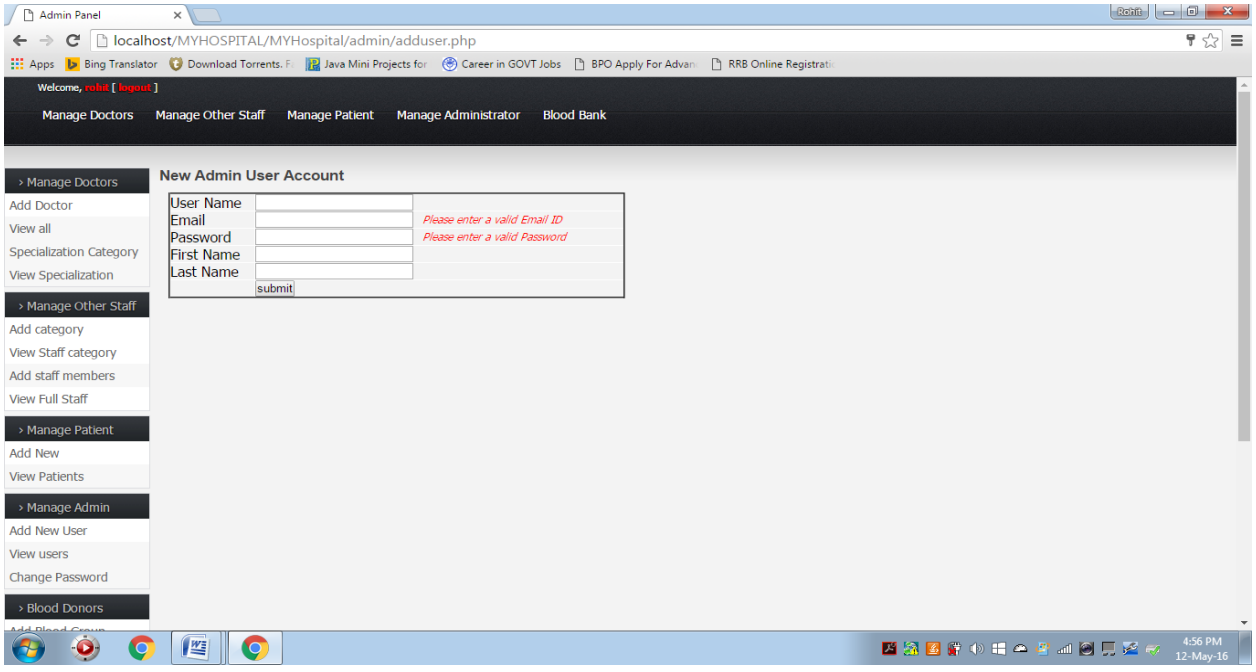


Fig. -19: New Admin.

16. Details of the current administrators.

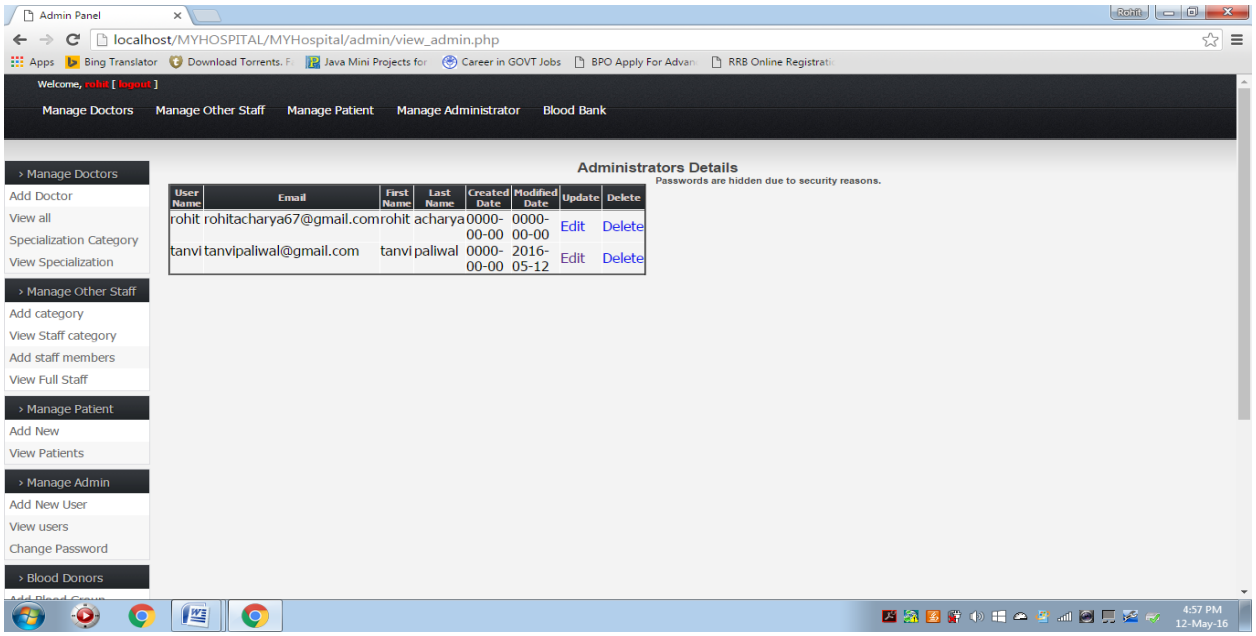


Fig. -20: Administration Details

17. Space to add a new blood group details.

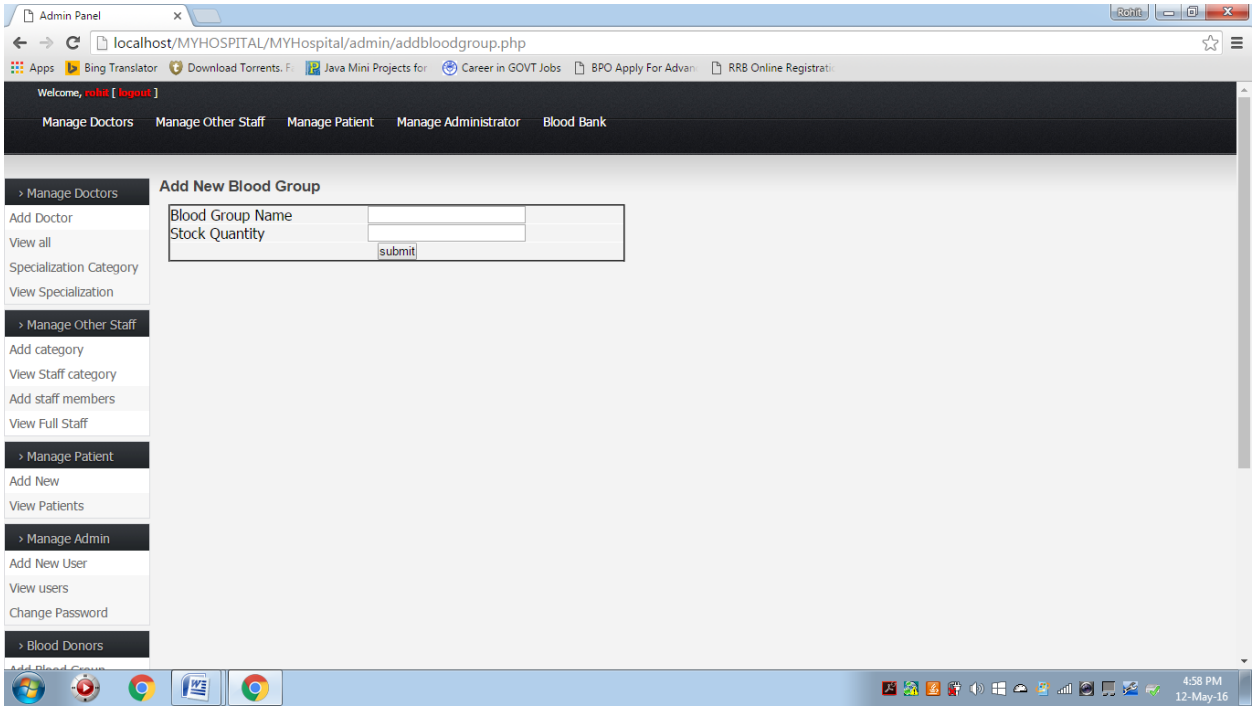


Fig. -21: Add Blood Group Name

18.

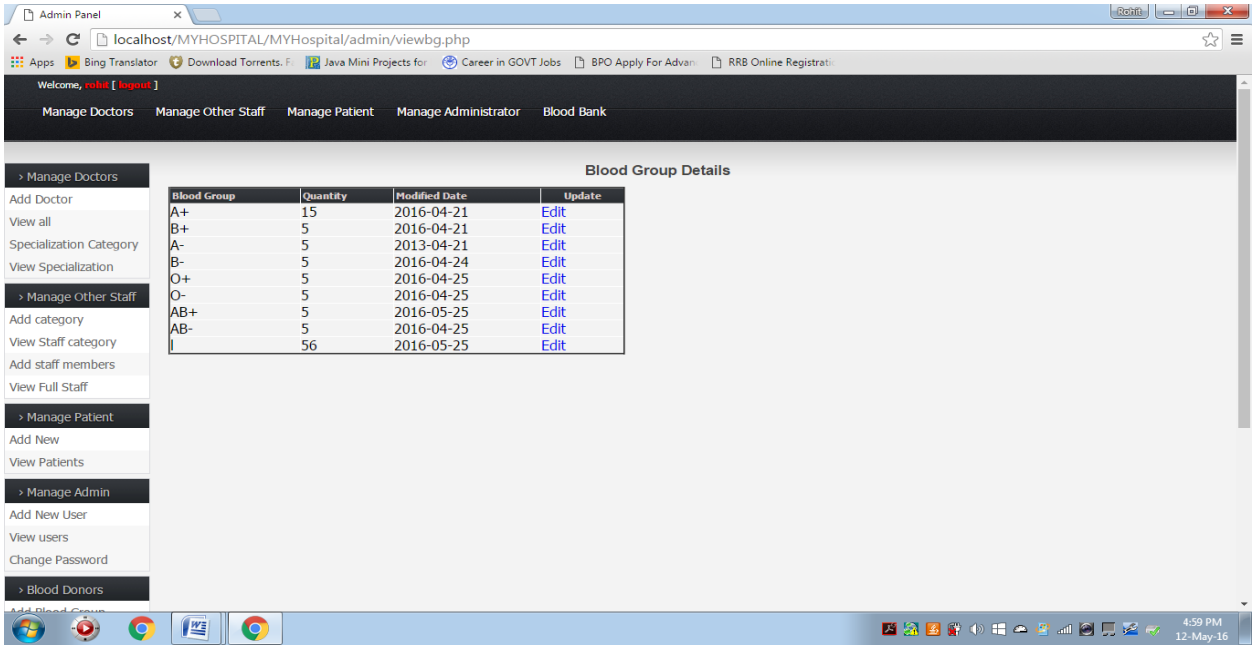


Fig. -22: Blood Groups in Stock

19. Space provided to keep the details of the blood donor.

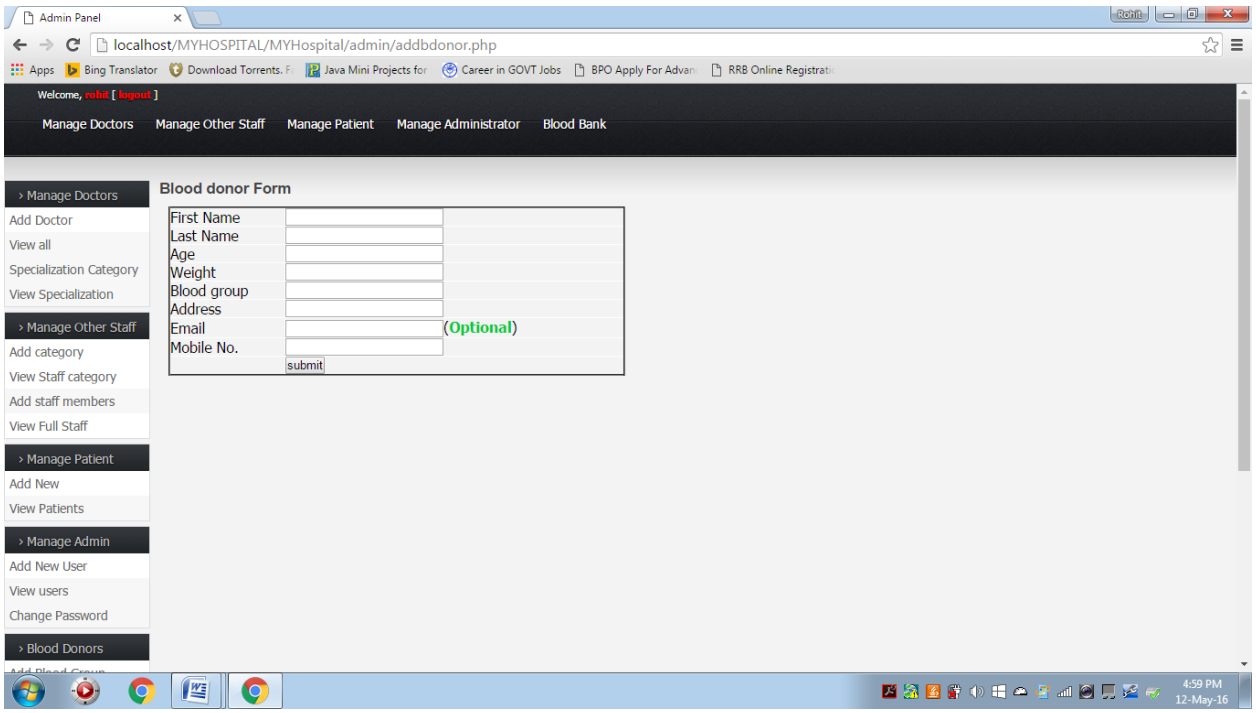


Fig. -23: Add Blood Donor

20.

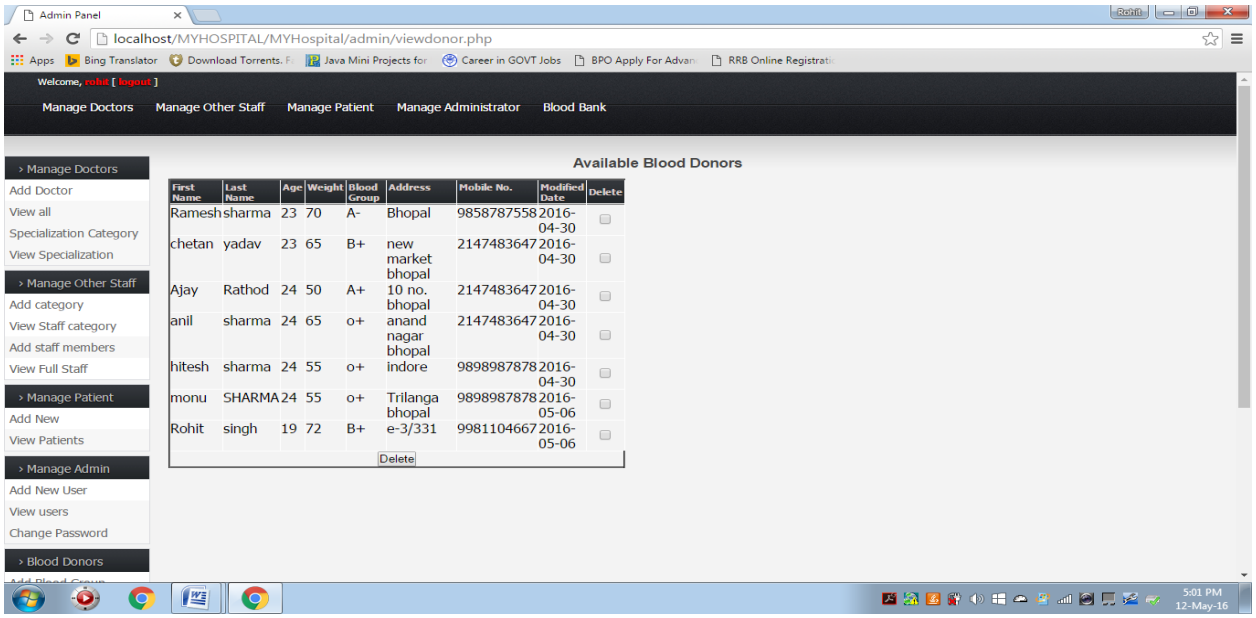


Fig. -24: List of all available Blood Donors

DATABASE AND TABLES-

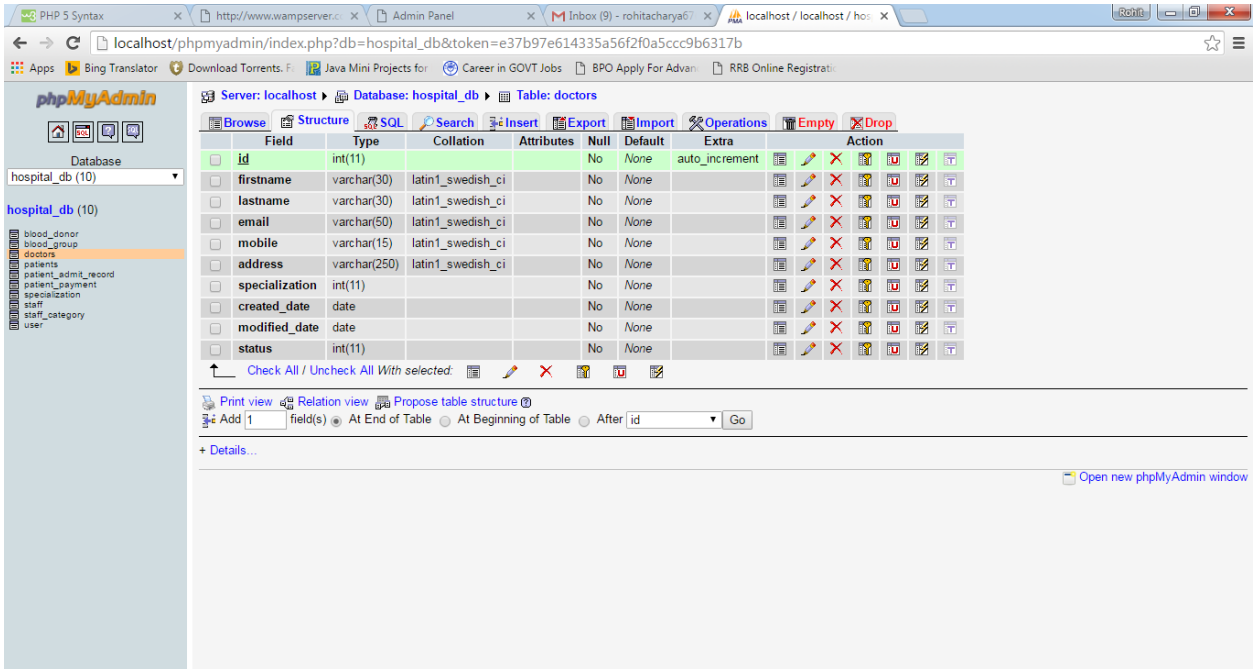


Fig. -25: Table of Doctors

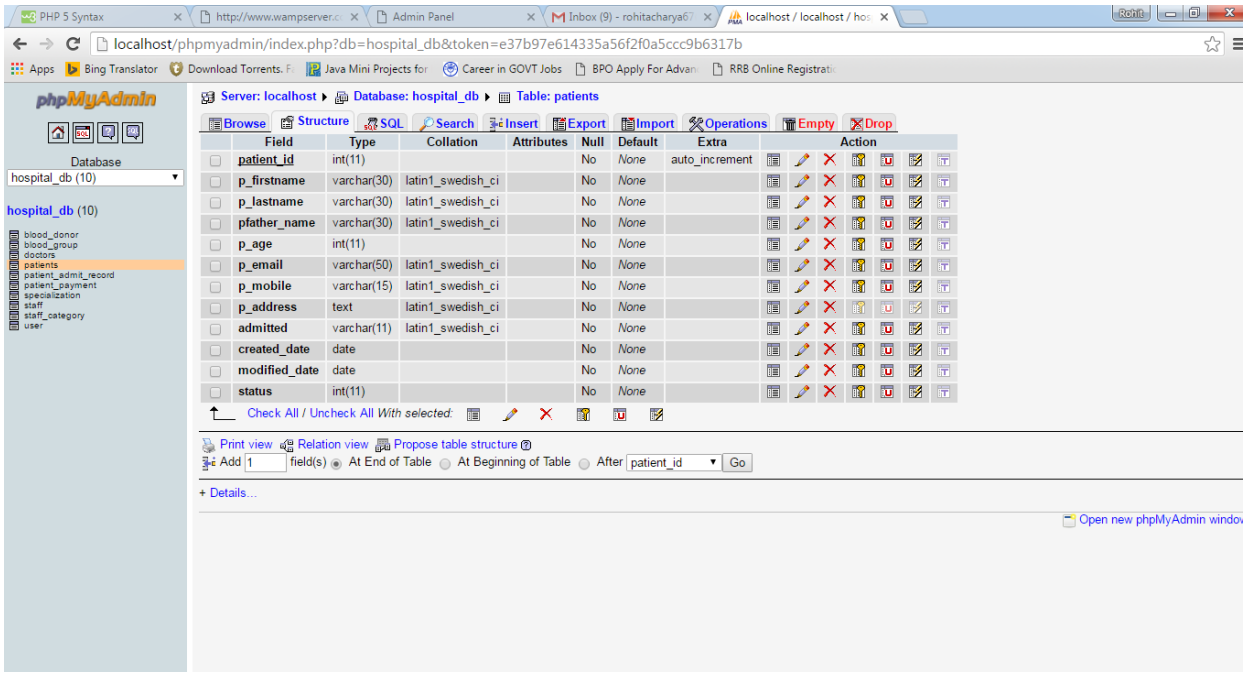


Fig. -26: Table of Patients

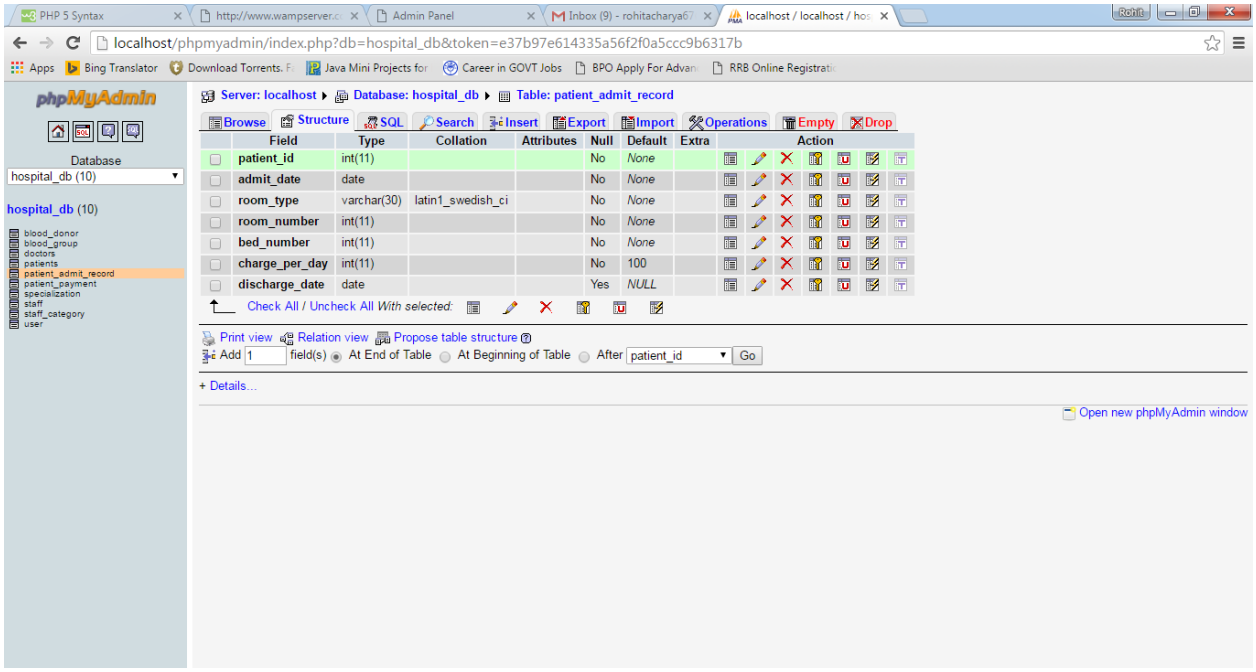


Fig. -27: Table of Admitted Patients

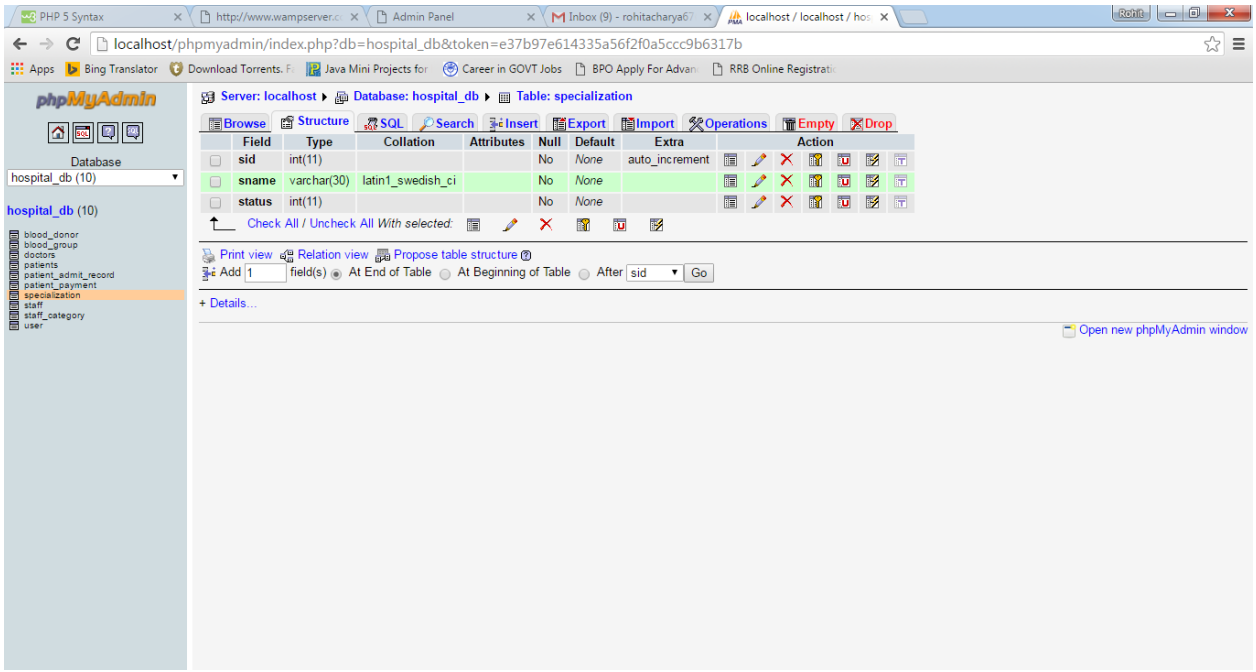


Fig. -28: Table of Doctor's Specialization

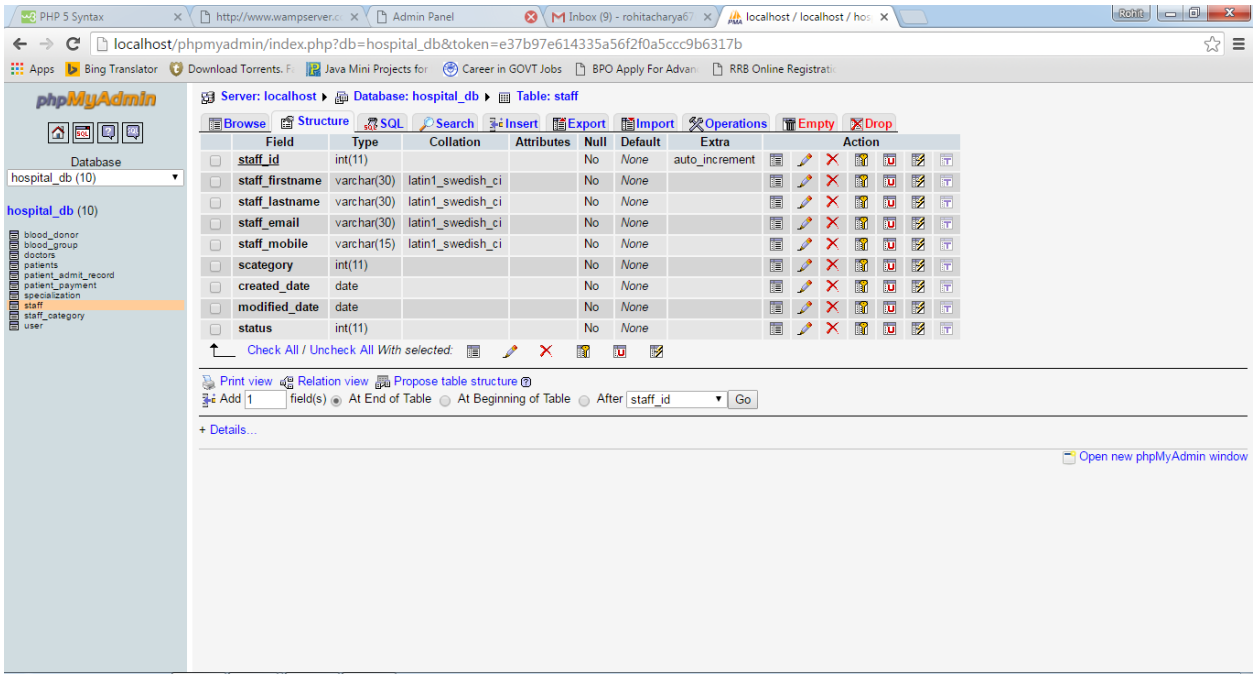


Fig. -29: Table of Staff Details

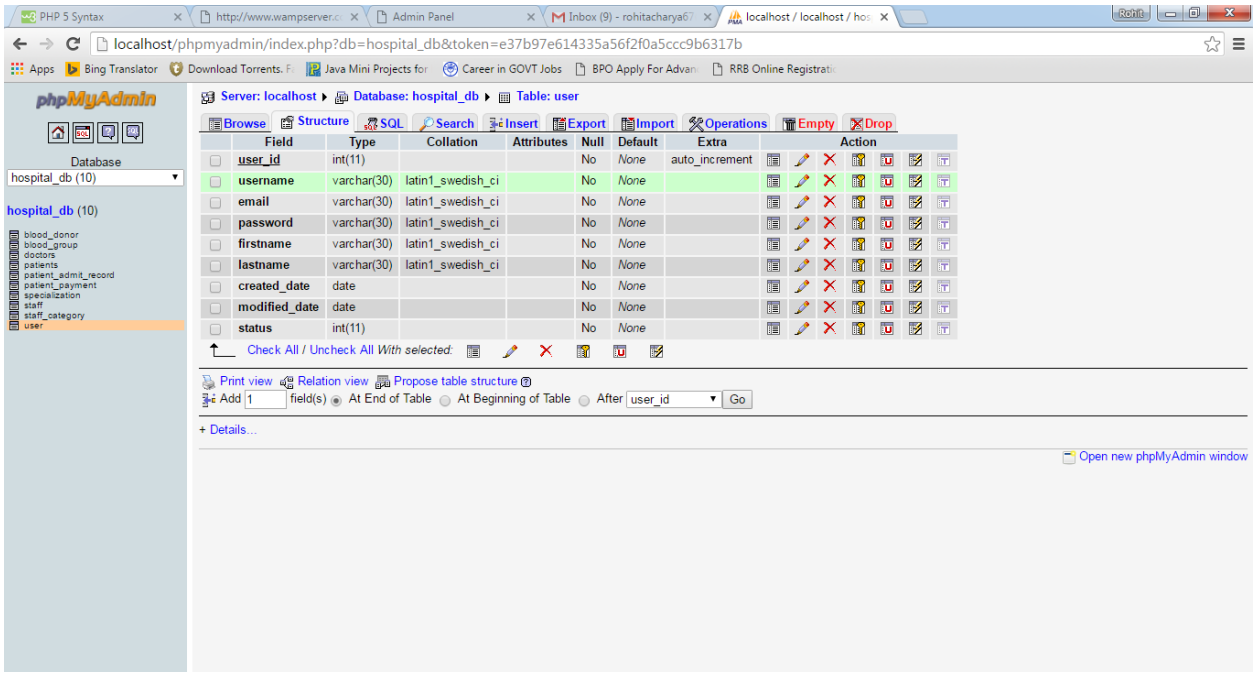


Fig. -30: Table of User Details

CHAPTER – 7

CONCLUSION AND FUTURE WORK

7.1 CONCLUSION:

This project has been a rewarding experience in more than one way. The entire project work has enlightened us in the following areas:

a) We have gained an insight into the working of the HOSPITAL. This represents a typical real world situation.

b) Our understanding of database design has been strengthened this is because in order to generate the final reports of database designing has to be properly followed.

c) Scheduling a project and adhering to that schedule creates a strong sense of time management.

d) Sense of teamwork has developed and confidence of handling real life project has increased to a great extent.

e) Initially, there were problem with the validation but with discussions, we were to implement validations.

The package was designed in such a way that future modifications can be done easily. The following conclusion can be deduced from the development of the project.

- Automation of the entire system improves the efficiency.
- It provides a friendly graphical user interface which proves to be better when compared to the existing system.
- It gives appropriate access to the authorized users depending on their permissions.
- It effectively overcomes the delay in communications.
- Updating of information becomes so easier.
- System security, data security and reliability are the striking features.
- The System has adequate scope for modification in future if it is necessary.

7.2 FUTURE WORK:

This application avoids the manual work and the problems concerned with it. It is an easy way to obtain the information regarding the various travel services that are present in our System.

Well I and my team member have worked hard in order to present an improved website better than the existing one's regarding the information about the various activities. Still, we found out that the project can be done in a better way. Primarily, In this system patient login and then go to reception. By using this patient will send request for consulting the doctor. Reception will set the date for doctor's appointment. After that doctor see his appointments and see the patients, surgeries also done.

REFERENCES

[1] Ivan Bayross, **SQL, PL/SQL programming language of Oracle,** Second Edition, BPB Publication.

[2] Lary Ullman, **PHP and MYSQL for dynamic websites,** fourth Edition, Quick Pro.

[3] Ramez Elmasri and Shamkant B. Navathe, **Fundamentals of Database System** (Fifth Edition), Addision Wesley.

[4] Ian Somerville, **Software Engineering**, Third Edition, Pearson Education.

[5] Luke Welling and Laura Thomson, **PHP and MYSQL Web development,** Fourth Edition, Tata McGraw Hill.