**DESIGN AND IMPLEMENTATION OF WEB BASED VETERINARY INFORMATION MANAGEMENT SYSTEM**

**BY**

**ARUKWE, OBUMNEME VINCENT**

**U15/NAS/CSC/099**

**Being a B.Sc project report submitted in the partial fulfillment of the requirement for the award of a Bachelor’s degree in Computer Science of the Godfrey Okoye University.**

**Supervisor: Mr. Benson**

**Department of Mathematics / Computer Science**

**Godfrey Okoye University, Enugu.**

**JULY, 2018**

**CERTIFICATION**

We hereby declare that the work presented herein was done by me, and not by a third party. Should I be convicted of having cheated in this work, I shall accept the verdict of the University.

**Arukwe, Obumneme Vincent, U15/NAS/CSC/099**

**APPROVAL PAGE**

This is to certify that this project work titled the **DESIGN AND IMPLEMENTATION OF A WEB BASED VETERINARY PATIENT MANAGEMENT SYSTEM** is authentic and the research work used for the project has been approved by the supervisor of the project and the head of department, Computer Science, Godfrey Okoye University, Enugu.

………………………………. …………………………..

**Mr. Benson Ikenna Date**

**(Project Supervisor)**

……………………………... ……………………………

**Dr. (Mrs) Monica N. Agu. Date**

**(HOD, Computer Science**)

**DEDICATION**

This project is dedicated to my loving parents Mr. and Mrs. Arukwe whom I adore and cherish for their priceless love, care, understanding and endurance. My love continues.

**ACKNOWLEDGEMENT**

My appreciation also goes to my Head of Department and Supervisor, Dr. (Mrs.) Monica N. Agufor working hard enough to make sure that we as students are given the right treatment and the right information in all ramifications. Also to all my departmental lecturers who taught me during the course of my study in school as well as my amiable course mates, am eternally grateful.

**ABSTRACT**

It has always been an issue that most veterinary hospitals do not have efficient methods of handling data for patients. There are often cases where the doctors offer wrong or repeated medication for a patient mostly because they do not have a standard record keeping facility that allows them to follow up a patient’s medical record from registration to discharge as well as check-up visits. It is in the light of this problem that the needs for strong, efficient, error-free and cost-effective methods of record keeping have to be brought into place. This project is the design and implementation of a web application for Veterinary Hospitals. The objective of this project is to provide a reliable record keeping facility for both medical practitioners and patients alike, provide a means for monitoring patient medical behavior in order to make for easy diagnosis and medical analysis, and improve the overall working experience of the Veterinary Hospital staff by making information management cost effective and easy on the long run. The programming languages used in this design are HTML (Hypertext markup language), CSS (Cascading Style Sheets), SQL (Structured Query Language), PHP (Personal Homepage) and Javascript. It allows medical practitioners and staff to have a profile (by registration) on the portal and register patients as well. Doctors have access to edit patient medical data and add new data for each visit while the ordinary staff does not. In the end, all the patient needs is a hospital card in order to track the patient id, all other information regarding the patient’s business with the hospital is already known because it is stored in the database.

**TABLE OF CONTENTS**

Title page i

Certification ii

Approval Page iii

Dedication iv

Acknowledgement v

Abstract vi

Table of contents vii

List of Figures ix

List of Tables x

## CHAPTER ONE: INTRODUCTION

* 1. Background of the study 1
  2. Statement of the problem 1
  3. Objective of the study 2
  4. Significance of the study 3

**CHAPTER TWO: REVIEW OF RELATED LITERATURE**

2.0 Introduction 4

2.1 Theoretical Background 4

2.2 Review of Related Literature 6

2.3 Summary 10

## CHAPTER THREE: SYSTEM ANALYSIS AND DESIGN

* 1. Introduction 11
  2. Analysis of Existing System 13
  3. Design of the Proposed System 14
     1. Input Design 17
     2. Output Design 18
     3. System Architecture 19
     4. Database Design 19

## CHAPTER FOUR: SYSTEM IMPLEMENTATION

4.0 Introduction 21

4.1 Choice of Programming Environment 21

4.2 Implementation Architecture 23

4.3 Software Testing 25

4.4 Documentation 28

4.4.1 User Manual 28

4.4.2 Source Code listing 29

**CHAPTER FIVE: SUMMARY AND RECOMMENDATION**

5.1 Summary 30

5.2 Conclusion 30

5.1 Recommendation 30

**REFERENCES** 31

**APPENDIX** 33

**List of Figures**

Fig 1: Structured System analysis and Design Methodology 12

Fig 2: Design of the proposed system using UML Use Case diagram 14

Fig 3: Input Design 18

Fig 4: Output Design 18

Fig 5: Staff Platform Implementation Architecture 23

Fig 6: Doctor’s Platform Implementation Architecture 23

Fig 7: Admin Platform Implementation Architecture 24

Fig 8: Screenshot of the Homepage 25

Fig 9: Screenshot of the Login Page 26

Fig 10: Screenshot showing registered patients 27

Fig 11: Screenshot of Form for entering new Visit 27

Fig 12: Screenshot of Confirmation of added Visit 28

**List of Tables**

Table 1: Database design for Patient Registration (Sample Database Design) 19

**CHAPTER ONE**

**INTRODUCTION**

**1.0 BACKGROUND OF THE STUDY**

Veterinary medicine is a branch of medicine that deals with the prevention, diagnosis and treatment of disease, disorder and injury in non-human animals. As a branch of medicine, it is necessary for the doctors to take active part in ensuring that the patients are well taken care-of and well followed up with respect to illnesses, diagnosis and the prescription of drugs. To do this, they often need an efficient management system in their clinics or hospitals that would keep track of the daily visits and returning visits of patients and the progress made in the treatment of the illnesses. This project aims at building an efficient, flexible, versatile web-based intranet for the patient management of veterinary hospitals and clinics.

**1.1 STATEMENT OF THE PROBLEM**

In most hospitals, it is often ambiguous, time consuming and exhausting with respect to the issue of patient management and follow-up. This is due to the following issues;

* Loss of patient data: Patient data is sometimes kept in physical storage file which get worn-out over a long period of time and get prone to wear and tear. Also, there is the issue of document losses which occur periodically.
* Time Cost: Most hospitals receive more than 200 patients daily. And each patient has a file dedicated to him / her. Having to write to these physical files everyday can be time consuming. Also, in a case where the patient re-visits, sorting out files is often a time consuming fact and also energy draining.
* Space: Physical files need physical space for storage, this physical storage space is often not available and often limited. As a result, there is often the need to create a complete apartment for the storage of these files, which, in the long run, is often not enough.

**1.2 OBJECTIVE OF THE STUDY**

The objective of this study is to achieve the following;

* An efficient, light and portable means of storing patient data in veterinary hospitals.
* An efficient method of communication between the various divisions in a hospital with respect to the treatment and transfer of patients from one department to another.
* To reduce the time cost of processing data. With a database management facility, the issues of sorting, searching and traversing manually comes to an abrupt end. Database management systems have very fast means of searching for data using keywords and identity keys. Such of which is not possible with manual technology.

**1.3 SIGNIFICANCE OF THE PROJECT**

This project is significant in the sense that it revolutionizes the patient management systems in veterinary hospitals, it will make communication faster and much easier, it will make data processing much more efficient and less prone to errors, it will allow one to treat each patient as a separate entity instead of being a part. If efficient, it could also be used in human hospital management systems as well as other areas where personnel data and activity management is involved; places like schools, industries, offices etc.

**CHAPTER TWO**

**LITERATURE REVIEW**

**2.0 INTRODUCTION**

This chapter aims at introducing the technologies used in the design and implementation of this utility. It would view related literature, concepts used in the technology as well as a brief breakdown of how they are used.

**2.1 THEORETICAL BACKGROUND**

The technologies used in the design and implementation of this application are dynamic web-design technologies and database management systems. Namely;

* **HTML:** Known as Hyper-Text Markup Language; the basic language used for the creation of web-pages. It is the background skeleton of every web-page. A standard piece of code in html is often like this;

**<html>**

**<head>**

**<title> My Webpage</title>**

**<link href = ‘style.css’/>**

**<script type = ‘javascript’ src = ‘text.js’></script>**

**</head>**

**<body>**

**Content goes here**

**</body>**

**</html>**

* **PHP:** Hyper-Text Pre-Processor; this language is used for connecting the web-page to the database management system for the dynamic storage and retrieval of data. It often looks basically like this;

**<?php**

**Echo “HERE I AM”;**

**Echo “I AM PROGRAMMING IN HTML”;**

**?>**

* **JAVASCRIPT:** Javascript is a scripting language used to add functionality and uniqueness to web-pages. Concepts like authenticating that the contents of a form are its supposed contents, animations, adding and deleting rows of a table when a button is clicked, alerting a user of an error, are all achievable through Javascript. Its basic syntax usually looks like this;

**<script type = ‘text/javascript’>**

**Function hello(){**

**Alert(“Hello World, This is a program”);  
}**

**</script>**

* **CSS:** CSS is an acronym for the ‘Cascading Style Sheet’. It is used for adding styles, color and beauty to the page. Issues like setting margins, border-width, background-color, visibility, height, width, opacity, font-size, font-weight etc, are all achievable using CSS. A sample of CSS syntax is described as follows;

**Body{**

**Background-color: white;**

**Padding: 0px;**

**Margin: 0px;**

**Opacity: 0.9;**

**}**

**Html-selector {**

**Font-size: 20px;**

**Font-weight: bold;**

**Font-family: times new roman;**

**}**

* **SQL:** SQL is an acronym for the Structured Query Language. It is the language that accesses the database and gets information from it by querying. The access may require querying the database for data, traversing the database or reading all information from the database. SQL is often accessed using PHP commands from the web-page. It often works with a server that allows it to access information from the database (often independent of the webpage) and display it on the web-page.

**2.2 REVIEW OF RELATED LITERATURE**

The idea of patient health records emerged in the early 1970s [1,2] with the goal of increasing patient engagement and empowerment, which in turn was intended to enable continuity of care, error reduction [3], treatment choice, and patient-provider partnership building [1,2].

An extension of traditional electronic health records, PHRs (Patient Health Records) created a patient-centric platform supporting the new vision of health services that enables patient-provider information sharing and collaboration, with the goal of improving health outcomes and reducing costs. In recent decades, great strides have been made toward achieving these far-reaching goals in research and practice. As a result of high improvements in the ability of information systems and ICT (Information and Communication Technology), the use of PHR data is becoming more commonplace [4]. As defined by the program, the initial stage of meaningful use encourages providers to integrate technology into medical practice, making vast amounts of patient data available electronically. Later stages of the program focus on empowering patients by providing them with online access to their health data.

The use of PHRs has grown since the rise of mobile computing and advancement of patients’ technical aptitude. As an extension of EHRs, PHRs have been developed to enable patients to manage their own health care. The amount of overlap in terms of data and functionalities between the traditional health records and PHR depends on the type of implementation which could be tethered, interconnected, or stand-alone [5]. Functionalities available through the PHR are intended to be used by patients, rather than by providers, and include appointment scheduling, prescription refill, and secure messaging [6]. The newly developed PHRs created a complementary source of clinical data such as patient-reported outcomes [7-9], physician ratings [10], medication adherence [11], and social support [12,13], and they allow for new data analytics techniques to detect, measure, and predict health-related outcomes. Outstandingly, countries like the United States have been leading in the field of PHR data analytics. One reason for the growth of health care analytics in the United States is the incentivization of such research through federal initiatives to deliver patient-centered care and quality-driven payment models [14,15]. The Partnership for the Future of Medicare [15] states that innovative methods, such as email consultations and self-monitoring, must be used to achieve individualized, effective care. Additionally, Medicare strives to make health care data more readily available and accessible, including quality and performance metrics. Taken together, these initiatives support health care data collection and utilization in the United States, making PHR analytics more feasible. However, the full potential of PHR cannot be realized until we have a better understanding of PHR data content, formats, and sources.

Tremendous amounts of patient data are now available through PHR systems. With patients’ permission, these data, along with the application of advanced data mining and machine learning, can provide significant new opportunities in research. For instance, models in areas such as disease prediction, patient risk assessment, and early symptom detection can now be improved, leading to major advances in health outcomes and cost optimization. However, along with new opportunities provided by PHR systems come data and user-related challenges. Data-related issues such as quality, privacy, and security pertain to collection, safe storage, and processing of large quantities of patient data from distributed information systems. Also, patients previously excluded from access to such systems may lack the expertise to understand the data [16].

After initial uses of PHR systems in the early 2000s, new data elements such as appointments, preferences, and system settings emerged. More recently, PHR data included reminders (E.g, appointment reminders, medication reminders, screening and laboratory work reminders, immunization reminders, preventive care reminders, and health maintenance reminders [21]), in addition to alerts [22], identification of personal health goals [19], and disease prevention. Tracking and monitoring data via e-journals and diaries also became available.

Today, PHR data are generated through different tools and devices. Tracking devices, now transmitting time-series PHR data, are used to monitor patients’ vital signs, such as blood pressure and glucose level (biomonitoring devices), and to detect abnormal events, such as alerts from implantable cardioverter defibrillators.

#### Scope of Patient Health Record Systems

PHR data were mainly used to provide added functionalities to patients. The provider search results [20], for example, helped patients locate health care providers and health-related services. Similar functionalities enabled patients to obtain health advice from support groups. Other functionalities assisted patients with preparing for medical encounters through visit preparation questionnaires [24]. Functionalities such as incentive programs empowered patients through self-health monitoring. Finally, a unique PHR data category discovered in our review, environmental information, captured community health concerns and environmental domains, which can be linked to functionalities such as assessment of environment-related risk factors and recommendations for preventive care.

**2.3 SUMMARY**

As a summary, we’ve noticed that medical information storage has become of vital use in the health institution today.

The technologies used in the construction of this medical facility is the web design methodology because it is easily hosted on a network and the programming languages used were HTML, CSS, JAVASCRIPT, PHP AND MYSQL Database.

This project explores the ability of computers to make offices paperless (in the sense that physical records are kept to the barest minimum).

**CHAPTER THREE**

**ANALYSIS AND DESIGN OF THE SYSTEM**

**3.0 Introduction**

The methodology used in this project is the Structured System Analysis and Design Methodology (SSADM) which is best described as the waterfall method for building information systems. It is made up of seven stages, which include:

* Feasibility Study: Which deals with investigating the goals and implications of the project (Whether this process is carried out or not depends on the size of the project. I.e. sometimes it is by-passed).
* Investigation: Here, the analyst comes to full understanding of the existing system as it is at the start of the project.
* Business System Options: The analyst decides on the overall design of the new system using results from previous stages.
* Requirements Specification: The analyst tries to develop a full logical specification of what the new system must do, the specification must be error free, free of ambiguity and free of inconsistencies.
* Technical System Options: Here, the analyst moves towards the physical implementation of the new system. At this stage, attention shifts from design and description to actual implementation.
* Logical Design: As the previous level describes the details of the implementation, the outputs at this stage are independent of implementation but mostly concentrate on the user interface and how they would interact with each other.
* Physical Design: This is the final stage of the design. At this stage, the analyst must have gathered all necessary information possible, from design and description to actual implementation. It is at this stage that the software engineers are expected to know how to build the system in specific details of hardware and software to the desired standards.

**FEASIBILITY**

**INVESTIGATION**

**BUSINESS SYSTEMS OPTIONS**

**REQUIRMENT DEFINITION**

**TECHNICAL SYSTEM OPTIONS**

**LOGICAL DESIGN**

**PHYSICAL DESIGN**

**Fig 1: Structured Software analysis and design methodology**

**3.1 Description of the existing system**

The existing system is pretty much manual with the following processes;

* + The patient walks into the hospital to see the receptionist
  + Hospital creates a list of cards continuously as well as a large diary for the input of patients and their visiting days.
  + The patient is given a registration number, manually as well as an appointment date.
  + The staff enters the patients data manually into the diary
  + When the patient visits the doctor, the doctor writes the medications or proposed treatment procedures and forwards it to the necessary department by hand. Hence, the transfer of messages and prompts is manual.
  + When the registry book for patients, is exhausted, the receptionist requires another book for the registration of patients. Next appointment dates are recorded in another registry book by the doctor in charge of that patient. Also, as soon as the registry book is exhausted, the doctor has to get another one.

During analysis however, the following problems were identified in the above system;

* **Added Cost:** There always has to be a constant supply of registration books periodically for both staff and doctors at the hospital, as well as patient cards for patient management and patient files to keep specific patient records. This alone is an added cost to record management.
* **Poor Storage space:** Because the registration process is manual, their often has to be reserved physical storage space to store old records of patient should they return at a later time. Also, searching and traversing through the archives to find old records is often energy and time consuming and needs the extra service of another staff who must be paid to search the records. This is an added cost as well.
* **Risk:** One is at risk of losing data when doing things manually, because all records are physical. As a result, accidents such as fire or misplacement can render records lost forever.

**3.2 Design of the Proposed System**

The proposed system is a combination of manual and automated designs. The manual part is the user entry while the automated part is done by the computer.

The Unified Modeling Diagram (User-case diagram) is given in figure 2;

**VET Medical staff**

**Registered Veterinary Doctor**

**Registered Normal Staff**

**Authentication**

**Unregistered Staff**

**Fig 2: Design of the Proposed System using Use Case Diagram**

The advantage of this system is as follows;

* **Cost:** This method is cost-effective because of the following features;
  + Physical files don’t exist anymore, every patient’s details is saved in the database and can be viewed by any staff member registered in the hospital portal from anywhere in the hospital.
  + The space allocated for the storage of files is limited to the size of the memory of the computer which is usually measured in Gigabytes (GB). Hence, an efficient computer with 500GB memory dedicated for this service is likely to take up data that can fill 5-6 large rooms if kept in physical files.
  + Error removal or editing is easier and cleaner. There is no need to dispose papers if there is an error. One can just easily edit a patient’s data by using the edit button presented next to the patient’s data in the database table.
* **Bureaucracy:** The process of file movement *to* and *fro* is reduced to the barest minimum. There is no need for a messenger whose job is to carry files from one office to another or whose job is to go on errands. Once the information is stored in the database, it can accessed from anywhere around the hospital as long as that staff member is registered in the hospital portal.
* **Unified Record Management:** Usually, in the existing record management system, every department in the hospital has to keep its own records of patient’s visiting the department and the outcome of the treatment given. However, with this system, records are unified. The receptionist fills in the personal data of the patient, the doctor adds the medical report and suggested treatment as well as the department to handle such cases, while the other departments use the doctor’s prescription as a guideline for their therapies and medications offered. In this way, a single patients’ record will carry the receptionist’s report, the doctor’s report as well as the department’s report.

The Limitations of this system however are as follows;

* In developing countries like Nigeria where there is inconsistent power supply and inflation of prices, running an automated patient management system is not very cost-effective, due to the fact that power supply is unsteady and the price of fuel for generators is high, hence the need for constant power supply may be a limitation of this application in some parts of the world.
* Installation and maintenance is difficult and costly. This is due to the fact that every doctor in the hospital is entitled to at least one workstation (A desktop computer setup) and every staff office should have at least one workstation. All of which must be connected to the server; meaning that the whole hospital must be networked to use the utility. It is also noteworthy that maintenance of this infrastructure often involves troubleshooting all workstations available which is costly, as well as the cost of keeping power supply constant.

**3.2.1 Input Design**

The input format used in this project is the form. The user enters the form and submits it for validation and processing. An example is the form that registers patients into the hospital’s database as seen in Figure 3.

**SUBMIT**

**Data 1:**

**Data 2:**

**Data 3:**

**Data 4:**

**Fig 3: Input Design**

**3.2.2 Output Design**

The data is displayed to the user in the form of tables, when the user submits the form as described above, the program queries the database using the information in the form and the data is displayed onto the user as a form. Figure 4 describes this process.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Data 1** | **Data 2** | **Data 3** | **Data 4** |
| **1** | **Output 1** | **Output 2** | **Output 3** | **Output 4** |

**Fig 4: Output Design**

**3.2.3 Database Design**

The database design is relational. It uses tables for record keeping. Table 1 illustrates the database table structure using the table for patient registration.

**Table 1: Database Structure for Patient Registration Design (standard database design)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Attribute** | **Null** | **Default** | **Extra** |
| **1** | **Id** | **Int(100)** |  | **No** | **None** | **AUTO\_INCREMENT** |
| **2** | **Patient\_id** | **Varchar(200)** |  | **No** | **None** |  |
| **3** | **Patient\_name** | **Varchar(200)** |  | **No** | **None** |  |
| **4** | **Patient\_type** | **Varchar(200)** |  | **No** | **None** |  |
| **5** | **Guardian\_name** | **Varchar(200)** |  | **No** | **None** |  |
| **6** | **Contact\_details** | **Varchar(200)** |  | **No** | **None** |  |
| **7** | **Date\_of\_registration** | **Timestamp(6)** | **On\_update CURRENT\_TIMESTAMP** | **No** | **CURRENT\_TIMESTAMP** | **ON UPDATE CURRENT\_TIMESTAMP** |

**3.2.4 System Architecture**

The architecture of a system is the conceptual model that defines the structure, behavior, and more views of the system. It is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. The following characteristics are noted.

* The administrator does not have access to change the medical or personal data of the patients directly. Instead, he would have to login through the staff entry or doctor entry to do so. This requires him taking information from the database to do so.
* The ordinary staff member does not have the permission to edit the medical report of the patients and their visits. He/she can only edit or change the personal data of the patient.
* The Doctor has exclusive rights to change, create new file for patients, delete patient data and update patient data as well as his own personal data. What differentiates him from the admin is the ability to see the complete information of all staff and doctors in the hospital.

**CHAPTER FOUR**

**SYSTEM IMPLEMENTATION**

**4.0 INTRODUCTION**

This chapter discusses the results observed during the implementation of the proposed design, as well as the platform used in the design and the programming language used. It also shows the documentation and the various linkages between the elements of the web application.

**4.1 CHOICE OF DEVELOPMENT ENVIRONMENT**

The platform used in the development of this website is made up of four main tools;

* **Programming Language:** As explained in Chapter II, the programming languages used are;
  + Hypertext Markup Language (HTML)
  + Cascading Style Sheet (CSS)
  + Javascript (JS)
  + Hypertext Pre-processor (PHP)
  + Structured Query Language (SQL)
* **Integrated Development Environment (IDE):** the environment used to write the programming languages and arrange them together to make meaningful code. It is used to test the correctness of syntax and semantics of code written using the programming languages employed. In other words, it is the building platform for the project. The IDE used for this project is Sublime Text Editor.
* **XAMPP SERVER:** This is the server that acts as a simulator, making it looking like the web application is already hosted on the internet. It has tools that allow you to connect to host the application on a single system as well as database management system tools for managing databases needed for the dynamic functionality of the application.
* **Database Management System:** A database-management system (DBMS) is a computer-software application that interacts with end-users, other applications, and the database itself to capture and analyze data. The database management software in this project was provided by the Xampp Server.

**4.2 Implementation Architecture**

* For Staff:

**INTRODUCTORY PAGE**

**LOGIN / REGISTER**

**VIEW / EDIT / ADD PATIENT BASIC DATA**

**LOGOUT**

**DATABASE**

**VIEW / EDIT / ADD PERSONAL DATA**

**Fig 5: Staff platform Implementation Architecture**

* For Doctors

**INTRODUCTORY PAGE**

**LOGIN / REGISTER**

**VIEW / EDIT / ADD PATIENT MEDICAL DATA**

**LOGOUT**

**DATABASE**

**VIEW / EDIT / ADD PERSONAL DATA**

**Fig 6: Doctor Platform Implementation Architecture**

* For the Admin

**INTRODUCTORY PAGE**

**LOGIN / REGISTER**

**VIEW / EDIT / ADD STAFF/ DOCTOR DATA**

**LOGOUT**

**DATABASE**

**VIEW PATIENT DATA**

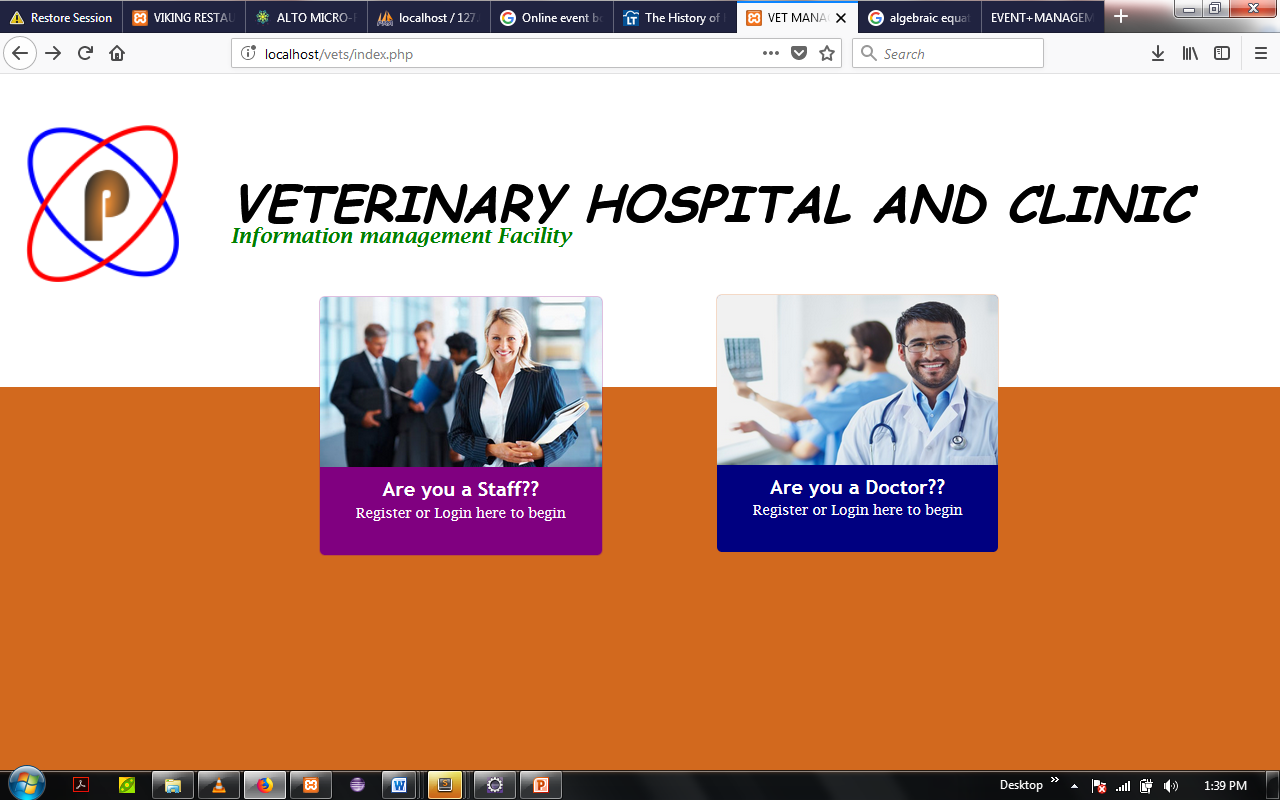
**Fig 7: Admin Platform Implementation Architecture**

**4.3 Software Testing**

This is a test of the application at the stages of its development in order to check for errors or bugs, and if found, are removed or fixed. This application is made for two main members of the organization;

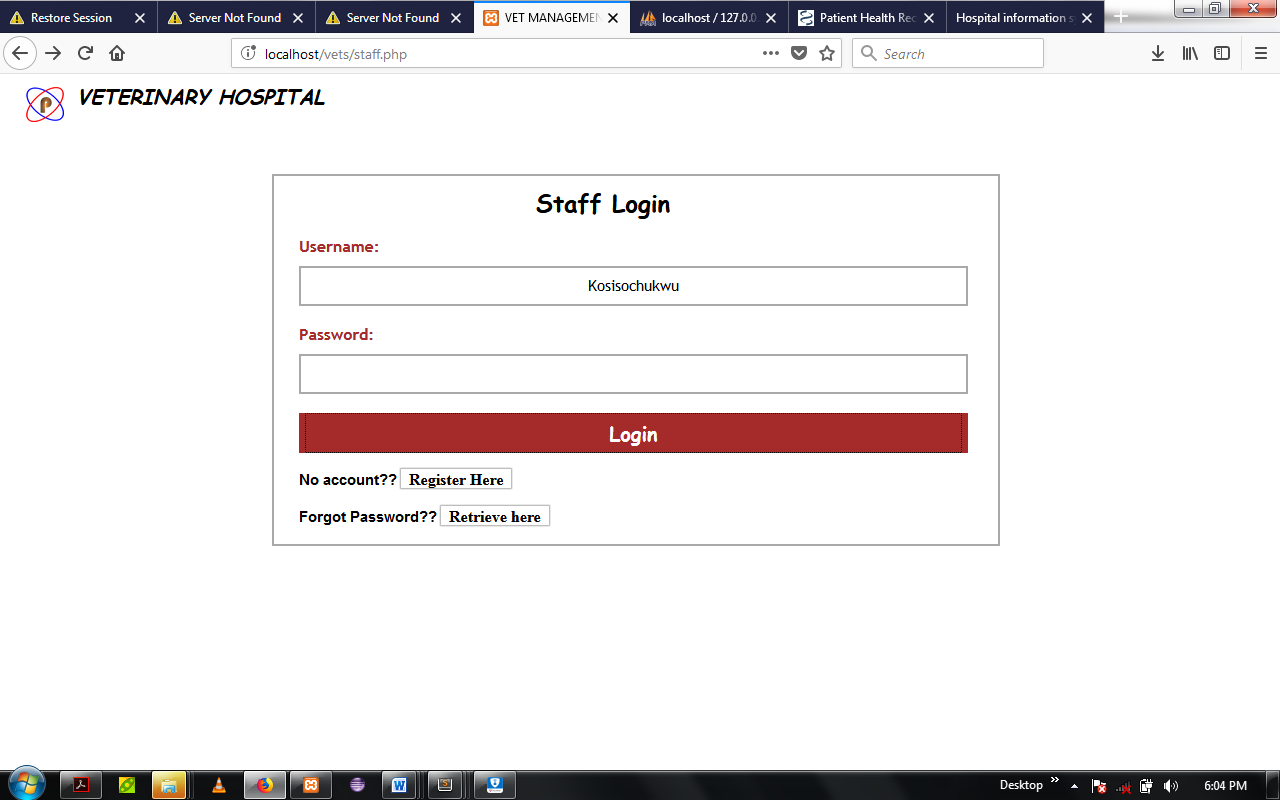
* The staff
* The Doctors

Figure 6 below displays the Homepage;



**Fig 8: Screenshot of the Homepage**

The Login Page for the User (Staff or doctor);



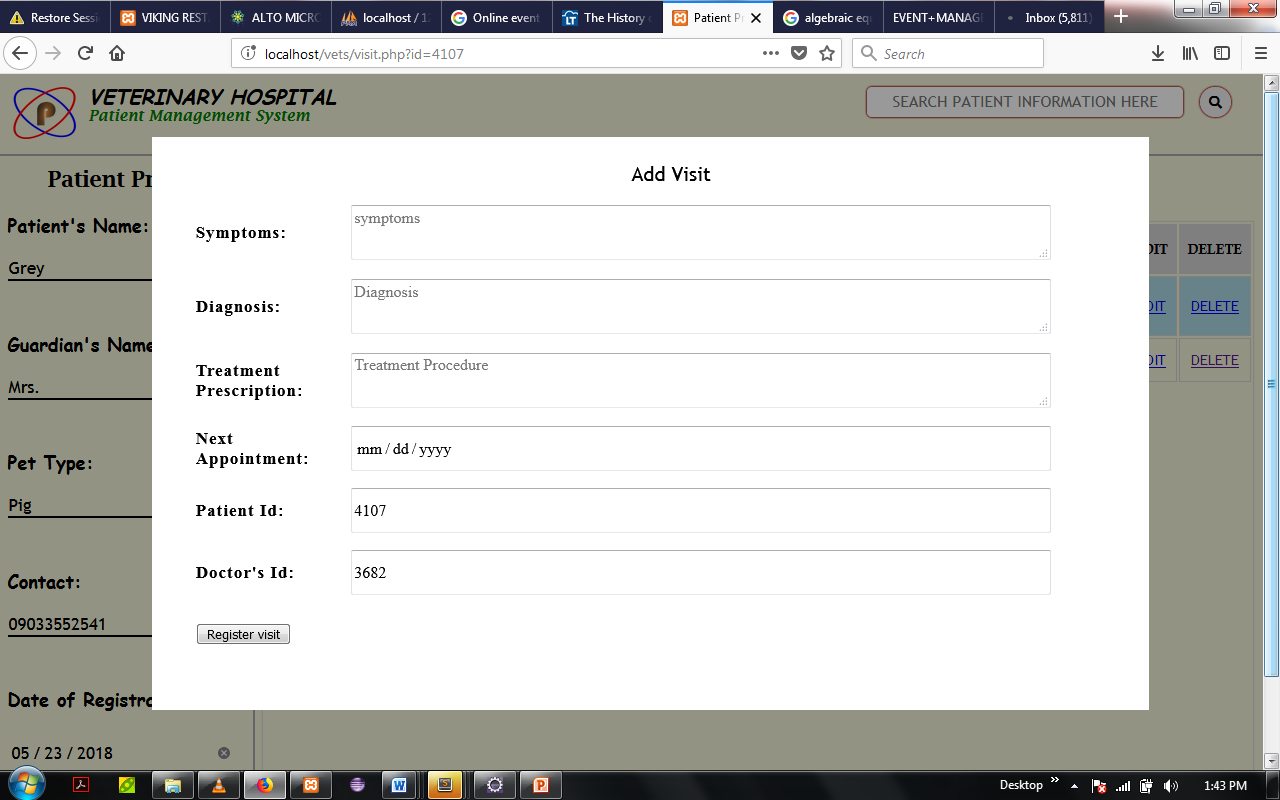
**Fig 9: Screenshot of Login Page**

Upon Login, the user can register a patient or view the patient data;



**Fig 10: Screenshot of Registered Patients**

To add new visit,



**Fig 11: Screenshot of Form for entering new visit**

After the data has been entered, the software sends a message of confirmation,



**Fig 12: Screenshot of confirmation of added visit**

After which the application returns to the general view page.

**4.4 Documentation**

**4.4.1 User Manual**

* To begin, boot the computer system and open your browser.
* On the search pane, type in “**localhost/vets**” and press “**Enter**” on the Keyboard.
* When the website loads completely, enter into any of the packages (staff or doctor) and login or create an account to begin.
* To logout, go to the patient pages on your accounts (staff or doctor) and click on the logout button visible on the dashboard.

**4.4.2 Source Code Listing**

1. Source code listing for the homepage is in Appendix A

2. Source code listing for the staff user page is in Appendix B

3. Source code listing for the doctor user page is in Appendix C

**CHAPTER FIVE**

**SUMMARY, CONCLUSION AND RECOMMENDATION**

* 1. **Summary**

This project is designed for the patient record keeping and management of veterinary hospitals. It is designed using the web application approach with database implementation and the programming languages used in the design include: HTML (Hypertext markup language), CSS (Cascading Style sheet), Javascript, PHP (Hyper-text preprocessor), SQL (Structured Query Language) for databases.

**5.2 Conclusion**

The project is significant because it improves the service experience of both staff and patients of veterinary hospitals in terms of patient record management and general hospital management and as such eliminates much of the problems that come with traditional record keeping.

**5.2 Recommendation**

This project has proved to be significant because it revolutionizes hospital management from manual to automated. It could also be improved in future to be able to attend to simple patient needs like drug allocation and disease detection.

**REFERENCES**

[1] Shenkin BN, Warner DC. Sounding board. Giving the patient his medical record: a

proposal to improve the system. N Engl J Med. 1973 Sep 27;289(13):688–92. doi: 10.1056/NEJM197309272891311.

[2] Hinman E, Holloway J. The patient carried personal health record: a tool to

increase patient participation in the treatment process. J Clin Comput. 1977;6(4):9.

[3] Jackson AN, Kogut S. Use of electronic personal health records to identify

patients at risk for aspirin-induced gastrointestinal bleeding. Consult Pharm. 2013 May;28(5):313–8. doi: 10.4140/TCP.n.2013.313.

[4] Mennemeyer ST, Menachemi N, Rahurkar S, Ford EW. Impact of the HITECH

Act on physicians' adoption of electronic health records. J Am Med Inform Assoc. 2016 Mar;23(2):375–9. doi: 10.1093/jamia/ocv103.

[5] Tang PC, Ash JS, Bates DW, Overhage JM, Sands DZ. Personal health records:

definitions, benefits, and strategies for overcoming barriers to adoption. J Am Med Inform Assoc. 2006;13(2):121–6. doi: 10.1197/jamia.M2025. http://europepmc.org/abstract/MED/16357345.

[6] Archer N, Fevrier-Thomas U, Lokker C, McKibbon KA, Straus SE. Personal

health records: a scoping review. J Am Med Inform Assoc. 2011;18(4):515–22. doi:10.1136/amiajnl-2011-000105 http://jamia.oxfordjournals.org/cgi/pmidlookup?view=long&pmid=21672914

[7] Gordon MF, Lenderking WR, Duhig A, Chandler J, Lundy JJ, Miller DS, Piault-

Louis E, Doody RS, Galasko D, Gauthier S, Frank L, Patient-Reported Outcome Consortium's Cognition Working Group Development of a patient-reported outcome instrument to assess complex activities of daily living and interpersonal functioning in persons with mild cognitive impairment: The qualitative research phase. Alzheimers Dement. 2016 Jan;12(1):75–84. doi: 10.1016/j.jalz.2015.04.008.

[8] Chung AE, Sandler RS, Long MD, Ahrens S, Burris JL, Martin CF, Anton K,

Robb A, Caruso TP, Jaeger EL, Chen W, Clark M, Myers K, Dobes A, Kappelman MD. Harnessing person-generated health data to accelerate patient-centered outcomes research: the Crohn's and Colitis Foundation of America PCORnet Patient Powered Research Network (CCFA Partners) J Am Med Inform Assoc. 2016 May;23(3):485–90. doi: 10.1093/jamia/ocv191. http://europepmc.org/abstract/MED/26911821.

[9] Bingham CO, Bartlett SJ, Merkel PA, Mielenz TJ, Pilkonis PA, Edmundson L,

Moore E, Sabharwal RK. Using patient-reported outcomes and PROMIS in research and clinical applications: experiences from the PCORI pilot projects. Qual Life Res. 2016 Aug;25(8):2109–16. doi: 10.1007/s11136-016-1246-1. http://europepmc.org/abstract/MED/26914103.

[10] Segal J. The role of the Internet in doctor performance rating. Pain Physician.

2009;12(3):659–64. http://www.painphysicianjournal.com/linkout?issn=1533-s3159&vol=12&page=659

[11] Heldenbrand S, Martin BC, Gubbins PO, Hadden K, Renna C, Shilling R, Dayer

L. Assessment of medication adherence app features, functionality, and health literacy level and the creation of a searchable Web-based adherence app resource for health care professionals and patients. J Am Pharm Assoc (2003) 2016;56(3):293–302. doi: 10.1016/j.japh.2015.12.014.

[12] Allen C, Vassilev I, Kennedy A, Rogers A. Long-term condition self-

management support in online communities: a meta-synthesis of qualitative papers. J Med Internet Res. 2016;18(3):e61. doi: 10.2196/jmir.5260. http://www.jmir.org/2016/3/e61/

**APPENDICES**

**Appendix A: Source code listing for Homepage**

<?php

include\_once 'libraries/functions.php';

include\_once 'libraries/config.php';

?>

<html>

<head>

<title>VET MANAGEMENT SYSTEM</title>

<link rel="stylesheet" type="text/css" href="css/style.css">

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">

<meta name="viewport" content="width=device-width, initial-scale=1">

<script type="text/javascript" src = 'js/common.js'></script>

<script type="text/javascript" src = 'js/jquery.js'></script>

</head>

<style type="text/css">

body{

background-color: chocolate;

}

div[id = 'entry'] {

width: 100%;

height: 45%;

background-color: white;

z-index: 5;

}

div[id = 'ace'] {

margin-top: -9%;

margin-left: 18%;

width: 90%;

font-family: Cursive;

font-size: 50px;

color: black;

}

div[id = 'ice'] {

margin-left: 18%;

width: 90%;

font-family: Lucida Bright;

font-size: 20px;

color: green;

font-weight: bold;

margin-top: 4%;

font-style: italic;

}

img[id = 'img'] {

width: 12%;

height: 50%;

margin-top: 4%;

}

div[id = 'staff'] img, div[id = 'doctor'] img {

width: 100%;

height: 170px;

border-top-left-radius: 4px;

border-top-right-radius: 4px;

margin-left: 0px;

margin-top: 0px;

}

div[id = 'diss'] {

text-align: center;

width: 100%;

font-family: Trebuchet MS;

font-size: 20px;

margin-top: 10px;

font-weight: bold;

}

div[id = 'text'] {

width: 95%;

margin-top: -10px;

text-align: center;

font-family: Lucida Bright;

font-size: 14px;

margin-left: 7px;

}

div[id = 'doctor'] {

width: 22%;

margin-left: 56%;

margin-top: -260px;

height: 37%;

z-index: 10;

padding: 0;

padding: 0px;

box-shadow: 0px 0px 1px 0px chocolate;

background-color: navy;

border-radius: 5px;

transition: all 0.5s;

color: white;

}

div[id = 'staff'] {

width: 22%;

margin-left: 25%;

margin-top: -90px;

height: 37%;

z-index: 10;

padding: 0px;

box-shadow: 0px 0px 1px 0px chocolate;

background-color: navy;

border-radius: 5px;

transition: all 0.5s;

color: white;

}

div[id = 'staff']:hover, div[id = 'doctor']:hover {

background-color: purple;

box-shadow: 0px 0px 1px 0px purple;

color: white;

}

</style>

<body>

<div id = 'entry'>

<img id = 'img' src="css/Pearl.png"><br>

<div id = 'ace'>VETERINARY HOSPITAL AND CLINIC</div>

<div id = 'ice'> Information management Facility </div>

</div>

<div id = 'staff' onclick = "javascript:ace('staff.php')">

<img src="pics/staff.jpg">

<div id = 'diss'> Are you a Staff?? </div>

<div id = 'text'> <p>Register or Login here to begin</p> </div>

</div>

<div id = 'doctor' onclick = "javascript:ace('doctor.php')">

<img src="pics/doctor.jpg">

<div id = 'diss'> Are you a Doctor?? </div>

<div id = 'text'> <p>Register or Login here to begin</p> </div>

</div>

</body>

</html>

**APPENDIX B: Source Code listing for Staff user page**

<?php

include\_once 'libraries/functions.php';

include\_once 'libraries/config.php';

?>

<html>

<head>

<title>VET MANAGEMENT SYSTEM</title>

<link rel="stylesheet" type="text/css" href="css/style.css">

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">

<meta name="viewport" content="width=device-width, initial-scale=1">

<script type="text/javascript" src = 'js/common.js'></script>

<script type="text/javascript" src = 'js/jquery-2.2.0.min.js'></script>

</head>

<body>

<img src="css/Pearl.png"><br>

<div id = 'ace'>PETCARE HOSPITAL AND CLINIC</div>

<div id = 'header'>

<form method = 'post'>

<?php

if(isset($\_POST['submitted'])) {

$uName = sanitize($\_POST['u\_name']);

$pass = sanitize($\_POST['pass']);

$sql = "SELECT \* FROM staff\_data WHERE username = '$uName' AND password = '$pass'";

$query = mysqli\_query($link, $sql);

if(mysqli\_num\_rows($query) == 1) {

session\_start();

$\_SESSION['u\_name'] = $uName;

header('location: patient.php');

}

else {

echo "<script> alert(Wrong Username or Password) </script>";

header('location: doctor.php');

}

}

?>

<label id = 'banner'> Staff Login </label>

<div id = 'username'>

<label> Username: </label><br>

<input type = 'text' name = 'u\_name' required>

</div>

<div id = 'password'>

<label> Password: </label><br>

<input type = 'password' name = 'pass' required>

</div>

<div id = 'sub'>

<input type = 'submit' name = 'submitted' value = 'Login'>

</div>

</form>

<div id = 'register'>

<label> No account?? <button id = 'reg'> Register Here </button></label><br><br>

<label> Forgot Password?? <button id = 'pass'> Retrieve here </button></label>

</div>

</div>

<div id = 'headers'>

<?php

if(isset($\_POST['submitted'])) {

$err\_flag = false;

if(!empty($\_POST['u\_name'])) {

$uname = sanitize($\_POST['u\_name']);

if (!check\_duplicate('username', 'u\_name', $uname)) {

$uName = $uname;

}

else {

$err\_msg[] = "Sorry, another user has already

registered with this username: $uname";

$err\_flag = true;

}

}

else {

echo 'We need a username please';

$err\_flag = true;

}

if(!empty($\_POST['f\_name'])) {

$fname = sanitize($\_POST['f\_name']);

}

else {

echo 'Your firstname cannot be empty';

$err\_flag = true;

}

if(!empty($\_POST['pass'])) {

$pass = sanitize($\_POST['pass']);

}

else {

echo 'Your account must be safe';

$err\_flag = true;

}

if(!empty($\_FILES['profile'])) {

$img = $\_FILES['profile'];

$checked\_img = upload\_image($img, $err\_msg);

}

else {

$err\_msg[] = "Please select your profile image";

$err\_flag = true;

}

if($err\_flag == false) {

$sql = "INSERT INTO staff\_data(username, firstname, profile\_image, password)

VALUES ('$uName', '$fname', '$checked\_img', '$pass')";

$query = mysqli\_query($link, $sql);

session\_start();

$\_SESSION['u\_name'] = $uName;

header('location: patient.php');

}

}

?>

<form method = 'post' enctype = 'multipart/form-data'>

<label id = 'banners'> Staff Registration </label>

<div id = 'username'>

<label> Username: </label><br>

<input type = 'text' name = 'u\_name' required>

</div>

<div id = 'firstname'>

<label> Fullname: </label><br>

<input type = 'text' name = 'f\_name' required>

</div>

<div id = 'profile'>

<label> Profile Image: </label><br>

<input type = 'file' name = 'profile' required>

</div>

<div id = 'password'>

<label> Password: </label><br>

<input type = 'password' name = 'pass' required>

</div>

<div id = 'sub'>

<input type = 'submit' name = 'submitted' value = 'Register'>

</div>

</form>

<div id = 'register'>

<label> Have an account?? <button id = 'regs'> Login Here </button></label><br><br>

</div>

</div>

<script type="text/javascript" language = 'javascript'>

var bt = document.getElementById('regs');

var btn = document.getElementById('reg');

//var span = document.getElementsByClassName('close')[0];

btn.onclick = function() {

$('#header').hide('slow');

$('#headers').show('slow');

}

bt.onclick = function() {

$('#headers').hide('slow');

$('#header').show('slow');

}

</script>

</body>

</html>

**Appendix C: Source Code Listing for Doctor User Page**

<?php

include\_once 'libraries/functions.php';

include\_once 'libraries/config.php';

?>

<html>

<head>

<title>VET MANAGEMENT SYSTEM</title>

<link rel="stylesheet" type="text/css" href="css/style.css">

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">

<meta name="viewport" content="width=device-width, initial-scale=1">

<script type="text/javascript" src = 'js/common.js'></script>

<script type="text/javascript" src = 'js/jquery-2.2.0.min.js'></script>

</head>

<body>

<img src="css/Pearl.png"><br>

<div id = 'ace'>VETERINARY HOSPITAL</div>

<div id = 'header'>

<form method = 'post'>

<?php

if(isset($\_POST['submitted'])) {

$dName = sanitize($\_POST['d\_name']);

$pass = sanitize($\_POST['pass']);

//$id = sanitize($\_POST['id\_code']);

$sql = "SELECT \* FROM doctor\_data WHERE username = '$dName' AND password = '$pass'";

$query = mysqli\_query($link, $sql);

if(mysqli\_num\_rows($query) == 1) {

session\_start();

$\_SESSION['d\_name'] = $dName;

header('location: doctor\_view.php');

}

else {

echo "<script> alert(Wrong Username or Password) </script>";

header('location: doctor.php');

}

}

?>

<label id = 'banner'> Doctor's Login </label>

<div id = 'username'>

<label> Username: </label><br>

<input type = 'text' name = 'd\_name' required>

</div>

<div id = 'password'>

<label> Password: </label><br>

<input type = 'password' name = 'pass' required>

</div>

<div id = 'sub'>

<input type = 'submit' name = 'submitted' value = 'Login'>

</div>

</form>

<div id = 'register'>

<label> No account?? <button id = 'reg'> Register Here </button></label><br><br>

<label> Forgot Password?? <button id = 'pass'> Retrieve here </button></label>

</div>

</div>

<div id = 'headers'>

<?php

if(isset($\_POST['submitter'])) {

$err\_flag = false;

if(!empty($\_POST['d\_name'])) {

$uName = sanitize($\_POST['d\_name']);

if (!check\_duplicate('username', 'd\_name', $uName)) {

$dName = $uName;

}

else {

$err\_msg[] = "Sorry, another user has already

registered with this username: $uname";

$err\_flag = true;

}

}

else {

echo 'We need a username please';

$err\_flag = true;

}

if(!empty($\_POST['f\_name'])) {

$fname = sanitize($\_POST['f\_name']);

}

else {

echo 'Your firstname cannot be empty';

$err\_flag = true;

}

if(!empty($\_POST['pass'])) {

$pass = sanitize($\_POST['pass']);

}

else {

echo 'Your account must be safe';

$err\_flag = true;

}

if(!empty($\_FILES['profile'])) {

$img = $\_FILES['profile'];

$checked\_img = upload\_image($img, $err\_msg);

}

else {

$err\_msg[] = "Please select your profile image";

$err\_flag = true;

}

if($err\_flag == false) {

$lname = rand(1,5000);

$sql = "INSERT INTO doctor\_data(username, FullName, id\_key, profile\_image, password)

VALUES ('$dName', '$fname', '$lname', '$checked\_img', '$pass')";

$query = mysqli\_query($link, $sql);

//echo "<script type = 'text/javascript'> alert(Your id\_key is $lname) </script>";

if($query) {

session\_start();

$\_SESSION['d\_name'] = $dName;

header('location: doctor\_view.php');

}

else {

?>

<script type="text/javascript">

alert('Could not register');

</script>

<?php

}

}

}

?>

<form method = 'post' enctype = 'multipart/form-data'>

<label id = 'banners'> Doctor's Registration </label>

<div id = 'username'>

<label> Username: </label><br>

<input type = 'text' name = 'd\_name' required>

</div>

<div id = 'firstname'>

<label> Firstname: </label><br>

<input type = 'text' name = 'f\_name' placeholder = ''required>

</div>

<div id = 'profile'>

<label> Profile Image: </label><br>

<input type = 'file' name = 'profile' required>

</div>

<div id = 'password'>

<label> Password: </label><br>

<input type = 'password' name = 'pass' required>

</div>

<div id = 'sub'>

<input type = 'submit' name = 'submitter' value = 'Register'>

</div>

</form>

<div id = 'register'>

<label> Have an account?? <button id = 'regs'> Login Here </button></label><br><br>

</div>

</div>

<script type="text/javascript" language = 'javascript'>

var bt = document.getElementById('regs');

var btn = document.getElementById('reg');

//var span = document.getElementsByClassName('close')[0];

btn.onclick = function() {

$('#header').hide('slow');

$('#headers').show('slow');

}

bt.onclick = function() {

$('#headers').hide('slow');

$('#header').show('slow');

}

</script>

</body>

</html>