

California State University, San Bernardino

CSUSB ScholarWorks

Theses Digitization Project

John M. Pfau Library

2011

Doctor patient module

Harpreet Singh Ujjal

Follow this and additional works at: <https://scholarworks.lib.csusb.edu/etd-project>



Part of the [Databases and Information Systems Commons](#)

Recommended Citation

Ujjal, Harpreet Singh, "Doctor patient module" (2011). *Theses Digitization Project*. 3952.
<https://scholarworks.lib.csusb.edu/etd-project/3952>

This Project is brought to you for free and open access by the John M. Pfau Library at CSUSB ScholarWorks. It has been accepted for inclusion in Theses Digitization Project by an authorized administrator of CSUSB ScholarWorks. For more information, please contact scholarworks@csusb.edu.

DOCTOR PATIENT MODULE

A Project
Presented to the
Faculty of
California State University,
San Bernardino

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in
Computer Science

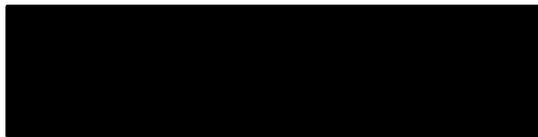
by
Harpreet Singh Ujjal
December 2011

DOCTOR PATIENT MODULE

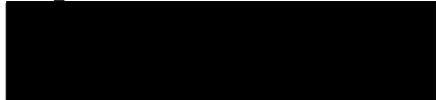
A Project
Presented to the
Faculty of
California State University,
San Bernardino

by
Harpreet Singh Ujjal
December 2011

Approved by:



Dr. R. J. Botting, Advisor, School of
Computer Science and Engineering



Dr. Ernesto Gomez



Dr. Kerstin Voigt

11/21/11

Date

ABSTRACT

The aim of this project is to design an efficient and reliable web application which would allow patients to make appointments with doctors in a certain hospital. A patient will be able to make appointments via various methods which include:

1. Online scheduling
2. via text messages
3. via phone calls

The application would also provide general information about the facilities available in the hospital, the doctors in the hospitals and their specialization [1]. The application will also have features like email alert, sms alert, schedule management with respect to doctors and patients.

The application is designed in N-Tier architecture in which each and every layer is designed such that it performs its own individual task. The application employs three layers as follows:

1. User interface layer: This layer provides user friendly interface and designed using XHTML, CSS and JAVA script to give it dynamic effects.

2. **Business Logic Layer:** This layer remains at the back of user interface and accepts user performed events and data. The data is sanitized and through the algorithms operated further using open source PHP technology.
3. **Database Layer:** This layer interacts with business logic layer and used to hold data. We have employed MySql as database which is one of the best compatible and freely available resources.

ACKNOWLEDGMENTS

I am very thankful to Dr. R. J. Botting, Dr. E. Gomez, Dr. K. Voigt and Dr. J. Mendoza for their invaluable help and assistance in the preparation of this study.

TABLE OF CONTENTS

ABSTRACT	iii
ACKNOWLEDGMENTS	v
LIST OF TABLES	ix
LIST OF FIGURES	x
CHAPTER ONE: MY PROJECT	
1.1 Introduction	1
1.2 Proposed Area of Study	4
1.3 Existing Systems	6
CHAPTER TWO: PROJECT ARCHITECTURE	
2.1 Introduction	8
2.2 Inner Architecture	10
CHAPTER THREE: TECHNOLOGY DESCRIPTIONS	
3.1 Client Technologies	23
3.1.1 Hyper Text Markup Language (HTML)	24
3.1.2 Cascading Style Sheets (CSS)	24
3.1.3 JavaScript	25
3.2 Middle-Tier Technologies	27
3.2.1 Hypertext Preprocessor (PHP)	27
3.2.2 Apache	28
3.2.3 Wamp Server	28
3.2.4 Expression Web 4 for Hypertext Preprocessor	29
3.3 Data-Tier Technologies	29

3.3.1 MySQL	30
3.3.2 Navicat	30
CHAPTER FOUR: PROJECT DESIGN	
4.1 Client Side Design	32
4.2 Data-Tier Design	41
4.3 Database Table Structure	43
4.3.1 Functionality of Tables	46
CHAPTER FIVE: PROJECT IMPLEMENTATION	48
5.1 User Implementation	50
5.1.1 Patient Screen Shots	50
5.1.2 Clerk Screenshots	57
5.1.3 Secretary Screenshots	59
5.1.4 Doctor Screenshots	62
5.1.5 Administrator Screenshots	64
5.2 Middle Tier Implementation	68
5.3 Data-Tier Implementation	69
CHAPTER SIX: TESTING	72
CHAPTER SEVEN: INSTALLATION	73
CHAPTER EIGHT: SUMMARY, CONCLUSIONS AND RECOMMENDATIONS	
8.1 Summary	76
8.2 Conclusion	77
8.3 Future Recommendations	78
APPENDIX A: NAVIGATION TREE	80

APPENDIX B: ALGORITHMS	83
REFERENCES	86

LIST OF TABLES

Table 1.	Access General Information	32
Table 2.	Patient Accessing the Appointment Portal	33
Table 3.	Patient Accessing Create Appointment and Submitting his Symptoms	34
Table 4.	Patient Accessing the Portal to Check Appointment Status	35
Table 5.	Administrator Managing General Information of Website	36
Table 6.	Administrator Managing Doctors and Patients and Granting Privileges	37
Table 7.	Administrator Assigning Doctors to Patient	38
Table 8.	Doctor Accessing Appointment Portal	39
Table 9.	Doctor Serving Appointments	40
Table 10.	Doctor Accessing Portal to Check Appointment Status	41
Table 11.	Database Table Structure	43

LIST OF FIGURES

Figure 1.	3-Tier Architecture	9
Figure 2.	Inner Architecture	11
Figure 3.	Website General Information	16
Figure 4.	Appointment Creation	18
Figure 5.	Notification Information	20
Figure 6.	Symptom Information	21
Figure 7.	Entity Relationship Diagram	42
Figure 8.	Entity Relationship Diagram Generated in Microsoft Access	46
Figure 9.	Website Homepage	51
Figure 10.	Patient's Homepage	53
Figure 11.	Patient Personal Details	53
Figure 12.	Patient's Change Password	54
Figure 13.	Patient's Security Information	54
Figure 14.	Patient's Credit Card Information	55
Figure 15.	Patient's Insurance	55
Figure 16.	Patient's Request Appointment	56
Figure 17.	Patient's My Appointments	56
Figure 18.	Clerk's Homepage	58
Figure 19.	View Pending Requests	58
Figure 20.	Process Fee	59
Figure 21.	Secretary Homepage	60
Figure 22.	View Pending Requests	61

Figure 23. Make Appointment	61
Figure 24. Doctor's Homepage	62
Figure 25. View Pending Requests	63
Figure 26. Make Prescription	63
Figure 27. Administrator's Homepage	65
Figure 28. Create Users	65
Figure 29. Manage Users	66
Figure 30. View Pending Requests	66
Figure 31. Process Fee	67
Figure 32. Create Appointment	67
Figure 33. Class Users Screenshot	68
Figure 34. Database Connection Screenshot	69
Figure 35. Wamp Server Screenshot	70
Figure 36. Database Screenshot in Navicat	71
Figure 37. Docpat Folder Screenshot	74
Figure 38. Screenshot in Microsoft Expression Web	75
Figure 39. Screenshot of Website on Local Host	75

CHAPTER ONE

MY PROJECT

This chapter is an introduction to my project. This chapter will tell us about the various steps that lead to development of this project.

1.1 Introduction

Hospitals today are not the same as they used to be a decade ago [6]. Computers have taken up nearly every department of hospitals especially the appointment scheduling system. The IT (information technology) solutions provide better services. IT has helped hospitals to create better interaction between patients and doctors especially the appointment systems in hospitals has grown in leaps and bounds which has made the life of patients a lot easier. IT has made hospitals more efficient and thus provides better services to patients.

The "doctor patient module" is a web application which can be used by hospitals to manage the general information of doctors, patients, secretaries and clerks working in the hospital together with the appointments created by the patients. The application also keeps track

of the previous appointments created by the patients and also the prescription given by the doctor. Other important features of the application include alerts through emails, public message boards, online communication where patient can chat with doctor through database.

It is a 3-Tier application which is designed by using PHP as front end and the database is implemented using MySQL.

The application is divided as follows:

1. User Interface
2. Business Logic
3. Database

1. User Interface: This layer is for end users. It is presented in the browser application and helps the users to operate the system using web forms and tabular reports. We have employed XHTML, java script, and embedded PHP code whenever required to give it a sleek look.
2. Business Logic: This layer consists of purely programming language (PHP in our case) whatever input and output is related to the end user is

managed here. This layer performs the following basic tasks.

- a. Validation
- b. Sanitization
- c. Verification
- d. Interaction with the database layer

This layer makes sure that the data going to the server database is free from any malicious code. Hence, prevents the possibility of SQL injections. It also makes sure that the data entered is consistent with the type of data chosen for database tables. For example if the type is numeric it verifies that only digits are entered and optional + or - sign are at the extreme left.

Besides this, the system also verifies other type of data before storing like e-mail addresses and credit card info that may otherwise not be handled at the database level.

On the other hand when pulling data from the database it organizes the raw data in

proper format and then presents the same to the user in tabular form.

3. Database: This layer is primarily used to hold data on the permanent basis and has inside applications. We can create objects in it that are related to data like tables, views, stored procedures, triggers etc. Further, when there are multiple operations required on data in a single instance we can use transactions to maintain integrity and consistency of data.

1.2 Proposed Area of Study

The main concern of my project is to build a web application using the newest and the latest software technologies such as PHP, MySQL, JavaScript, and XHTML. The challenges to develop a web-based application related to a hospital always need to have deep functional knowledge of working of the hospital system. For my project I required a lot of knowledge of the latest appointment systems that were available for hospitals and also the terminologies that were used in hospital appointment systems. Most of these terminologies as well as technologies and known to us but still there were some

new technologies and terminologies that I was unaware of. Another concern was to develop this project in such a technology that it should have a rich client user interface and also to make it user friendly enough so that the users operate it efficiently.

Therefore, the list of tasks that needs to be implemented is as follows:

- a. Gather knowledge about the functionality of hospitals.
- b. Knowing various terminologies used in a hospital.
- c. Understanding new versions of PHP and WAMP framework.
- d. Learning concepts of HTML and CSS for designing the framework.
- e. Understanding MySQL and the various operations of SQL.
- f. Learning the use of JavaScript.

Since, most of the servers support PHP execution, there will be no problem in getting my project live.

1.3 Existing Systems

The objective of my project is to develop a web application which makes a patient interact with the hospital in an efficient manner. The system should keep record of all the patient info like:

- a. User name
- b. Social security
- c. Insurance policy
- d. Credit card info

This type of personal info makes security a major concern. There are various hospital websites already available. These include:

<http://www.kaiserpermanente.org> [5]

<http://www.amandeephospital.org>

<http://www.siteandsmile.com>

<http://www.mumps.com>

These websites do provide a lot of information to the patient however the appointment system through these websites is still not very efficient.

Therefore as the computer technologies have advanced the hospitals need to be upgraded to these new technologies so that the hospital departments together with the appointment-scheduling department of the

hospitals become more efficient. Network and internet security are the main areas in which hospital websites want to advance. Thus, this website will provide all those latest features with the help of which objectives like network and internet security can be achieved together with the efficiency and the use of new technologies also makes the web application very user friendly.

CHAPTER TWO

PROJECT ARCHITECTURE

2.1 Introduction

The web application was developed using 3-tier architecture. The 3-tier Architecture model is divided into three layers:

- Presentation services
- Business services
- Data services

Each layer has its own particular role to play in the application as shown in the figure below:

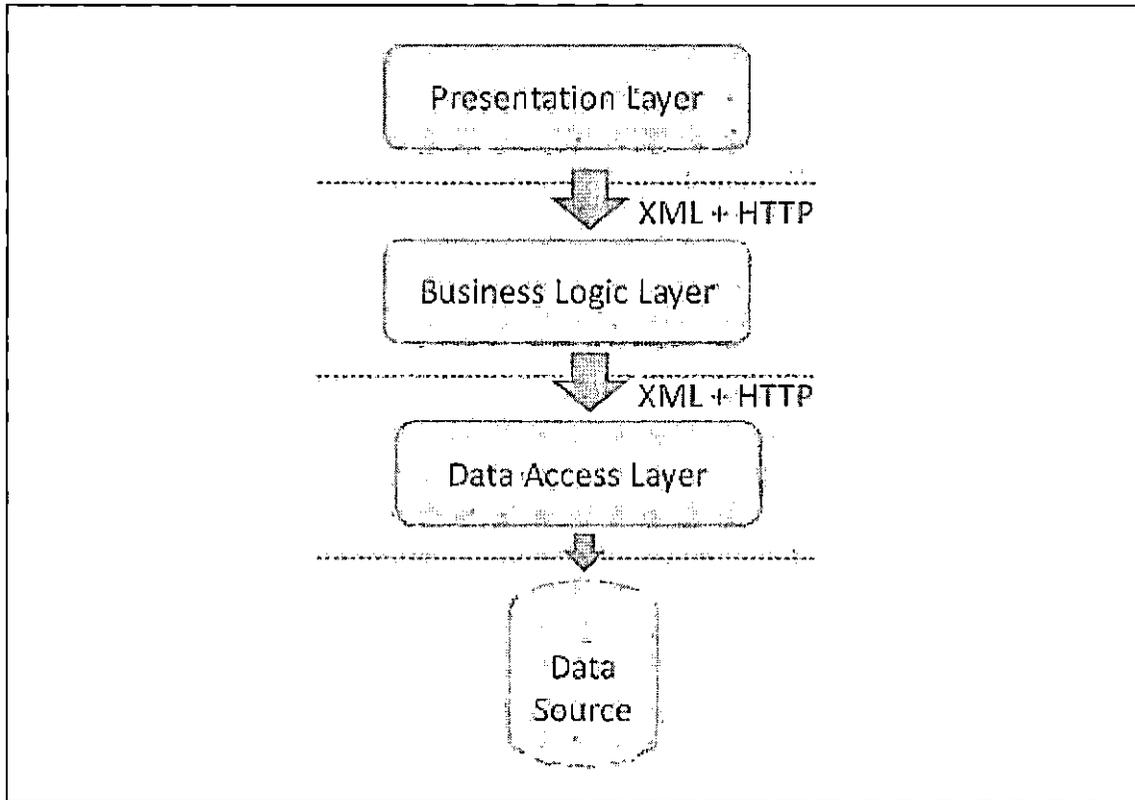


Figure 1. 3-Tier Architecture

First layer is the presentation layer which includes web browser and client side services such as HTML to interpret and display web pages and java script for user input validation and HTML for interface utility.

The business service plays a vital role and most of the work is done by it such as implementing business rules and application logic, validating data in and out of data services layer.

The last layer is data services, which perform the tasks such as storing information and managing the stored database, e-mail and directory services data.

ASP: (Active server pages) is used for building web applications. It works on the server side and act as intermediary for fulfilling the request for asp file. File contains text, HTML code and server side script. File can be saved with .asp extension. We can combine the HTML pages for creating web pages.

IIS: (Internet information service) It provides an application web development environment. It is also compatible with apache server.

COM: (component object model) it helps the components and application to operate. It is also language independent which provides freedom to the developer to use any language and tool.

2.2 Inner Architecture

The inner architecture of my project is divided into:

I. User Interface system

II. Business logic system

III. Database system

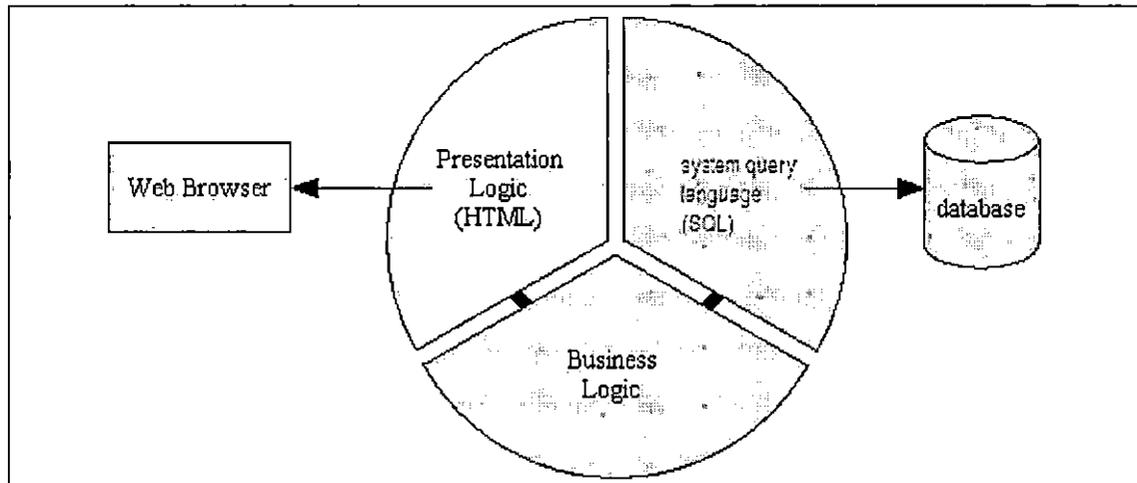


Figure 2. Inner Architecture

I User Interface:

The user system is the interface to the user and the system. The users (administrator, doctor, patient, clerk, secretary) interact with the privileges and features provided to the user. It further interacts with the Business Logic Layer according to the protocols defined in the database. Once a user logs on, his status is confirmed and then granted authentication. The identification that whether he is a administrator, doctor, patient, clerk or secretary is done by the system.

The five types of users of the system are:

- A. Administrator
- B. Doctor
- C. Patient
- D. Clerk
- E. Secretary

A. Administrator:

Admin has complete authorization of the system. The various tasks of the administrator include:

1. Updating the website with new rules and events.
2. Add a new doctor, clerk, or secretary.
3. Delete an existing doctor, clerk, or secretary.
4. Grant user privilege depending upon the account.
5. Maintains appointment schedule for doctor and patient.

B. Doctor:

1. Access appointment directory of the doctor.

2. Access the history and prescriptions of the last appointment with the doctor.
3. Check status of an appointment.
4. Alert via e-mail.

C. Patient:

The various tasks of the clerk include:

1. Create a new account
2. Access and make changes to his account.
3. Make appointments through his account.
4. Pay his fee through his account.
5. Alerts via e-mails, text messages, and phone.

D. Clerk:

The various tasks of the clerk include:

1. Access info of patient requesting for appointments.
2. Process patient fee
3. Alert secretary whether fee has been paid or not.

E. Secretary:

The various tasks of the patient include:

1. Access patient info.

2. Make appointments of the patients with the available doctor.
3. Make appointments only after receiving update from clerk that patient has paid fee.
4. Alert the doctor and patient after the appointment has been created.

II Business Logic Layer:

Business logic layer separates the user layer from the data layer in a multilayered system. This layer separates the business logic of an application from withstanding modifications or replacements of other layers. For e.g., in an application with a properly separated business logic layer and data access layer, the data access layer could be rewritten to retrieve data from a different database, without affecting any of the business logic. This practice allows software application development to be more effectively split into teams, with each team working on a different tier simultaneously.

The major components of this web application are:

- A. Web application information
- B. Appointment information

C. Notification information

D. Symptoms information

The following figures represent the various components of the web application [16]. In these figures:

- a. Oval: These represent the entities.
- b. Rectangle: It represents repository of data.
- c. Arrows: It represents dataflow.

1. Web application information consists of the general information of the hospital. It includes the information of various patients and employees of the hospital. The information of the latest news and events related to the hospital are also present in this component. It also contains the contact information of the hospital.

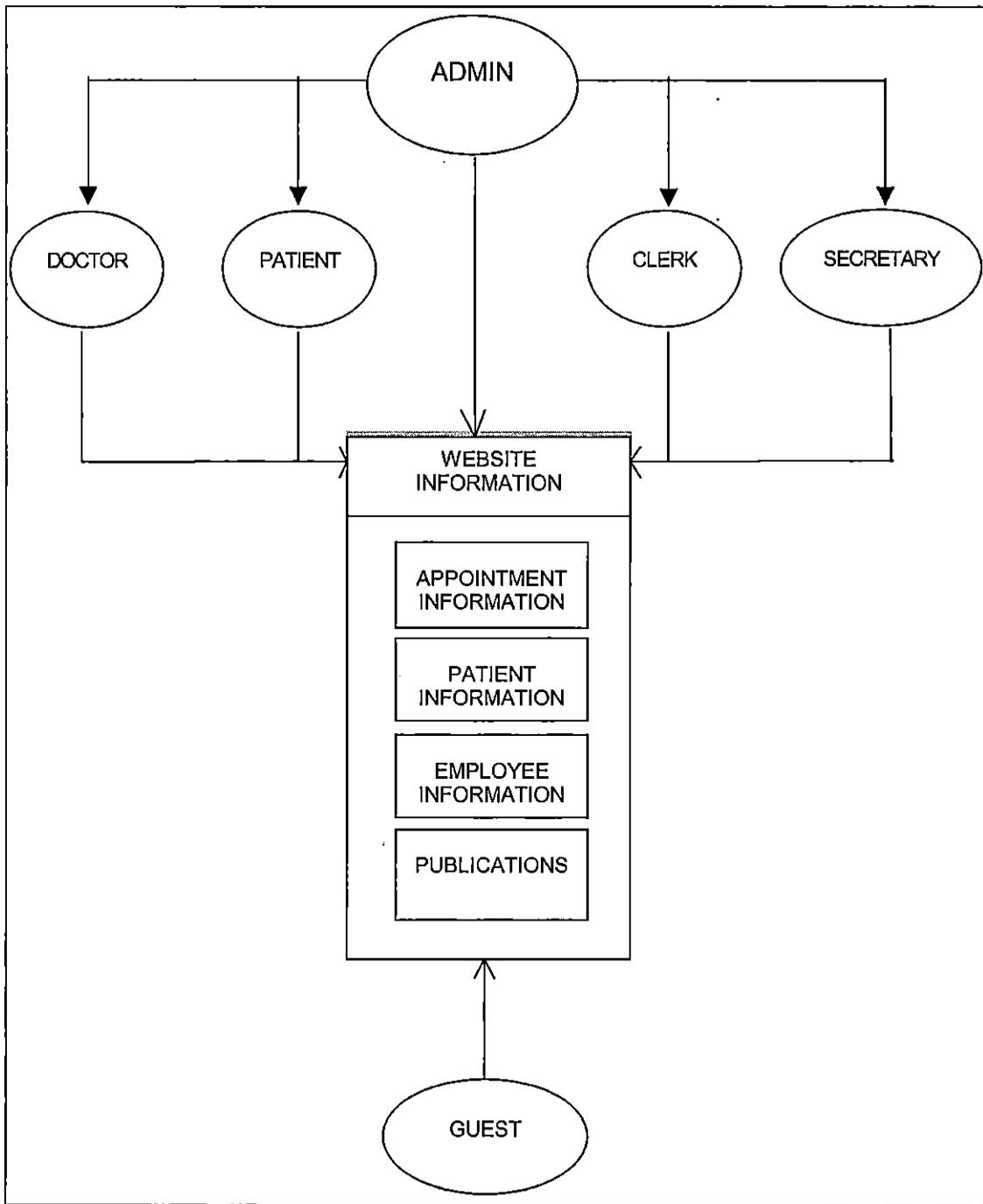


Figure 3. Website General Information

2. Appointment information: It contains all the information related to the patient and the doctor. The appointment is created by the secretary and the administrator maintains a portal which keeps track of all the appointments. The patient can track the status of his appointment and the doctor can write prescription on the pending appointment. The appointment information component diagram is as shown below:

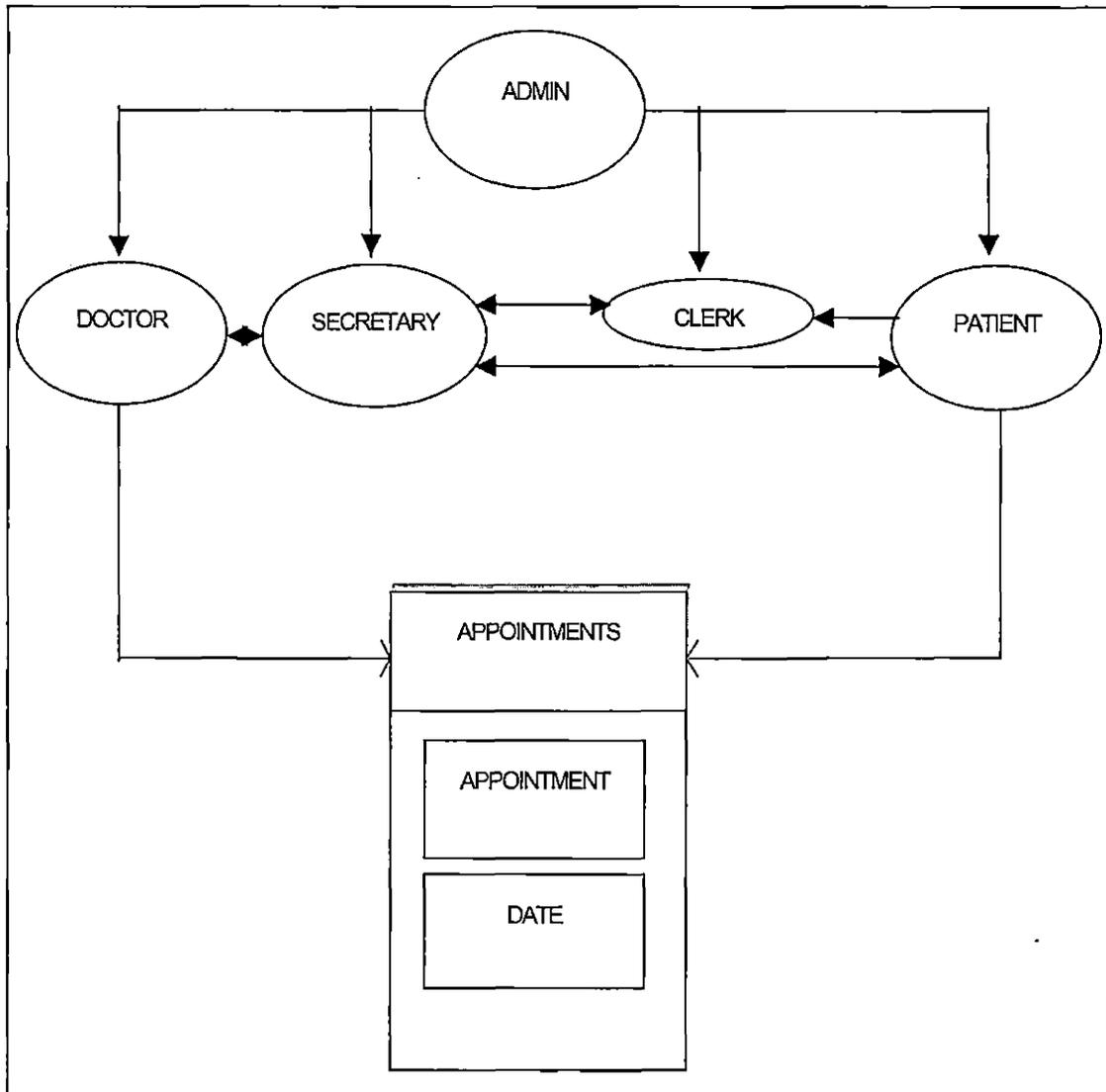


Figure 4. Appointment Creation

3. Notification information: This system notifies the doctors when patients create an appointment with them. The doctors are notified via email that an appointment has been created with them. It shows the time and the date of the

appointment. The patients are notified in the same way if the appointment cannot be created at a certain time and date. If the patient cancels or reschedules his appointment again the doctor would be notified and if the doctor cancels the appointment due to some reason the patient would be notified. The notification component diagram is as shown:

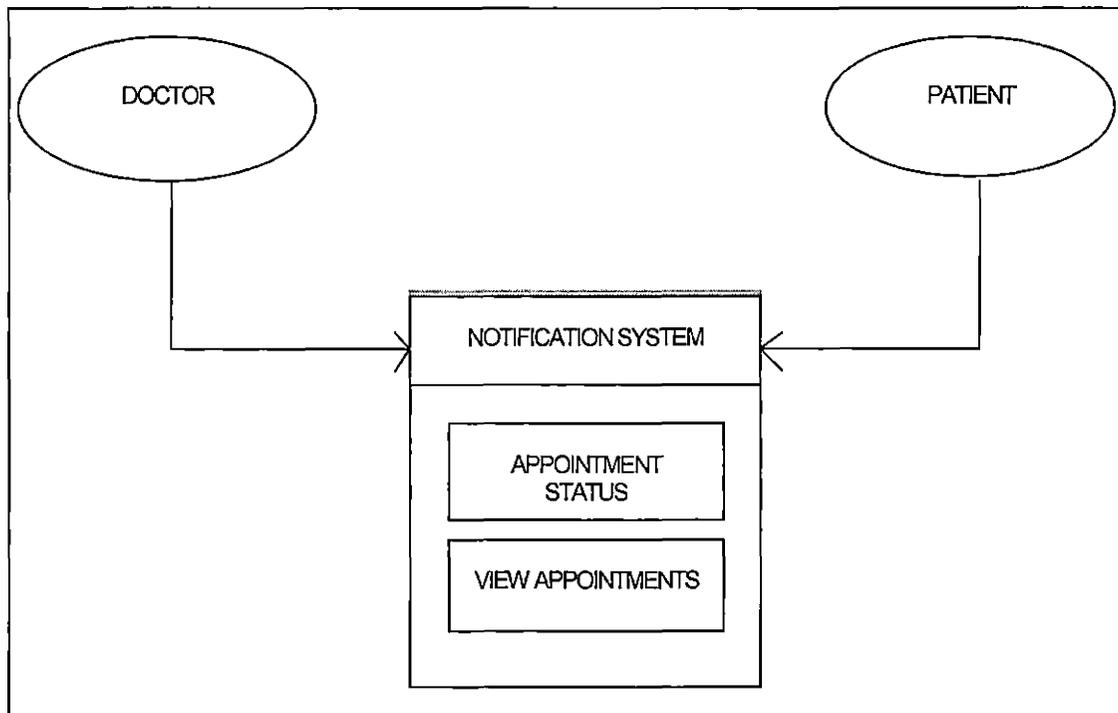


Figure 5. Notification Information

4. Symptoms information: This subsystem allows patients to tell the symptoms of their sickness to the doctor on the website itself. This process makes the treatment of the patient quicker since in this case the does not have to go see the doctor personally. In other cases if the symptoms of the patient are know earlier the doctor can refer to other books or doctors before the patient comes to see the doctor which will result in more accurate treatment. The component diagram of the symptom information is as shown:

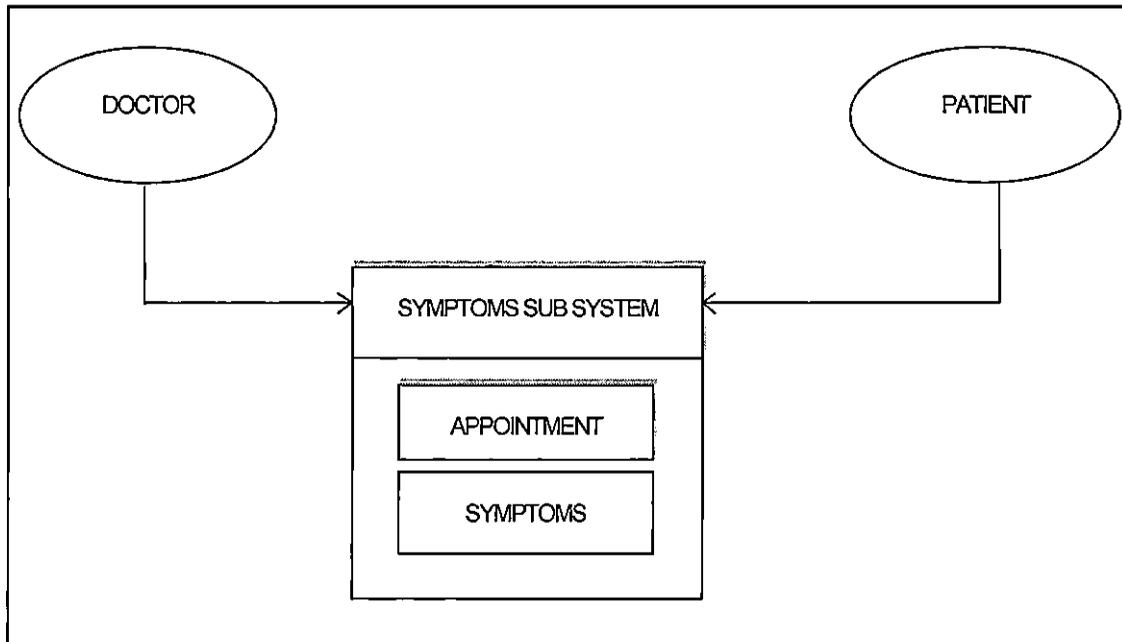


Figure 6. Symptom Information

III Database system: This system contains the collection of all the data related to the doctor patient module.

The database system thus has:

- A. Data of all the employees.
- B. Data of all the patients.
- C. Data of all the appointments.

Data of all the employees is the collection of data of all the employees of the hospital. It includes the data related to the administrators, doctors, secretaries, and clerks. Data of patients is also present in the database system. The data of all the appointments created by

patients is also present in the database
subsystem.

CHAPTER THREE

TECHNOLOGY DESCRIPTIONS

The choice of the technologies used in the project is keeping in viewpoint the advancement in technology and to make it more user-friendly. The language used in the project is PHP (Hypertext preprocessor) which is a Server-side HTML embedded scripting language. Many of HTML tags are used for making forms in the project like <frame> tag is used etc. It is compatible with Apache Server and Wamp server. It is easy to build dynamic pages with it as it can handle JavaScript also.

Wamp server (WAMP - Windows, Apache, MySQL, PHP) is a combination of all Apache, MySQL, and PHP [4]. It helps us to test our application on the local host before we test on the remote host. Once it is installed we can change its setting by configuring it. The port is also changed from 80 to 81 as browser use port 80.

3.1 Client Technologies

In order to make our project more users friendly we have implied users friendly client technologies. Following tools are used for designing the user interface

- HTML

- CSS
- JavaScript

3.1.1 Hyper Text Markup Language (HTML)

HTML file is a text file containing small markup tags [6]. The file has an extension of .html if it opens in windows. In case of dos it has an extension of .htm. The markup tags are used to tell the Web browser how to display the Web page. HTML file can be created using simple text editor. It is a static scripting language. HTML documents are text files made up of HTML elements. HTML elements are defined using HTML tags.

HTML is a common scripting language. It usually make use of various tags like <form>, <frame>, <table> etc. for designing the web pages. All the designing of the forms including the lists, tables, frames, and background colors are done through the HTML tags. It is also compatible with the languages like PHP. Moreover there is no need for specific configuration in the system to run projects.

3.1.2 Cascading Style Sheets (CSS)

Cascading Style Sheets is used to separate the presentation from the structural markup of a web site. CSS style can keep the structure of our document compact

and fast, also helps in designing the appearance of the content. CSS is a tool used for enhancing the features of HTML. The structure of the document created by HTML is static and the structure of the document created by ASP is dynamic. It has the features that can make sites faster, less complicated, easy to modify, according to the need of mushrooming technologies.

CSS is highly desirable component in the project because most of the formatting is done through CSS. We have a separate folder in our project, which has a cascading Style Sheets with page margin settings from left to right to top to bottom. We have used it in our project so that our site can run easily on different screen resolutions also. It is easily compatible with HTML also.

It is very useful tool because it allows many forms to share the formatting so if changes are done in it they are applied to all the pages, which avoid a lot of over work.

3.1.3 JavaScript

JavaScript is a scripting language that helps to create dynamic web pages. It has many keywords taken from java and its code is somewhat influenced by C language.

It also has many features of C like functions, loop etc. It also provides security as it does not provide access to secret information that is sent from one site to another. JavaScript is a case sensitive. JavaScript is event driven and is based on the concept of event handlers. Programs consist of statements that are executed by the computer. There are two varieties of JavaScript [15]:

1. Client side JavaScript

2. Server side JavaScript

1. Client - Side JavaScript statements are embedded in HTML documents which are further extracted by the browser and are executed.
2. Server Side JavaScript - are the JavaScript that runs on the server -side on the server, we need our request to be processed, fetch resources or information and manipulate them so that they can be sent back to the client. This can be done using XML, or HTML -Formatted text, which is sent across the wire using PHP, Java, and .NET are the popular server-side languages.

3.2 Middle-Tier Technologies

A middle tier technology is the gateway through which the data passes to the end user. There is a separation between the presentations, processing and data management tasks. Our project involves the following technologies as middle tier technologies:

- PHP
- Apache
- Wamp
- Expression web 4 for PHP

3.2.1 Hypertext Preprocessor (PHP)

PHP is server side scripting language which is helpful for designing dynamic and interactive websites [7]. PHP can also include HTML and Java script code which are easily interpreted by the server. It is compatible with MYSQL server also and provides a better option for updating the database dynamically. PHP provides us an option for choosing operating system like Microsoft Windows, LINUX, UNIX, and Mac OSX. It supports the web server like Apache that we have used in our project. PHP takes input from a text file and output from another file and generally, the output will be HTML.PHP mainly focuses on server side scripting, and is similar

to ASP, Java Server Pages, and mod perl. PHP also promote rapid application development.

3.2.2 Apache

Apache is a web server, which is maintained by open community of developers [10]. It can be easily downloaded and be used to test the application on the local machine. It supports the password authentication of the database. It helps in testing the database in local host. Apache comes inbuilt in a Wamp server.

Apache has the capability of resolving the errors and problems. It also acts as the virtual host. Apache web server has the features like CGL, SSL, and virtual domains. Apache needs a suitable multitasking operating system to run on. Apache uses TCP/IP protocols as its foundation, providing an implementation of HTTP. It is a reliable, efficient, and can easily extensible we server. It has free open source code.

3.2.3 Wamp Server

Wamp server provides an environment for web development using Windows [3]. It supports Apache, PHP, and MYSQL database. We can easily manage Wamp server. It has tray of icons through which it is easy to manage the web server setting with the options like restart all

services in case any of the service has been deactivated. It also allows installing the latest releases of Apache, PHP and MYSQL.

3.2.4 Expression Web 4 for Hypertext Preprocessor

Expression web 4 helps to create standard websites. It supports the web browser like Internet Explorer, Firefox with local Client and Safari with an Online services beta. We can create our WebPages using the built in tools like labels, text boxes, buttons etc. It has three tabs design, split code. It also supports PHP, Java, and HTML codes. We can also test on webpage using F6 key.

3.3 Data-Tier Technologies

The data-tier of application consists of the database that is stored on the server where database includes all the personal data as well as the authenticated information regarding the username and password. All the information stored can be retrieved any time. Therefore data integrity becomes very essential part of the database. This tier also helps to enhance performance and allows update of the database as per

need. The technologies used in the Data-Tier of Our project are as follows:

- MySQL
- Navicat

3.3.1 MySQL

My SQL is database that stores data in the form of tables [10]. It can be easily downloaded, as it is open source database. It is basically used in web applications and is compatible with apache, Wamp server, and Navicat. It can also handle php code. We have used MySQL with apache, PHP and Navicat. It is quite easy to handle code in it. It comes as a tool in Wamp server.

MySQL is a choice of database, which is used in web applications. MySQL works on many different system platforms like Mac OS X, Microsoft Windows. MySQL is an RDBMS.

3.3.2 Navicat

Navicat is a graphical database used to manage data [4]. It includes tables, views, reports, query etc. New connection can be established for both local host as well as for the remote host. It also provides the facility for backup, import, and export of data. It is compatible with

Expression web 4, wamp server, and apache also. MYSQL queries can also be implemented in this.

CHAPTER FOUR
PROJECT DESIGN

The project is divided into the following sub categories:

Client side design

Middle tier design

Data tier design

The explanation is as follows:

4.1 Client Side Design

The following table explains the client design

Table 1. Access General Information

Use Case #		Use Case 1
Use Case Name		General information access from hospital website
Use Case Actors		Guest Patient
Use Case Goals		Accessing information about the appointment such as doctors working in it and the fee.
Use Case State	Before	N/A
	After	N/A
Conditions		Anybody can get general information about hospital by surfing the web page.

Table 2. Patient Accessing the Appointment Portal

Use Case #		Use Case 2
Use Case Name		Patient accessing the appointment portal.
Use Case Actors		Patient
Use Case Goals		Patient can access information about all their appointments. List of appointments for a specific patient are shown in patient account.
Use Case State	Before	Patient not logged in.
	After	Patient logged in and can view information about all his/her appointments.
Conditions		<ol style="list-style-type: none"> 1. User with patient privileges can login to the system. 2. Patients can view information specifically related to them only.

Table 3. Patient Accessing Create Appointment and Submitting his Symptoms

Use Case #		Use Case 3
Use Case Name		Patient accessing create appointment and submitting his symptoms.
Use Case Actors		Patient
Use Case Goals		Patients can access create appointment. Patient can submit symptoms online.
Use Case State	Before	Patient writes his symptoms and creates an appointment.
	After	Above mentioned appointment is successful.
Conditions		<ol style="list-style-type: none"> 1. User with patient privileges can login to the system. 2. Patients can view information specifically related to them only.

Table 4. Patient Accessing the Portal to Check Appointment Status

Use Case #		Use Case 4
Use case Name		Patient accessing the portal to check appointment status.
Use Case Actors		Patient
Use Case Goals		Patients can access the portal and can check the status of his appointment.
Use Case State	Before	Patient not aware of appointment status.
	After	Appointment status known to Patient.
Conditions		<ol style="list-style-type: none"> 1. User with patient privileges can login to the system. 2. Patients can view information specifically related to them only.

Table 5. Administrator Managing General Information of Website

Use Case #		Use Case 6
Use Case Name		Administrator managing general information of website.
Use Case Actors		Administrator
Use Case Goals		Administrator can add, change or delete employees or patients of the website.
Use Case State	Before	Administrator makes above mentioned changes.
	After	Changes get reflected in website after saving.
Conditions		User with administrator privileges can login to the system.

Table 6. Administrator Managing Doctors and Patients and Granting Privileges

Use Case #	UC7	
Use Case Name	Administrator managing doctors and patients and granting privileges	
Use Case Actors	Administrator	
Use Case Goals	Administrator can add, change or delete information of doctors and patients. Only administrator can add doctor and patient in the system and they cannot register in the system themselves.	
Use Case State	Before	Administrator makes above mentioned changes.
	After	Changes get reflected in website after saving.
Conditions	1. User with administrator privileges can login to the system.	

Table 7. Administrator Assigning Doctors to Patient

Use Case #		UC8
Use Case Name		Administrator assigning doctors to patients.
Use Case Actors		Administrator
Use Case Goals		Administrator assigns doctors to patients to form an appointment
Use Case State	Before	Administrator makes above mentioned changes.
	After	Changes get reflected in the website after saving and get added in case directory of both doctor and patient.
Conditions		1. User with administrator privileges can login to the system.

Table 8. Doctor Accessing Appointment Portal

Use Case #		UC9
Use Case Name		Doctor accessing appointment portal.
Use Case Actors		Doctor
Use Case Goals		Doctor can access information of all the appointments to him
Use Case State	Before	Doctor not logged in
	After	Doctor logged in and can view information about all appointments made with him/her.
Conditions		<ol style="list-style-type: none"> 1. User with doctor privileges can login to the system. 2. Doctors can view information specifically related to them only.

Table 9. Doctor Serving Appointments

Use Case #		UC10
Use Case Name		Doctor serving appointments.
Use Case Actors		Doctor
Use Case Goals		Doctor writes prescription according to the symptoms.
Use Case State	Before	Doctor makes above mentioned changes
	After	Above mentioned changes are successful.
Conditions		<ol style="list-style-type: none"> 1. User with doctor privileges can login to the system. 2. Doctor can view update information of cases assigned to them only.

Table 10. Doctor Accessing Portal to Check Appointment Status

Use Case #		UC11
Use Case Name		Doctor accessing the portal to check appointment status
Use Case Actors		Doctor
Use Case Goals		Doctor can access the portal and check the status of the appointment.
Use Case State	Before	Doctor not aware of appointment status
	After	Appointment status known to Doctor
Conditions		<ol style="list-style-type: none"> 1. User with doctor privileges can login to the system. 2. Doctors can view information of case specifically assigned to them only.

4.2 Data-Tier Design

ER diagram: ER diagram or entity relationship

diagram is a graphical representation of the entities and their relationship. The various kinds of symbols used are:

Rectangle: These represent entity set.

Diamond: These represent entity relationship.

Lines: These link the attribute of an entity and also link the relationship between two or more entity sets.

Ovals: These represent attribute.

The following ERD is designed for our project.

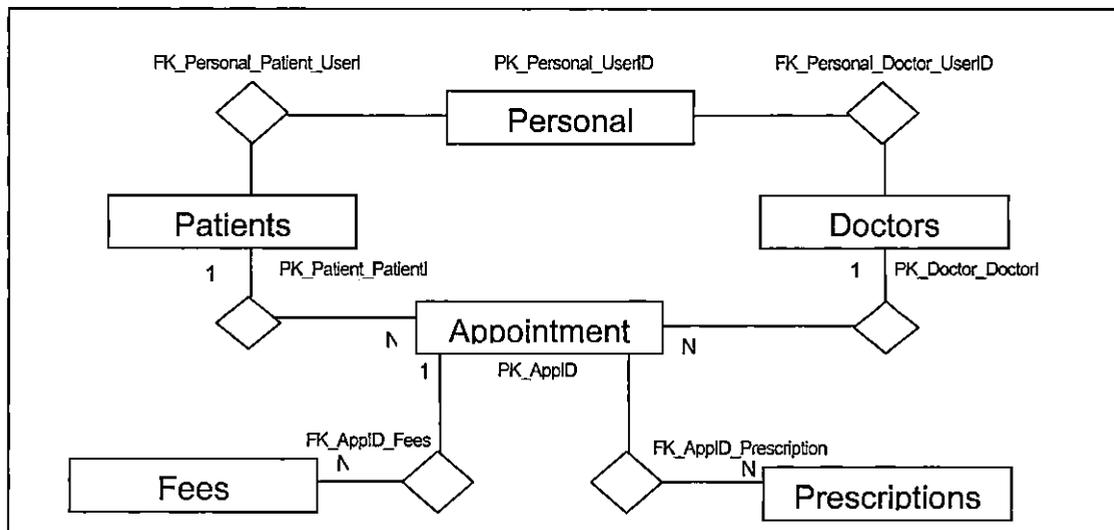


Figure 7. Entity Relationship Diagram

4.3 Database Table Structure

Table 11. Database Table Structure

Personal Details						
Field Name	Type	Length	Constraint	Format	Remarks	
Login Information						
User ID	@ N		PRIMARY			
Email ID	* A/N	50	UNIQUE			
Username	* A/N	25	UNIQUE			
Password	* A/N	25				
Sec Que	* N				INDEXED 1, 2, 3, 4...	
Sec Ans	* A/N	100				
Role ID	* N	1			1-Patient, 2-Secretary, 3-Clerk, 4-Doctor	
Status	* N				1-ACTIVE, 0-BLOCKED	
SSN	* N	9				
Personal Information						
First Name	* A	35	Not Null			
Last Name	# A	35	Not Null			
DOB	# DATE	10		MM/DD/YYYY		
H.No	* A/N	10				
City	* A	35				
State	* A	35				
Zip code	* N	5				
Phone	# N	10				
Mobile	* N	10	Not Null			

Credit Card Information					
CCNo	\$	N	16	UNIQUE	Credit card Number
Owner	\$	A	35	Not Null	
CCType	\$	N	1	Foreign Key	From CCMaster
Date of Expiry	\$	DATE	10		MM/DD/YYYY
Date of Issue	\$	DATE	10		MM/DD/YYYY
CVV No	\$	N	3		
CCType	\$	N	1	Default-1	1-VISA, 2 MASTER CARD
Insurance Policy Details (Patients)					
Policy No	\$	A/N	20		
Company	\$	A	100		
Insurance Type	\$	N	1	Foreign Key	From Insurance Master
Expiry Date	\$	DATE	10		MM/DD/YYYY
Date Issue	\$	DATE	10		MM/DD/YYYY
Doctor Details					
User ID	*	N		Foreign Key	From Users
Doctor ID	@	N		Primary Key (Auto)	
Specialty	*	A/N	100		
DOJ	*	DATE	10		MM/DD/YYYY
Patient Details					
User ID	*	N		Foreign Key	From Users
Patient ID	@	N		Primary (Auto)	
Disease	*	A/N	100		
Date Reported	*	DATE	10		MM/DD/YYYY

Appointments						
App ID	@	N				
Secretary ID	*	N		Foreign Key	From Users	
Patient ID	*	N		Foreign Key	From Patients	
Doctor ID	*	N		Foreign Key	From Doctors	
Date of Appointment	*	DATE	10		MM/DD/YYYY	
Remarks	#	A/N	1000			
Status	*	N	1			0 On Hold, 1 Finished
App Type	*	N	1			0-Message, 1-Email, 2-Phone, 3-Emergency
Fees						
App ID	*	N				
Amount	*	N	(10,2)			
Pay Mode	*	N	1			Indexed: 1-Cash, 2 CC, 3 Insurance, 4 Other
Date Paid	*	DATE			MM/DD/YYYY	
Receipt No	*	N				
Reference	*	A/N	25			
Prescription						
App ID	*	N		Foreign Key		From Appointments
Prescription	*	A/N	1000			
Remarks	#	A/N	1000			

Legend:

- * Entry is Mandatory
- # Entry id Optional
- @ Automatically Generated
- \$ Entry Mixed (Must for Patients, Optional for Doctors)

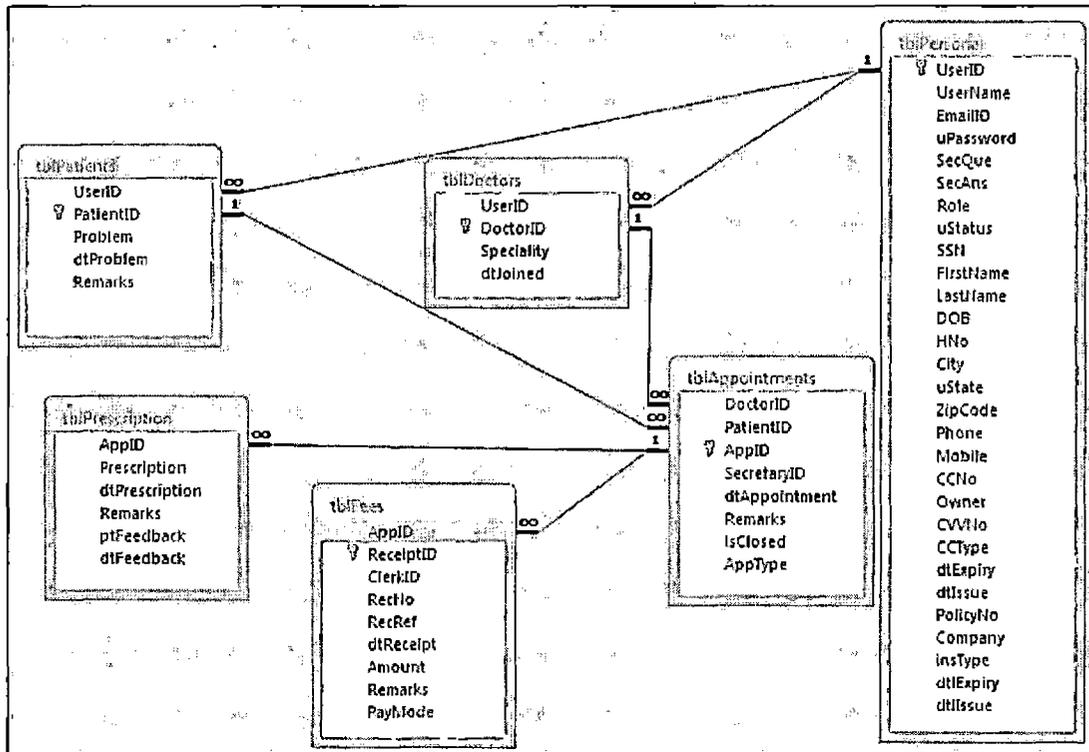


Figure 8. Entity Relationship Diagram Generated in Microsoft Access

4.3.1 Functionality of Tables

Personal details: This table further has tables like,

Login information: This table is used to store the

personal details like user ID, email ID, User name, Password, Sec Que, Sec Ans, the Role ID, Status, and the SSN of the employee or the patient.

Personal information: This table is used to store first name, last name, DOB, house#, city, state, zip code, phone#, cell#.

Credit card information: This table is used to store CC no., owner, cc type, date of expiry, date of issue, cvv no.

Insurance policy details (patients): This table is used to store policy no., company, insurance type, expiry date, date of issue.

Doctor details: This table is used to store User IDS, Doctor ID, specialty, DOJ (date of joining).

Patient details: This table is used to store User ID, Patient ID, Disease, date reported.

Appointment details: This table is used to store App ID, Secretary ID, Patient ID, Doctor ID, Date of Appointment, Remarks, Status, and Appointment Type.

Fee details: This table is used to store App ID, Amount, Pay mode, date paid, Receipt no., Reference.

Prescription details: This table is used to store App ID, prescription, Remarks.

Navigation tree: refer appendix navigation tree

Algorithms: refer appendix algorithms

CHAPTER FIVE

PROJECT IMPLEMENTATION

The main purpose of this project is to develop a web application that allows patients to interact with doctors through appointments. The main focus of the project is to make the appointment system more efficient in terms of appointments and these appointments can be made via text, email, or in person. Live chat with doctor through database is also another feature of this web application. A portal is also maintained with keeps track of all the appointments.

All requirements like appointment status, online submission of symptoms related to a particular appointment, selecting the date and time for an appointment, emergency appointment in case of emergency have been implemented.

There are five kinds of users in this project and these are:

Administrator

Patient

Clerk

Secretary

Doctor

Administrator has complete control of the system and he can create and delete patients and employees whenever required. Once the administrator creates the patient and the employees they can be linked together. The patient and doctor can be linked together with the help of the clerk and the secretary. The patient would request for an appointment and can also submit his symptoms and choose a certain doctor. The clerk would then process his fee and let the clerk know that the patient has paid his fee and has asked for a certain date of appointment. The secretary would then check the doctor's schedule and thus confirm the appointment. The doctor can see all the appointments in the portal and the patient on the other hand can check the status of his appointment.

The implementation of the project is done using PHP, MySQL, HTML, CSS, and java script. MS expression is used for developing PHP pages and Navicat is used for SQL server, which is a powerful database administration and development tool for Microsoft SQL server. Navicat well designed graphical user interface (GUI) lets you quickly and easily create, organize access and share information in a secure and easy way, talking SQL server

administration. WAMP web server package was used to host websites on local host. WAMP consists of windows, Apache, Mysql, PHP. Apache web server was used to host websites on local host and to implement the database WAMP's Myself service was used.

5.1 User Implementation

The various users with which we create the client implementation PHP pages are:

1. Patient
2. Clerk
3. Secretary
4. Doctor
5. Administrator

The following screen shots explain the functionality

5.1.1 Patient Screen Shots

Home Page. The web page appears when an administrator, employee, patient or a guest enters the website. This page has a register link where a patient can register. It has a login link where the administrator, patient, or the employee can login and are directed to their personal accounts. Guests can also view

the hospital information and the contact us information on the home page.



Figure 9. Website Homepage

Patient Home. This page appears once the patient's login is successful. The patient has various links in his homepage which include

1. Personal details: This allows a patient to view or change his personal details.
2. Change password: This allows a patient to change his password.
3. Change sec. info: This allows a patient to change his security information.

4. Credit card info: This allows a patient to fill up his credit card information.
5. Insurance: This allows a patient to list his insurance details.
6. Request App: This allows a patient to request for an appointment and also allows the patient to select a date and time. It also lets a patient choose the kind of appointment he wants and the type of payment mode that he would want to choose. It also allows a patient to choose a doctor for appointment.
7. My Appointment: This allows a patient view the appointments that he has created.

Logout

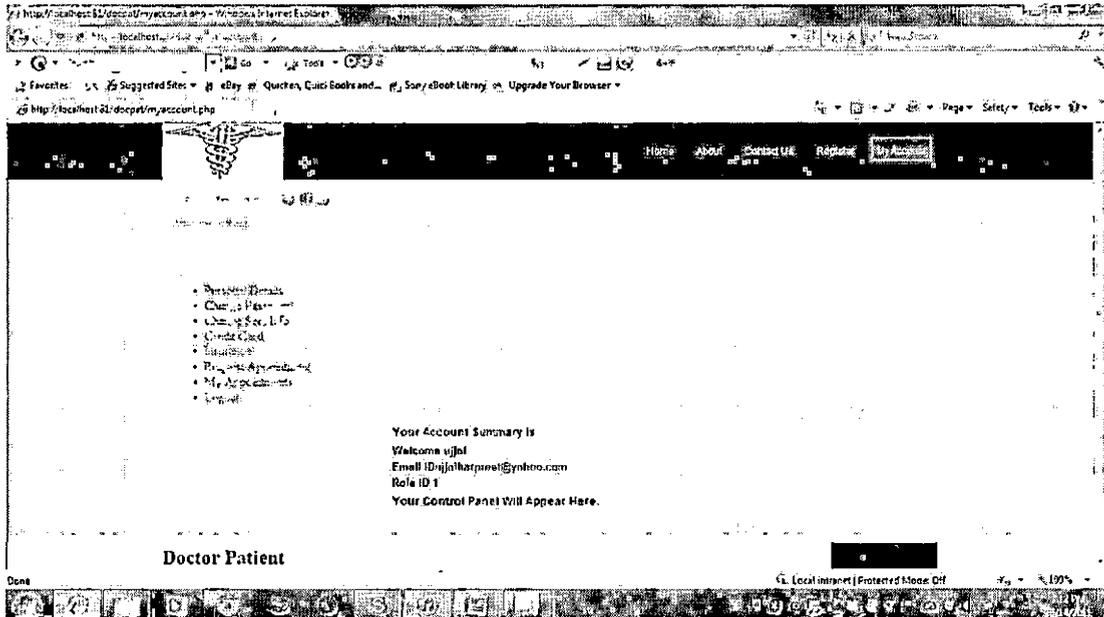


Figure 10. Patient's Homepage

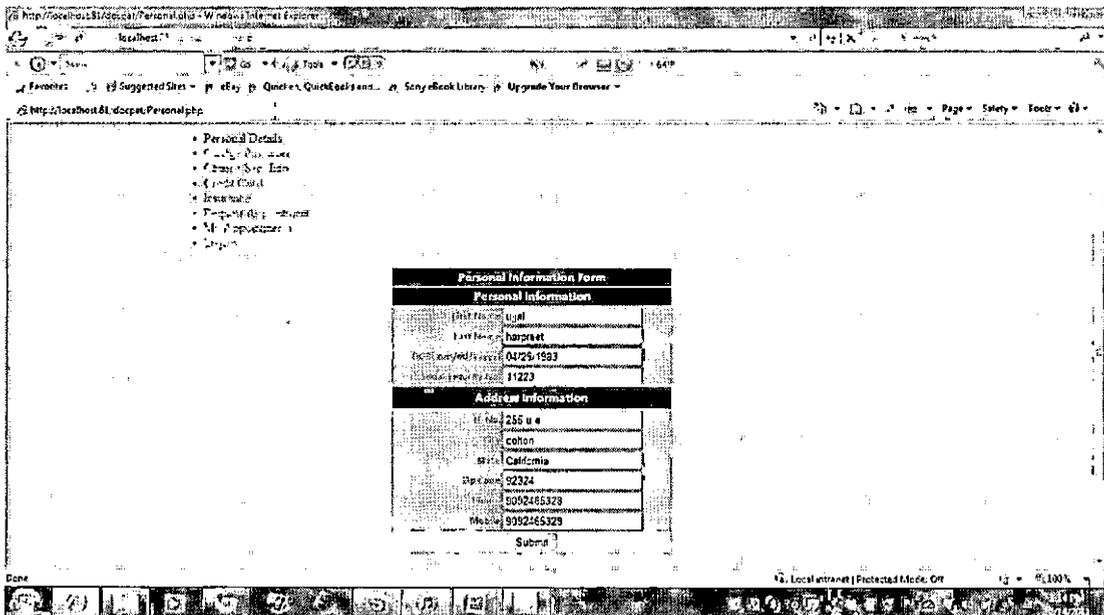


Figure 11. Patient Personal Details

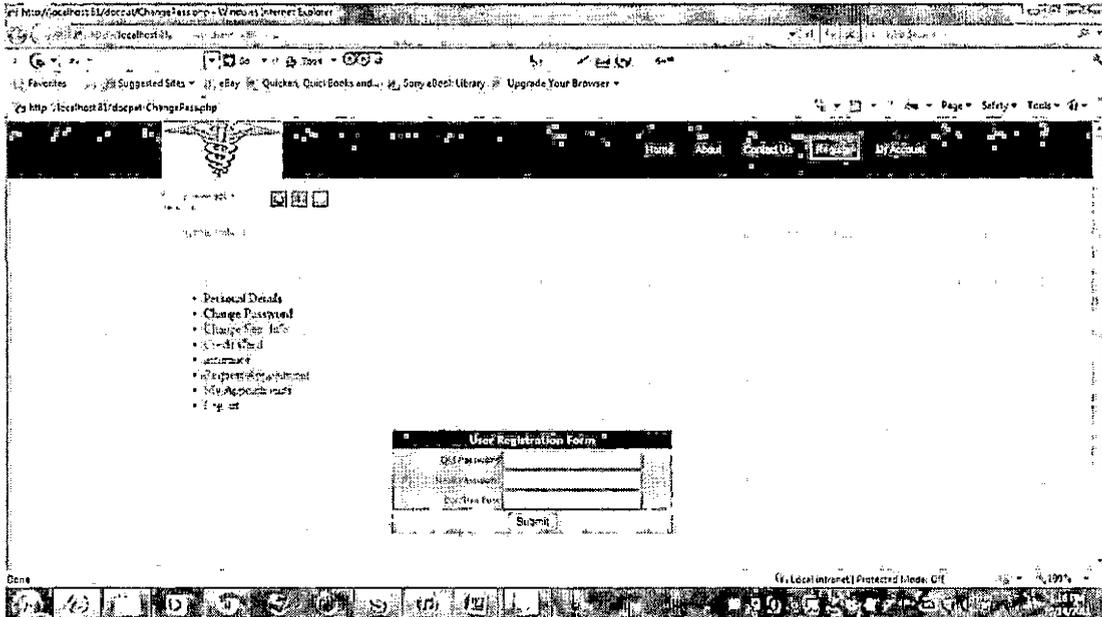


Figure 12. Patient's Change Password

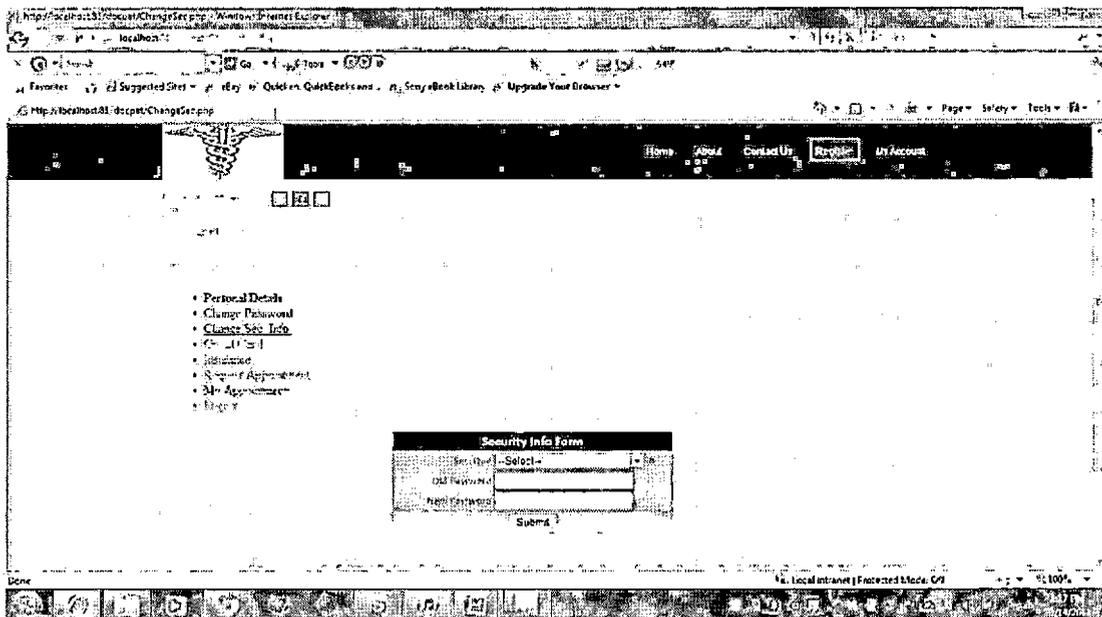


Figure 13. Patient's Security Information

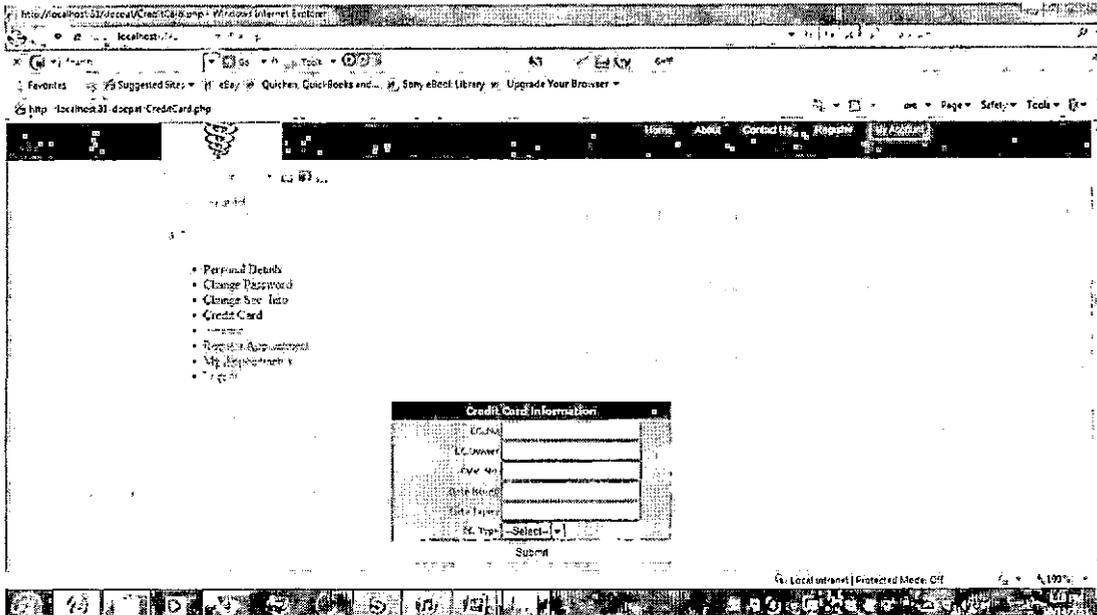


Figure 14. Patient's Credit Card Information

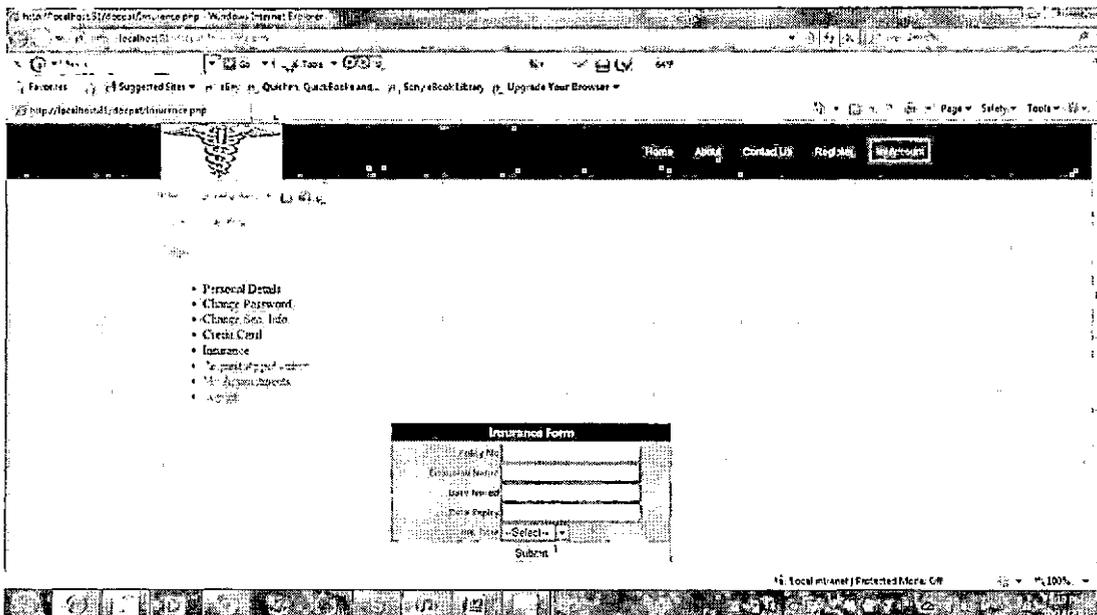


Figure 15. Patient's Insurance

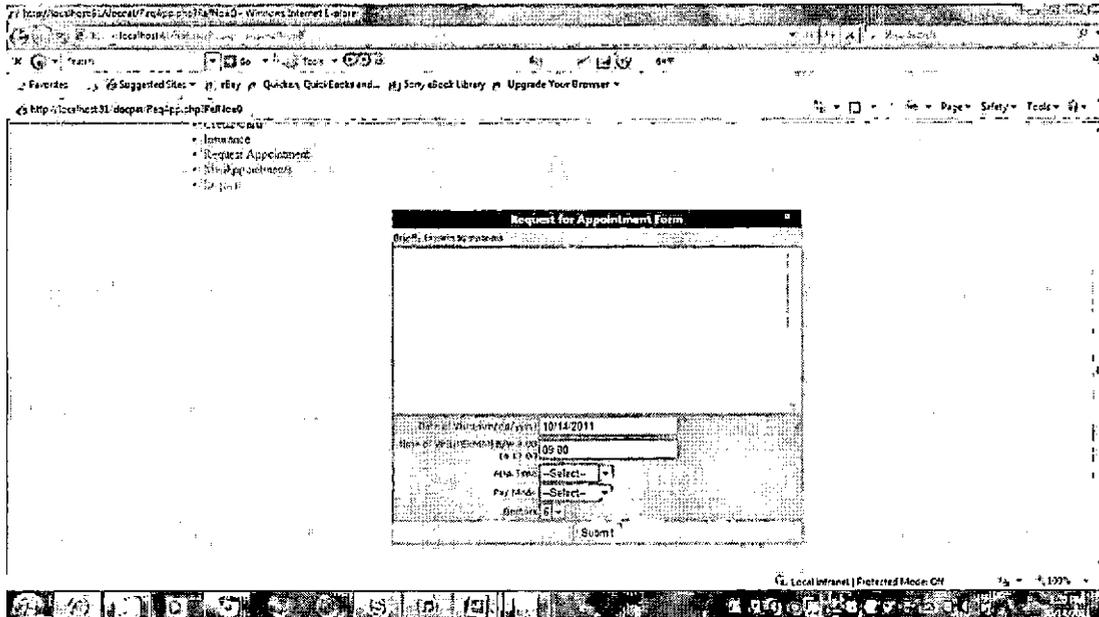


Figure 16. Patient's Request Appointment

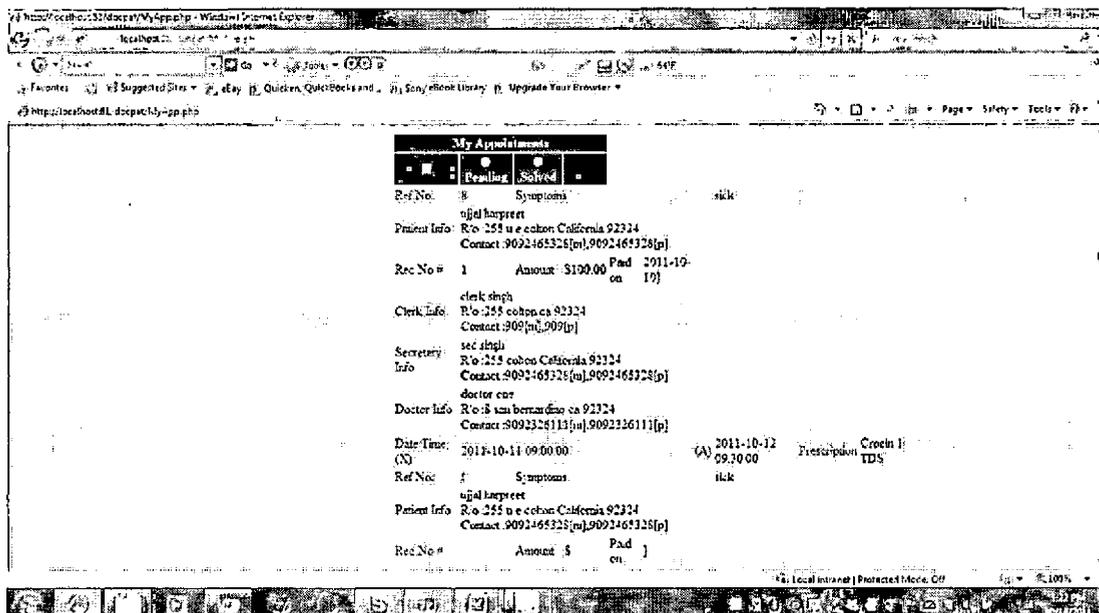


Figure 17. Patient's My Appointments

5.1.2 Clerk Screenshots

Clerk Homepage. The clerk's homepage appears once the clerk logs in successfully. The clerk's homepage will have all the links present in the patient's homepage except request appointment and my appointments. In the clerk's homepage these pages are replaced by

1. View pending requests: This allows the clerk to see the request for appointments by the patients that are pending.
2. Process fees: This allows the clerk to process the fee of the patient asking for appointment and if it is a success the information is given to the secretary so that the secretary can confirm the patient's appointment.

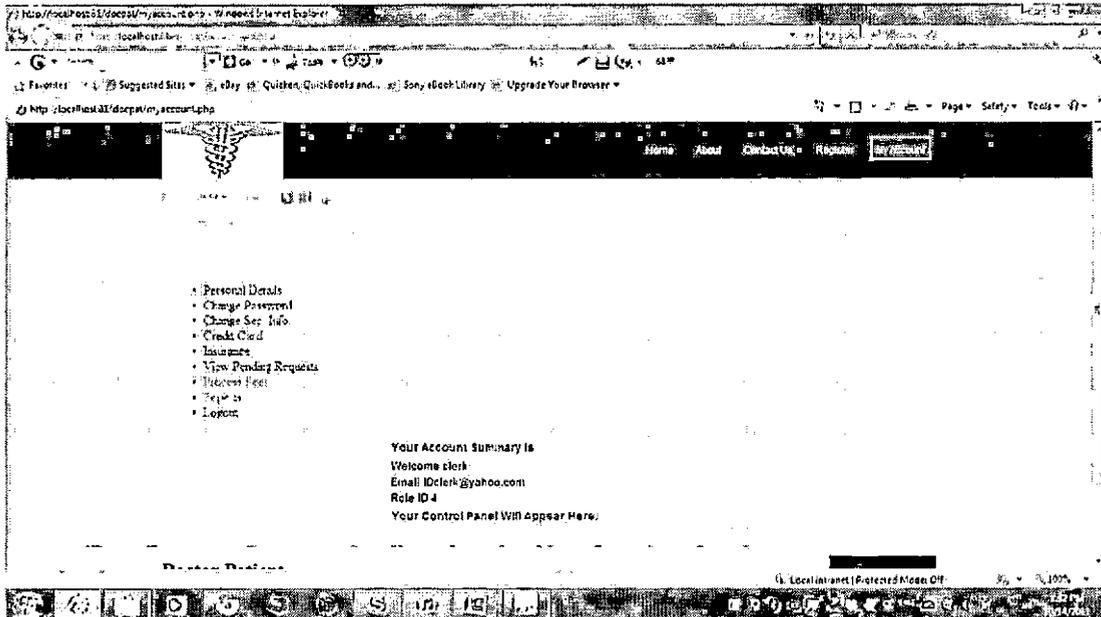


Figure 18. Clerk's Homepage

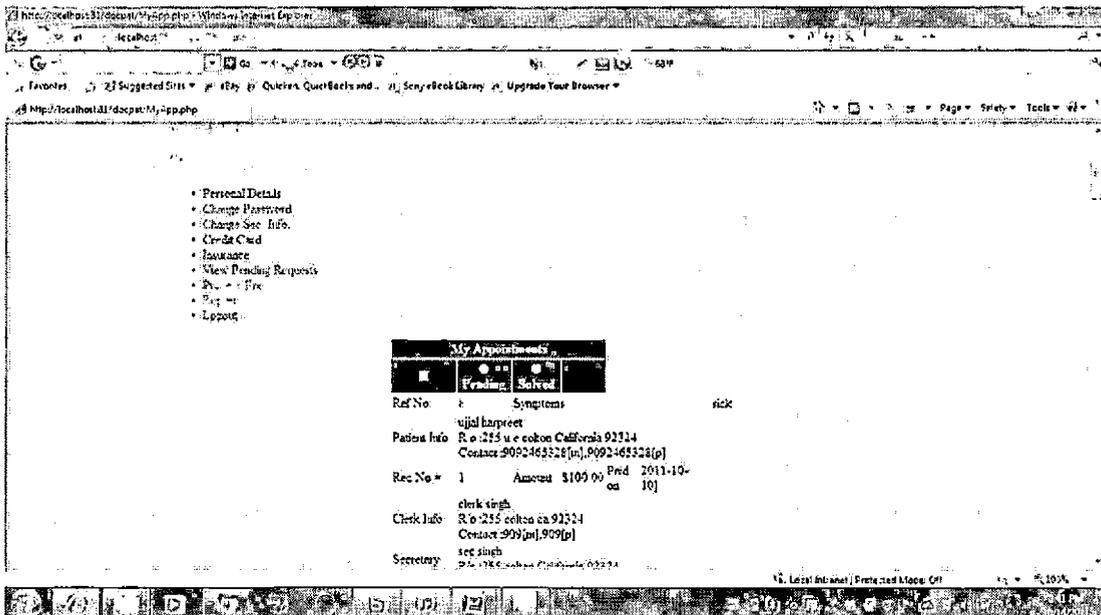


Figure 19. View Pending Requests

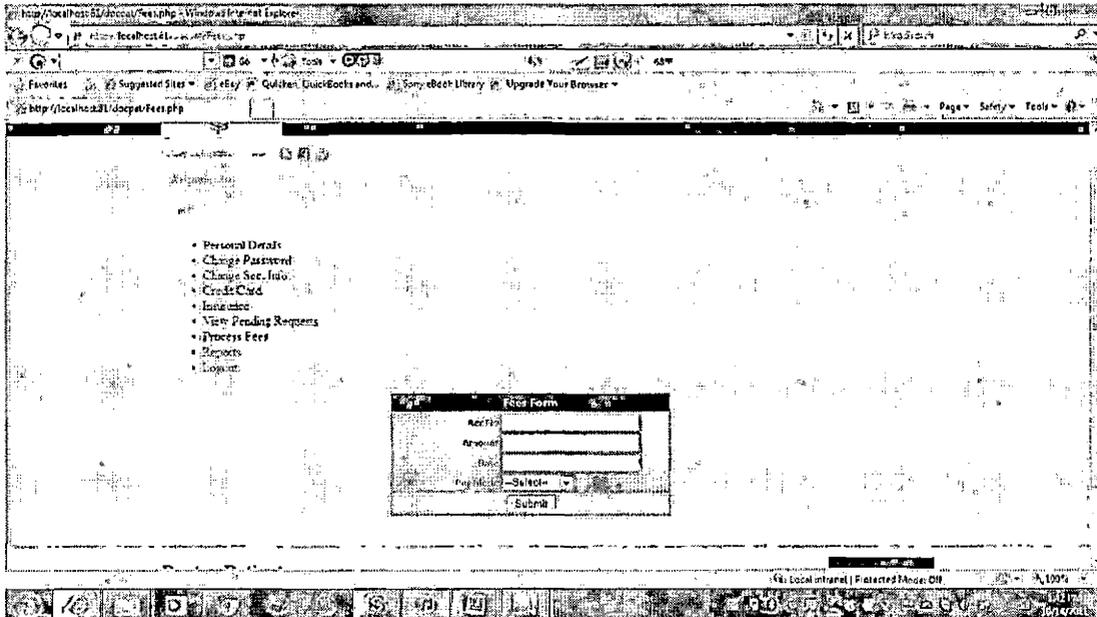


Figure 20. Process Fee

5.1.3 Secretary Screenshots

Secretary Homepage. The secretary's homepage will appear once the secretary logs in successfully. The secretary home page has all links present in the patient's homepage except request appointment and my appointment. These pages are replaced by:

1. View pending requests: This page shows the various appointments created by patients. It also shows which patients have paid their fee and which have yet to pay.
2. Make appointment: This page allows the secretary to confirm appointments of the

patients who have paid their fee. The secretary can also modify the time and date of the appointment if needed.

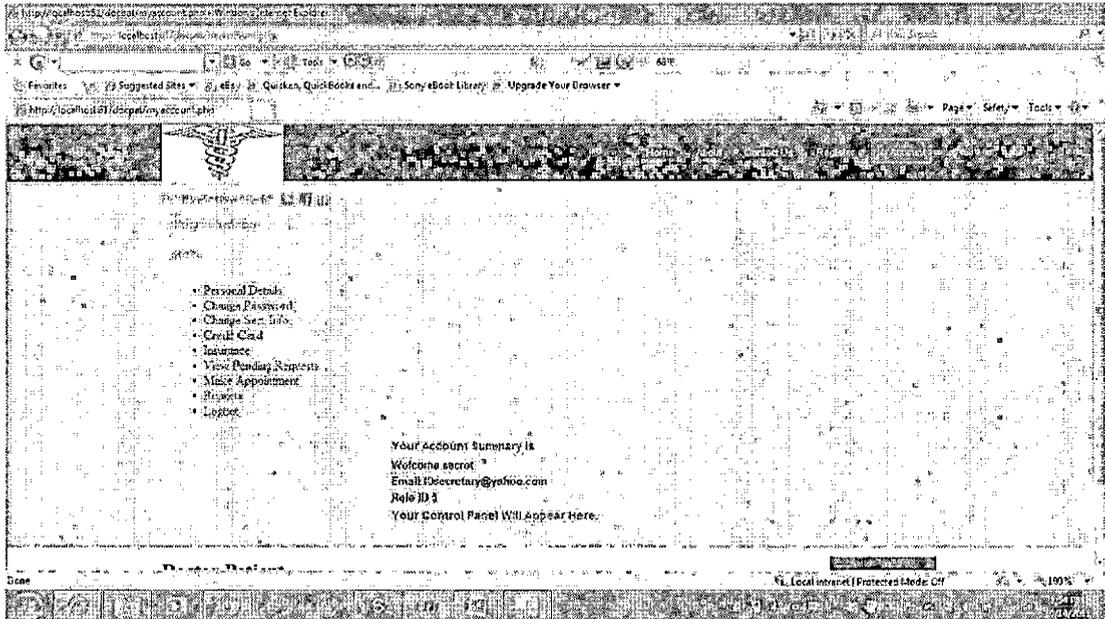


Figure 21. Secretary Homepage

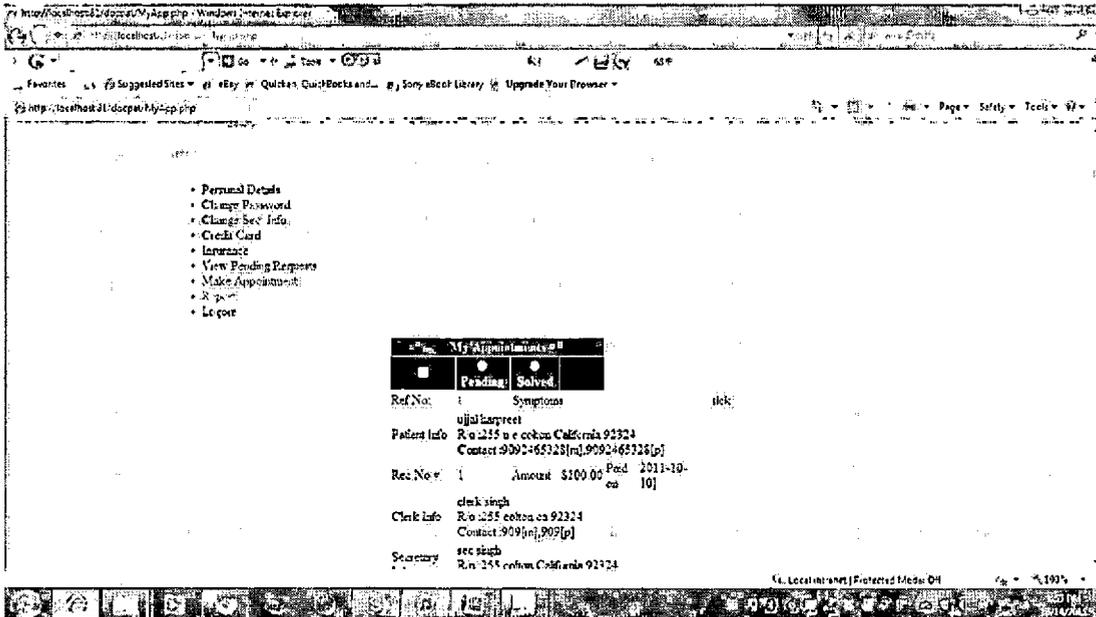


Figure 22. View Pending Requests

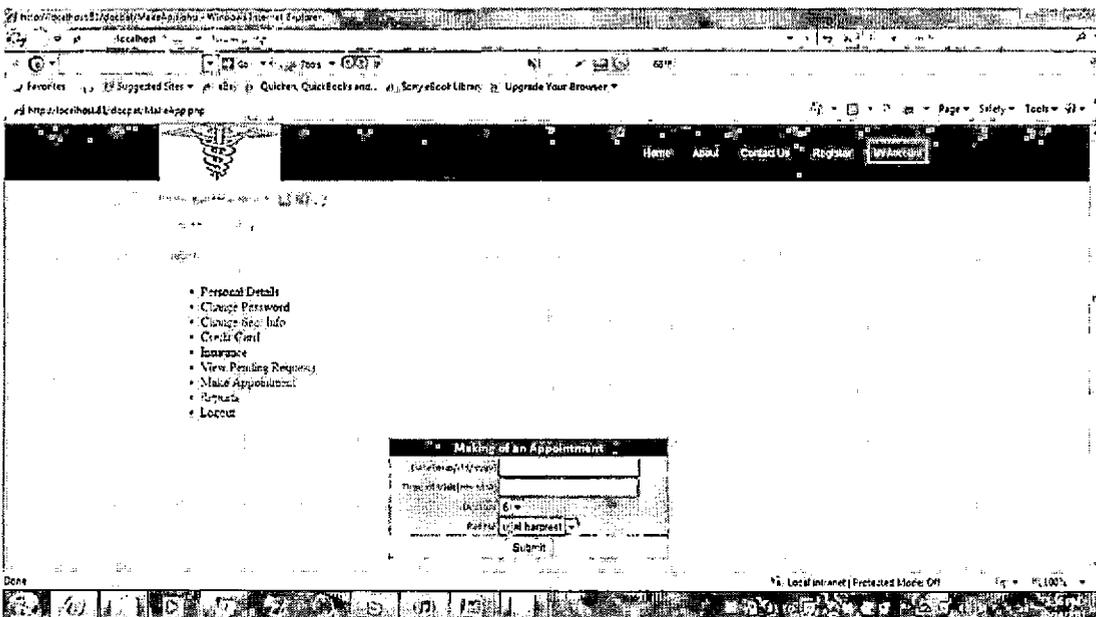


Figure 23. Make Appointment

5.1.4 Doctor Screenshots

Doctor Homepage. The doctor's homepage has the same links as the patient's homepage except request appointment and my appointments. In the doctor's homepage these links are replaced by:

1. View pending requests: This shows the various appointment requests that have been created by patients.
2. Make prescription: This allows the doctor to write prescription to the patient who has paid his fee and thus the appointment is closed.

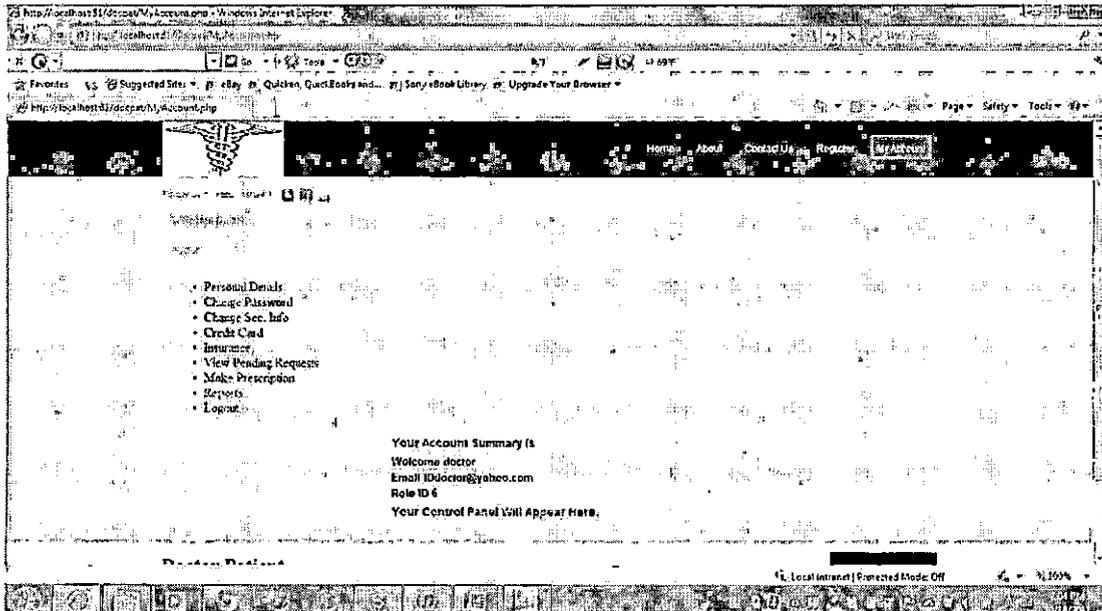


Figure 24. Doctor's Homepage

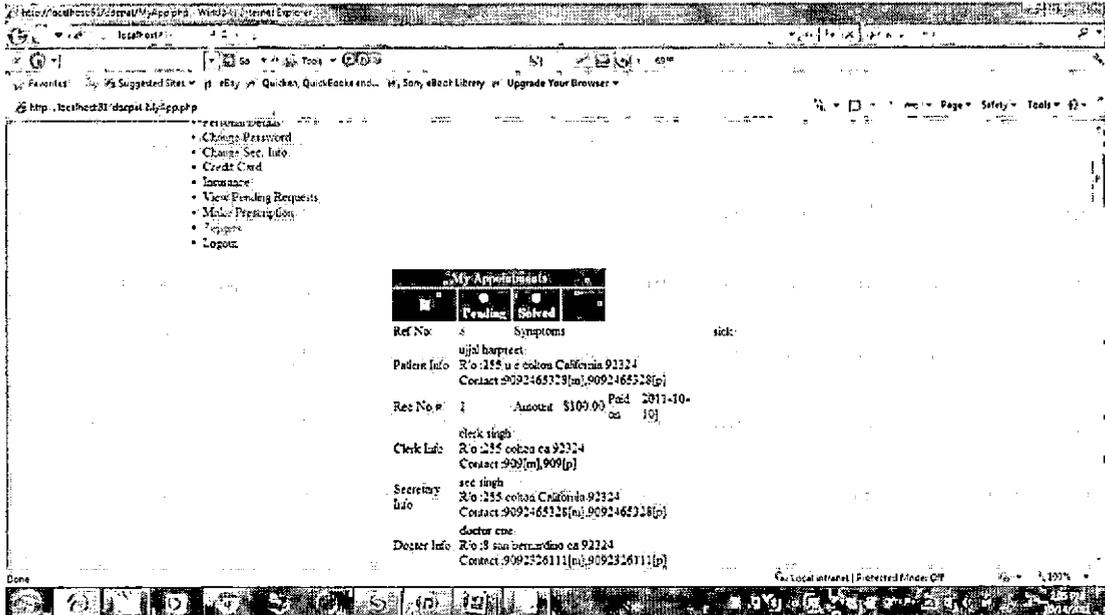


Figure 25. View Pending Requests

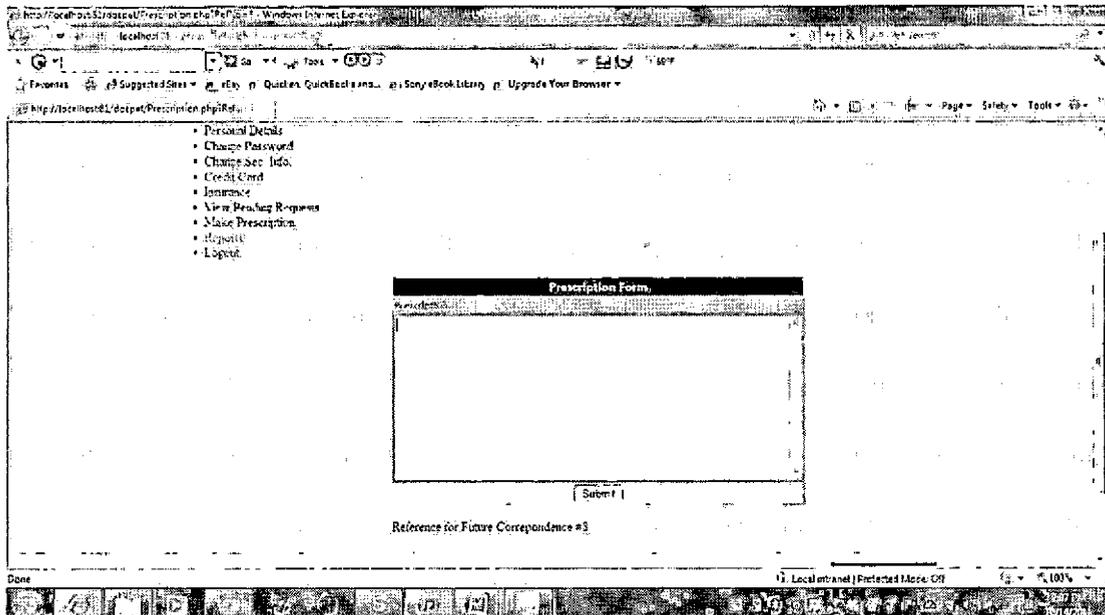


Figure 26. Make Prescription

5.1.5 Administrator Screenshots

Administrator Homepage. Administrator's homepage contains all links present in the patient's homepage except request appointment and my appointments. The new links that are also present in the administrator's homepage are:

1. Create users: This link is used by the administrator to create new users depending on their role. The administrator can create an employee or a patient using this link.
2. Manage users: This link is used by the administrator to delete the employees or patients. It also tells the administrator as to how many employees and patients are there in the system together with their roles.
3. View pending requests: This link is used by the administrator to view the patients asking for appointment.
4. Process fee: This link is used by the administrator to process fee of a patient.
5. Make appointment: administrator can confirm the appointment of a patient using this link.

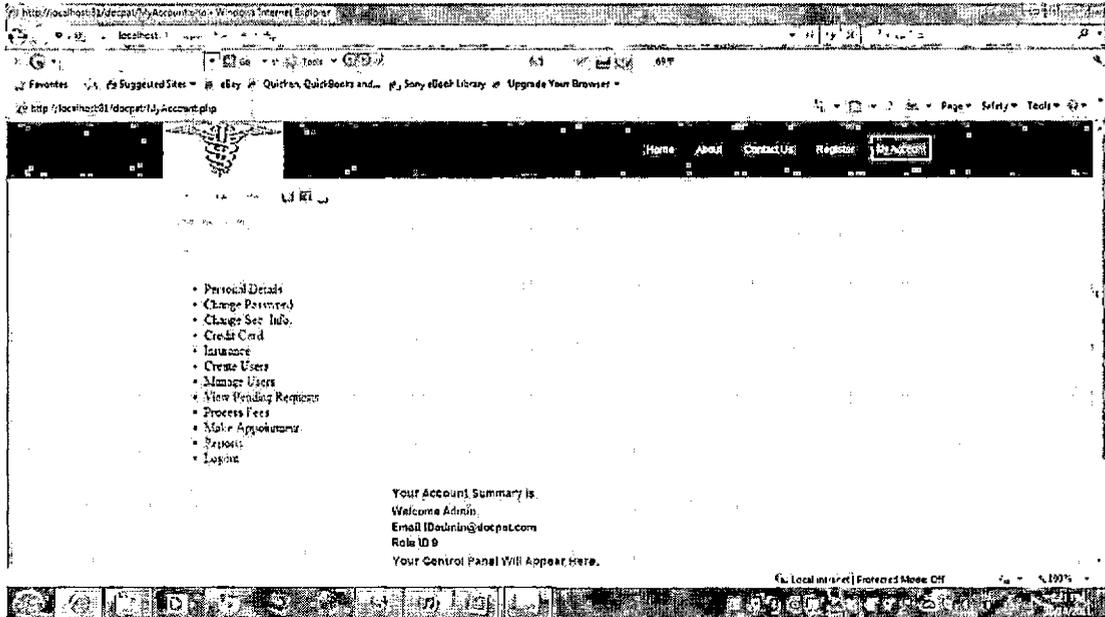


Figure 27. Administrator's Homepage

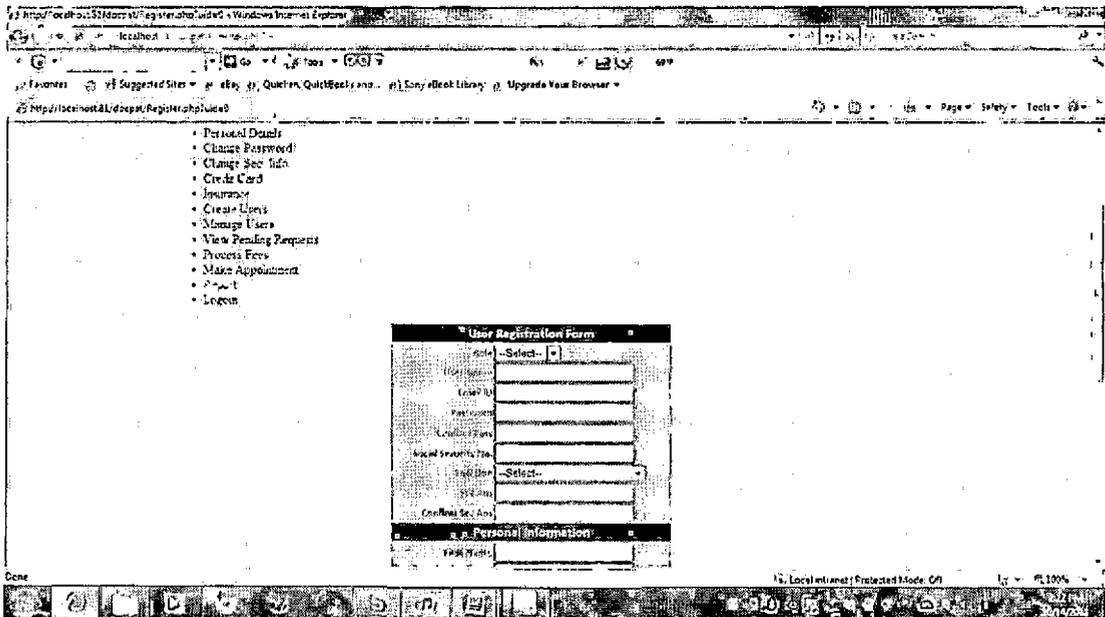


Figure 28. Create Users

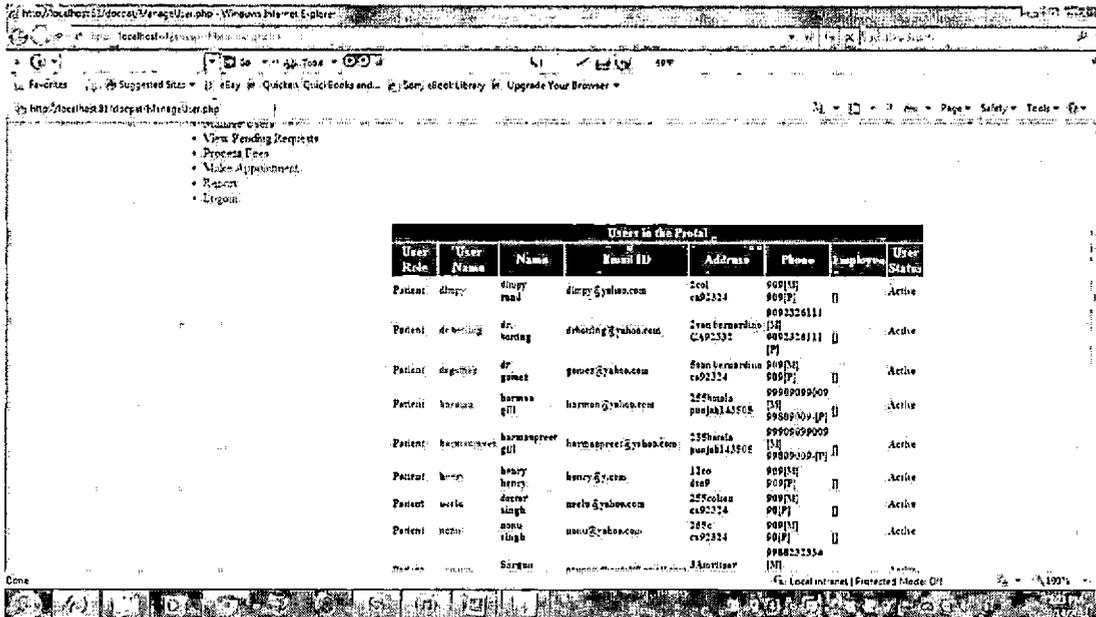


Figure 29. Manage Users

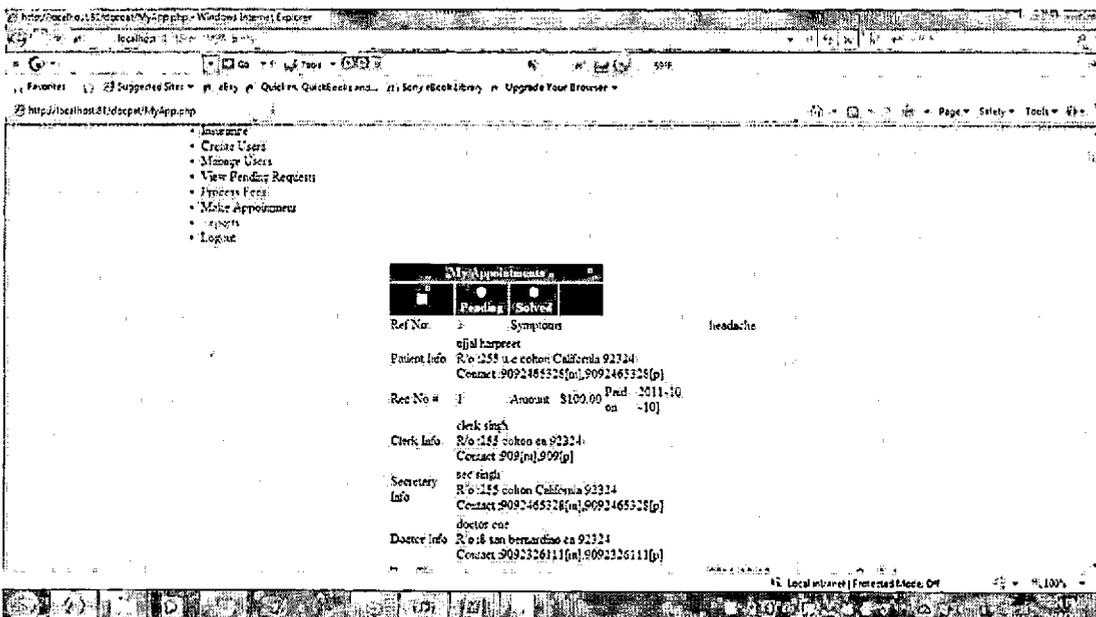


Figure 30. View Pending Requests

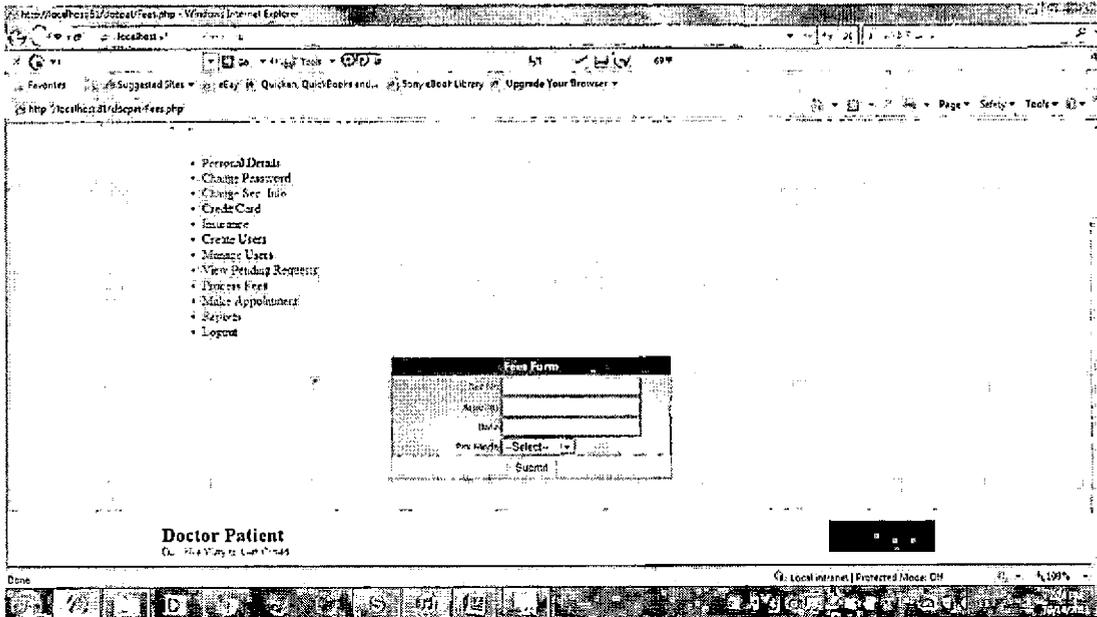


Figure 31. Process Fee

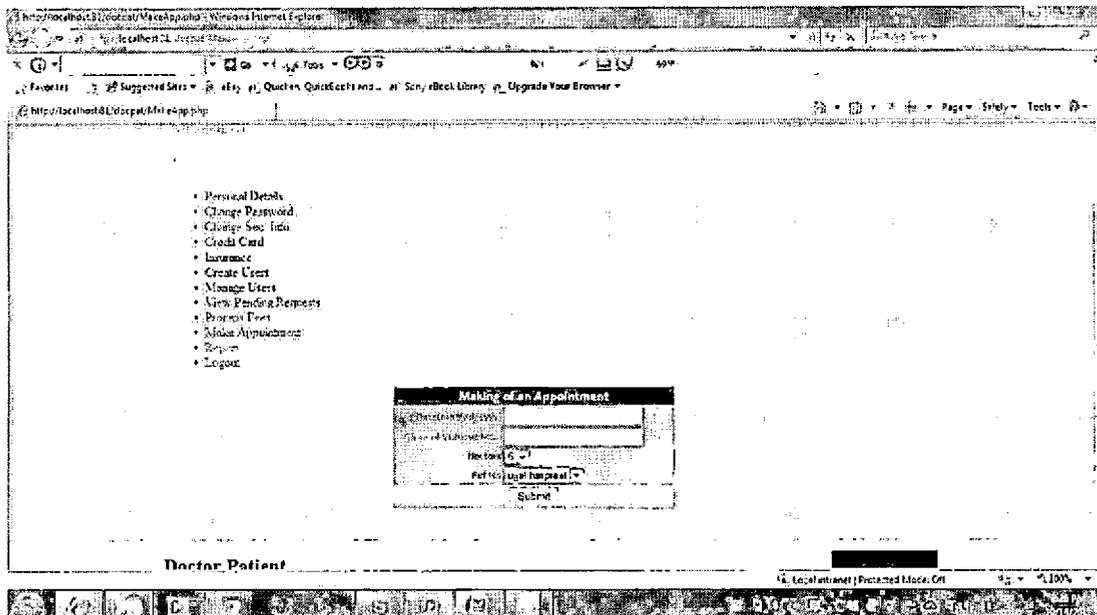


Figure 32. Create Appointment

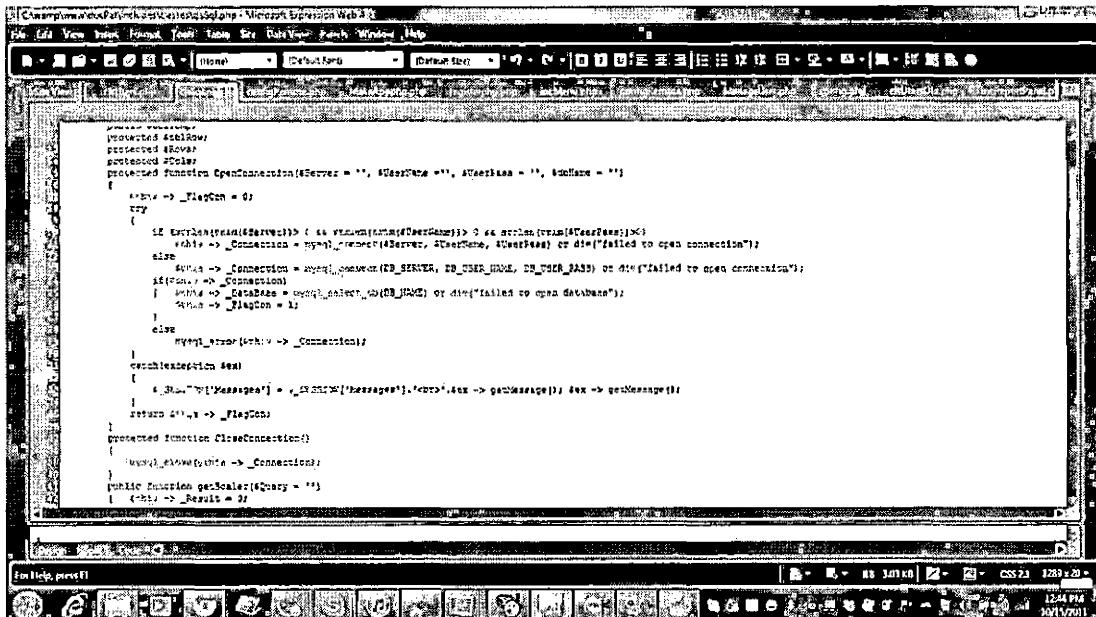


Figure 34. Database Connection Screenshot

5.3 Data-Tier Implementation

MySQL is used as a database to manage the doctor patient module web application. The data related to appointments together with the data of patients, doctors, secretaries, and clerks is stored in the form of tables in MySQL. WAMP server package is used for hosting website on local host.

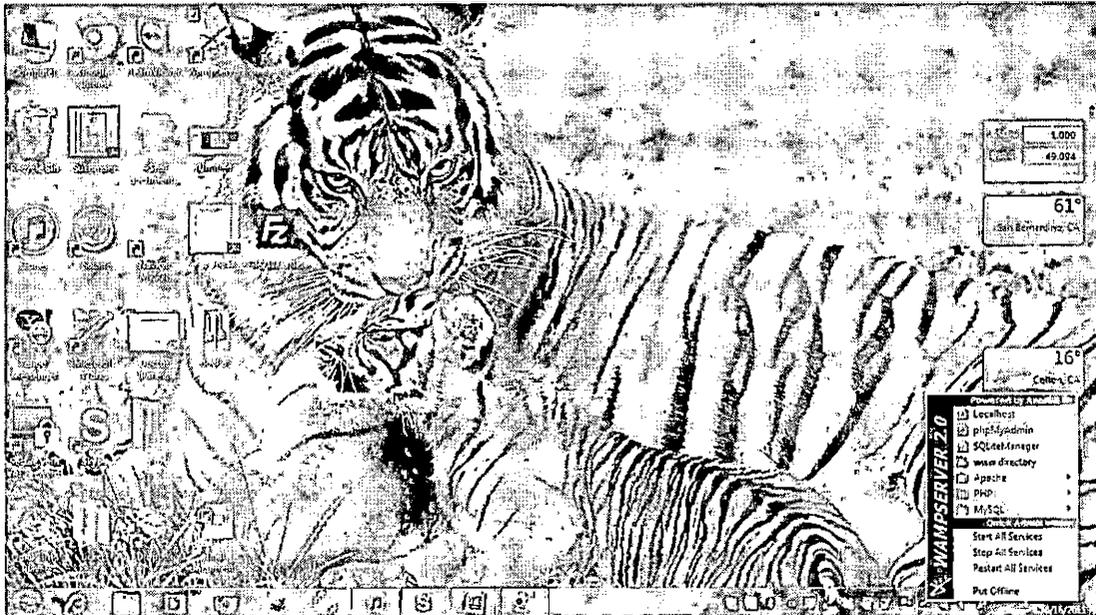


Figure 35. Wamp Server Screenshot

When the shortcut of the WAMP server is clicked, the WAMP server goes online and so do the services of the Apache and SQL server. We then open Navicat which is also open source software and is a powerful database administration and development tool for MySQL. We now create a database in Navicat named dbdocpat which contains the database of our project.

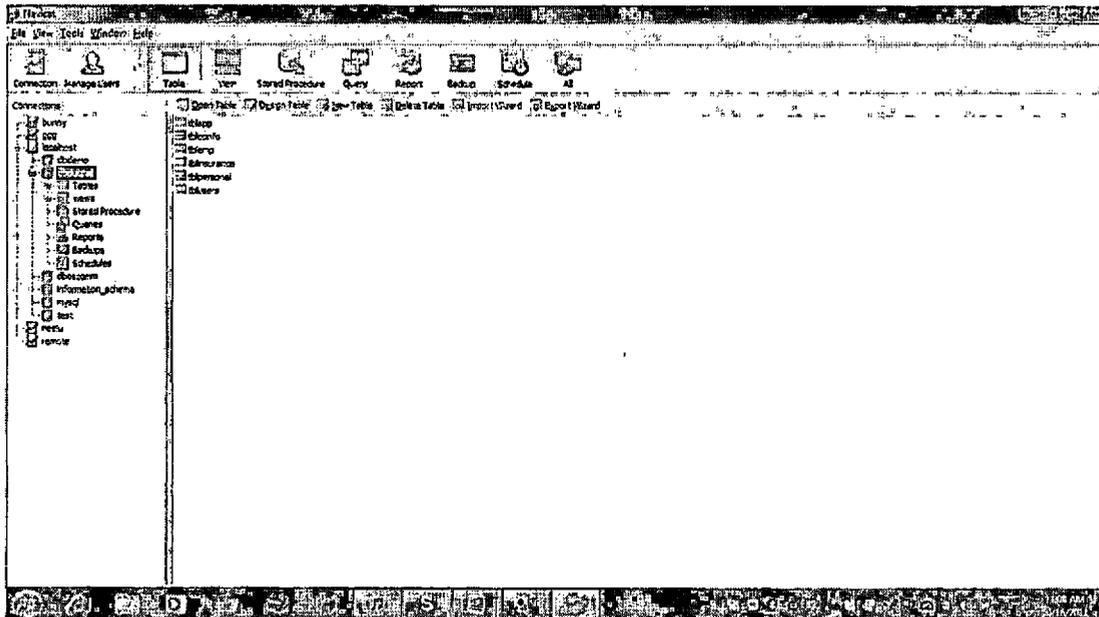


Figure 36. Database Screenshot in Navicat

CHAPTER SIX

TESTING

Reliability, performance, security are the main areas in which web applications should always perform well. Throughput and efficiency are other important factors.

The following testing was carried out in order to know these factors in the doctor patient module.

1. Unit testing: Each developed webpage was tested with data.
2. Integration testing: Integration of all the WebPages and also the navigation from one page to another was tested.
3. Functional testing: The project was tested to obtain the desired functionality.
4. Performance testing: Speed of the WebPages was tested i.e. response times of the WebPages of the application were also tested.

All tests were carried out on local host and the results obtained were as desired. Therefore the performance of the web application is very good.

CHAPTER SEVEN

INSTALLATION

Doctor Patient Module is a web application that is developed in PHP with MySQL as the database. All data related to doctors, patients, secretaries, clerks, administrator and the appointments is kept in the form of tables in the database. WAMP server package has been used which contains the Apache server. Apache server enables us to host the website on local host.

WAMP server needs to be installed on our machine [2]. It can be easily downloaded as it is open source software. After installing the WAMP server is put online. We then test port 80 in the server menu of the Apache server since it is the default port of Internet Explorer. It will tell us whether port 80 is available or what needs to be done to make it available. Once port 80 is available, then we go to MS Expression Web and open the docpat folder.

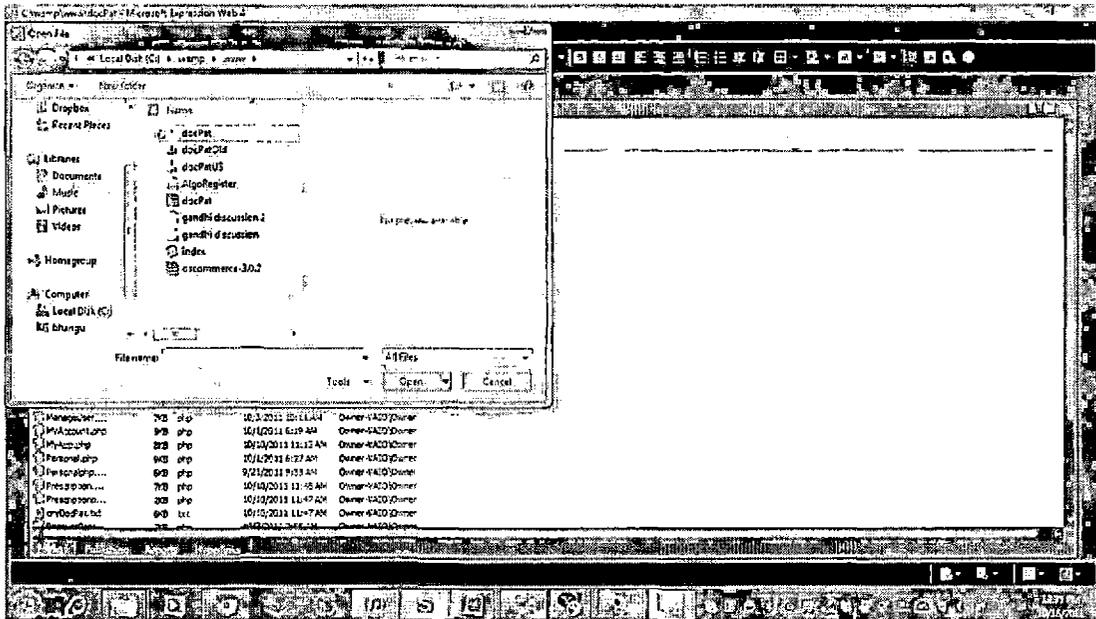


Figure 37. Docpat Folder Screenshot

All the data from the docpat folder will now be imported to ms expression web. Now we click the about webpage form which will result in.

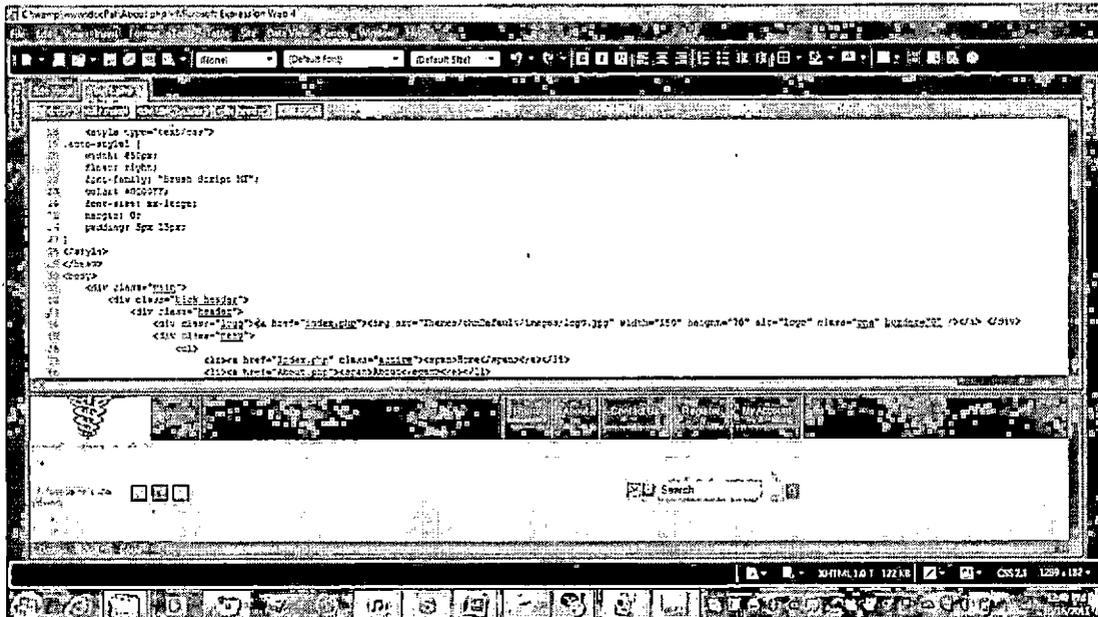


Figure 38. Screenshot in Microsoft Expression Web

We now hit F12 which directs us to the website on our local host.

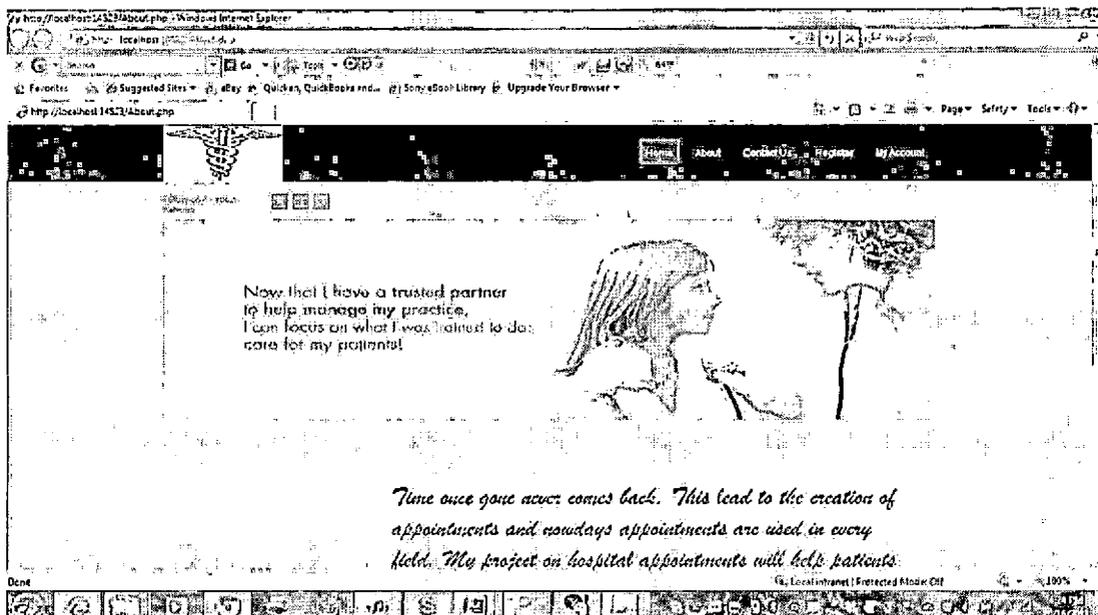


Figure 39. Screenshot of Website on Local Host

CHAPTER EIGHT

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

8.1 Summary

Doctor patient module is a web application which has been developed by the use of the latest technologies available. An attempt has been made to provide an efficient method for the creation of appointments in hospitals by different ways. The patients too will benefit a lot from this web application since they will be able to create different types of appointments also they would have different ways to pay their fee.

The web application has been made very user friendly and even a person with very little computer knowledge can access it with ease. Creating appointments in different ways online, submission of the patient's sickness symptoms online, doctors making prescription online are some of the new features of this web application.

All open source technologies like PHP, MySQL, HTML, have been used and these technologies can be easily downloaded and installed.

8.2 Conclusion

The system that handles appointments in hospitals is not the same as it used to be several years ago. The notebook that contained a portal of the appointments is now replaced by computers. The need at this point of time is improving the appointments system that is handled by the computers. The numbers of patients have increased in hospitals and so have the departments and employees of the hospital hence an appointment system should be efficient enough to handle all these changes.

The developed web application is very reliable and can handle the increased load of patients and employees very efficiently. The web application also contains new features like emergency appointments, choosing a certain doctor for appointment from a number of doctors, submissions of the symptoms of the patient's sickness online, the doctor can make prescription online, patients have been given a number of ways in which they can create their appointments and together they have number of ways in which they can pay their fee. Technologies used in the project are all open source technologies which include PHP, MySQL, HTML which make the project very cost

effective and thus any kind of hospital system from a big hospital to a small clinic can use it.

8.3 Future Recommendations

The web application has been designed to meet most of the requirements in an appointment system but still there are some recommendations for future work:

Video application: this would allow the doctor and patient to talk and see each other. This would make treatment more accurate since in case of a body injury the doctor would be able to see the exact wound and hence would provide better prescription.

Online pharmacy: this would allow the patient to buy medicine through the web application after the doctor's prescription.

Calendar integration: this would ease the patient's workload since now for an appointment he would just click the date rather than writing the date and time down.

Glossary

ASP - active server page

COM - component object model

CSS - cascading style sheet

IIS - internet information service

IT - information technology

HTML - hypertext markup language

SQL - standard query language

WAMP - windows apache mysql php server

APPENDIX A
NAVIGATION TREE

Navigation Tree

Home

About

Contact

Technology

Forum

Login | Register

My Account (Role Based)

Role – Patient

Brief History of Appointments / Transactions

Make Appointment

Get Appointment Status

Feedback

Manage Account

Change Password

Change Address

Log Off

Role – Secretary

Current Pending Appointments

Doctors' Availability

Make Appointment

Manage Account

Change Password

Change Address

Sign Off

Role – Doctor

Current Pending Appointments

Feedback from Patient

Make Prescription

Closed Appointments

Manage Account

Change Password

Change Address

Sign Off

Role – Clerk

Current Pending Appointments

Payment Details

Verify Patient and Mark Status

Get Total Collection

Get Pending Payments

Manage Account

Change Password
Change Address
Sign Off
Role – Admin
Manage Employee
Activate / Deactivate
Create
Destroy Record
Get Records
Sign Off

APPENDIX B
ALGORITHMS

Algorithms

Register

- Get Login Details
- Perform Validations
- Check if Desired User Name and Email ID Available
- If Available then
 - Create User
 - Report Success
- Else
 - Report Account Not Available
- End If

End Register

Login

- Get Login Details
- Perform Validations
- Check If the Account is authenticated
- If Authenticated
 - Create User Session and Hold User Information
 - Check Role
 - Select Role
 - Case Patient (1)
 - Move to My Account Page (Patient)
 - Case Doctor (4)
 - Move to My Account Page (Doctor)
 - Case Secretary (3)
 - Move to My Account Page (Secretary)
 - Case Clerk (2)
 - Move to My Account Page (Clerk)
 - Case Admin (9)
 - Move to My Account Page (Admin)
 - End Select
- Else
 - Report Error in Credentials
- End If

End Login

Change Password

- Check if the Person is logged in and Session is Active
- If Yes then
 - Ask for Old Password, New Password and Confirm Password
 - Validate Data (New Password = Confirm Password, Old Password Not Null)
 - If Validated then

```
Verify if User Has the Same Password as Entered
If Verify = Success then
    Change Password
Else
    Report Invalid Data Entered
End If
Else
    Report Validation Errors
End IF
End If
End Change Password
```

REFERENCES

- [1] Customer paradigm, "online appointment scheduling", www.customerparadigm.com, jun. 2010. [online]. Available: http://www.customerparadigm.com/index/325/online_appointment_scheduling_system.php [accessed: 15 Jul. 2011]
- [2] SourceForge Enterprise, "WAMP installation", <http://www.wampserver.com/en/>, Sep 2010. [online]. Available: <http://www.wampserver.com/en/download.php> [Accessed: 25 July. 2011]
- [3] Wikimedia Foundation Inc, "WAMP", wikipedia.org, Mar.2011. [Online]. Available: <http://en.wikipedia.org/wiki/WAMP> [Accessed: 28 Jul. 2011].
- [4] PremiumSoft, "Navicat for MySQL", www.navicat.com, mar. 2008. [online]. Available: http://www.navicat.com/en/products/navicat_mysql/mysql_overview.html [accessed: 1 Aug. 2011].
- [5] Hospital website, "kaiser permanente website", www.kaiserpermanente.org., jun. 2011. [online]. Available: <https://www.kaiserpermanente.org/> [accessed: 2 Aug. 2011].
- [6] Cliff McDaniel, "Ten best hospital websites", www.adazing.com, jun. 2011. [online]. Available: <http://www.adazing.com/10-best-hospital-websites/> [accessed: 6 Aug. 2011].
- [7] Jon Duckett, Beginning HTML, XHTML, CSS, and JavaScript, indianapolis, IN: Wiley Publishing, Inc., 2010.
- [8] Cristian Darie and Emilian Balanescu, Beginning PHP and MySQL E-Commerce: From Novice to Professional, Second Edition, NewYork, NY: Apress inc., 2008.
- [9] Cheryl D. Wise, Foundations of Microsoft Expression Web: The Basics and Beyond, NewYork, NY: Apress inc., 2007.

- [10] Brad Bulger, Jay Greenspan, and David Wall, MySQL/PHP Database Applications, Second Edition, Indianapolis, IN: Wiley Publishing, Inc., 2003.
- [11] Timothy Boronczyk, Elizabeth Naramore, Jason Gerner, Yann Le Scouarnec, Jeremy Stolz, Michael K. Glass, Beginning PHP6, Apache, MySQL6 Web Development, Indianapolis, IN: Wiley Publishing, Inc., 2009.
- [12] Lee Babin, Nathan A. Good, Frank M. Kromann, Jon Stephens, PHP 5 Recipes: A Problem-Solution Approach, Indianapolis, IN: Wiley Publishing, Inc., 2005.
- [13] Richard York, Beginning JavaScript and CSS Development with jQuery, Indianapolis, IN: Wiley Publishing, Inc., 2009.
- [14] Jaimie Sirovich, Cristian Darie, Wrox, Professional Search Engine Optimization with PHP A Developer's Guide to SEO, Indianapolis, IN: Wiley Publishing, Inc., 2007.
- [15] Chris Shiflett, Essential PHP Security, Cambridge, Massachusetts, O'Reilly Inc., 2005.
- [16] Frank W. Zammettia, Practical JavaScript, DOM Scripting, and Ajax Projects, NewYork, NY: Apress inc., 2007.
- [17] Private communication, charanjit singh bal, my law firm, 2010.