HOSPITAL MANAGEMENT SYSTEM (HMS)

Building a Modern, User-friendly and Responsive HMS

Ernest J. Okoromi

Bachelor Thesis

The Department of Information Systems and Technology (IST)
Main field of study: International Bachelor's in Computer Engineering (B. Eng.)

Credits: 15

Semester/year: 2021

Supervisor: Patrik Österberg

Examiner: Stefan Forsström

Course code: DT099G

Lagos, Nigeria
....
Location and date

International Bachelor’s in Computer Engineering
....
Programme/Course

Ernest J. Okoromi
....
Name (all authors names)

Abstract

A Hospital Management System (HMS) is an integrated software/system or application – be it web, desktop or mobile – that handles various workflows in a Hospital. A well-built HMS manages the smooth performance of the whole healthcare system from administrative to medical, financial, legal and patient management. This is the cornerstone for the successful operation of the healthcare facility.

A well-tuned Hospital Management workflow ensures that several important decisions that affect the staffs and patients are made in a quick and most efficient manner. This can be very difficult to implement nowadays without the distinct hospital management system, especially given the huge number of patients that require critical medical attention in relation with the available facilities.

In this project, I intend to develop, implement and explore hospital management software, what functions it performs and how it will help the healthcare industry in my Country to be more effective, efficient and patient-centric.

Thus my main focus in this project will be to design a unique Hospital Management System for a private healthcare provider called **WeCare Hospital**. This system will improve hospital experience for patients, doctors and the hospital authorities. The application will run on the internet, and will be robust and fast enough to be accessed with all kinds of network systems (3G or 4G smartphone) as it will be a light and fluid application. The whole system is developed with modern web development technologies such as PHP, JavaScript, HTML5 and CSS3.

Keywords: Hospital Management System, PHP, HTML5.

Acknowledgements

I would like to use this opportunity to say thank you to all my friends and family members who have been of help to me during my studies at MIUN. Special thanks also go to my Supervisor Dr. Patrik Österberg. My appreciation also goes to all who took part in the questionnaire survey, as without their invaluable input, this thesis report would not have been possible.

Table of Contents

Terminology

AJAX	Asynchronous JavaScript and XML
CSS	Cascade Style Sheet
CSR	Client Side Rendering
HTML	Hypertext Mark-up
HMS	Hospital Management System
HTTPS	Hypertext Transfer Protocol
IMAP	Internet Message Access Protocol
LDAP	Lightweight Directory Access Protocol
PHP	Hypertext Pre-processor
POP3	Post Office Protocol
RDBMS	Relational Database Management System
SQL	Structured Query Language
TCP/IP	Transfer control protocol/Internet Protocol
URL	Uniform Resource Locator

1 Introduction

Many Nigerians have often complained about the poor state of the healthcare industry and their service delivery in the Country especially in respect with their customer relations, patient services, among other things. This is mostly due to the archaic, outdated, obsolete and poor state of the management and organization of the hospitals. They are usually poorly funded. Thus, due to paucity of funds, lack a good HMS. What I therefore decided to do, as a programmer is to build a modern, robust, fluid Hospital Management Systems in order to help our Hospitals in their daily operations.

Many hospitals in Nigeria rely on paperwork to operate and therefore, maintain huge records using this method. This manual operation is tedious and time consuming. With increase in workload and ever increasing patients requiring medical care, records, data and information being processed by hospital staffs becomes massive. This will invariably requires huge quantities of file cabinets and office space. Unfortunately, many private hospitals lack such space. Furthermore, history shows that in some special cases, when a patient is re-admitted in case of re-occurrence of an ailment, retrieval of previous records of the said patient is always difficult if not impossible. This is usually due to the disarray of the file cabinet. Meanwhile, those health-care providers with an existing HMS seem to have a system that is old, slow, non-intuitive, less interactive and buggy.

Therefore, I intend to solve this challenge by designing a system that is modern, powerful, flexible and easy to use with the end-goal of delivering considerable benefits to WeCare hospital in particular. The system will make hospital management and administrative process hassle free. This comes with the added benefits of having reliable patients' information for the purpose of providing better service delivery and support for effective decision making for patient care, hospital administration and quality management by the staffs.

1.1 Background and motivation

Hospital is a place where no one willingly wants to visit but due to unexpected health challenges, there are times when we need to. But there are several acute challenges whenever I decide to pay a visit to most hospitals in my city in Lagos, Nigeria. The first and most inconvenient experience is waiting in the queue for long hours as a result of burdensome paperwork. This old system is not very user friendly. There are lots of other problems that make our hospital experience very unpalatable in Nigeria and most developing countries. Though there are also automated hospital management system being used in some standard private hospitals but they do not provide much functionality for all stakeholders involved in the process. The system is not intuitive enough therefore it's difficult for inexperienced or novice users to operate.

I think it necessary for HMS to be efficient and robust enough in order for the patients to have an application with easily accessible information and personal data like: prescriptions, test reports and other important information. A good and modern HMS should give patients the flexibility to browse through all the doctors available according to their specialization and remotely, from the comfort of their homes, book appointment based on their needs, preference and ailment. A doctor/staff will also experience the similar benefits as already stated. This will remove paper works altogether and make things easy for everyone using the system.

The need to come up with this HMS was based on personal experience visiting **WeCare** hospital and I have also gathered experience from other people by observation and questioning. Despite the fact that people who are sick are very vulnerable and require good care and attention, yet they experience their worst nightmare when they visit the hospital. Many people have lost their dear ones in the process. Statistics have shown that in Nigeria, people seldom visit the hospital unless it is extremely necessary only after their ailment becomes critical or severe. This is due mostly to the fact already explained above. From observation, patients get frustrated and even sometimes angry as a result of cumbersome paper work or outdated and archaic HMS currently obtainable in several hospitals which do them a great deal of disservice. This gave me the perfect motivation to build something for them and offer them some help at their most vulnerable moment.

Thus, my main focus is to make peoples life easier in their hour of need. This necessitates the design of such a system that will reduce tons of paperwork and save people’s time.

1.2 Overall aim and problem statement

The objective of this project is to develop hospital management application employing the latest, best and most integrated web technologies. This system is unique for its simplicity and efficiency with the hope of migrating from file based system and old HMS to a new web based system. This software will help the health-care provider to be more efficient and responsive in handling their daily activities. Meeting this need will be a notable advancement in the industry.

This web application was designed and customized for **WeCare Hospital**, a private medical provider in Nigeria. It is to be deployed and used for the primary purpose of automating all hospital management activities and processes.

1.3 Research questions/Scientific goals/Verifiable goals

The concrete and verifiable goals of this project are;

1. To conduct a survey to know what the current features the current users of the existing platform would prefer to the existing one.
2. To design an efficient web platform for the better automation of the daily management tasks of WeCare Hospital.
3. To design a web application that allows for easy information sharing between the Admin, Nurses and Doctor in the Hospital, thus ensuring seamless running of hospital processes.
4. To build a robust system that allows for free flow of communication between the three levels of users described in section 1.4.
5. To design an easy to use system for all – novice and professional users.
6. To build a system that delivers a faster and smoother performance for all users.

1.4 Scope

In this project, my main focus was on designing a robust HMS that is user friendly, intuitive and clean. Most hospitals in my vicinity mostly use paper work or outdated desktop HMS that is heavy, buggy and not-fit-for-purpose. But the system I'm proposing will be a major upgrade over what is currently obtainable.

I have not dealt with the issues of security since I do not envisage at this stage, the possibility of hackers trying to probe the system.

During deployment, the staffs at the hospital were advised to use strong password. And only administrators have the full privilege to read, add, modify, delete, edit and update records. More advanced security features might be part of the system in the future if it so requires adding it to the website such as installing an SSL certificate to the site or using a web application firewall.

1.5 Outline

The project consists of several chapters that describe the overall developmental process of the system:

Chapter 2 presents the theory of the project; in Chapter 3 I discussed the methodology used in attaining the results; Chapter 4 shows the design concept employed in building the project; Chapter 5 present the implementation; Chapter 6 present the Results and future work, and chapter 7 discusses the conclusion. The project ends with references.

1.6 Contributions

The whole project was designed with the various modules conceived after I carried out surveys and questionnaires with the management, staffs and patients of WeCare Hospital. After much dialogue with the managements, I came to the understanding the type of system they will need to automate and enhance the quality of their health-care package for better service delivery.

After this, I interacted with the Staffs/Doctors which gave me an overall mental picture of what kind of information they would like to manage in their module; lastly, I communicated with the Patients. This particular group of users told me their priority is a system that is easy and has a one-touch access to relevant and useful information.

2 Theory

Under this section, we will now consider in depth the whole web technologies used in the project. We will give the definition and then consider how they interact with one another in a typical internet environment. After which we will take a look at the picture of web architecture. This is necessary in order to have a mental picture of the whole process of interaction between the various languages (client and server side) and how they parse information from one end to another.

2.1 The Illustrations of Web Architecture

Web Architecture comprise of two dissimilar sets of programs that run separately yet concurrently with the collective aim of working harmoniously for providing solutions to the users. Normally, the two sets of programs contain the code running in the browser which works depending on the inputs of the user and the code in the server which works in accordance with the requests of protocols (i.e. HTTPS). In other words, web programmers and/or developers need to be able to decide on the functions of the code on the server and likewise, on the browser and how these two will function in relation to each other. [1]

Fundamentals of Web Applications

A web application is a complex piece of software. It consists of many components like the user-interface, login-screen, an in-app store, database, etc. it defines the interactions between applications, middle-ware systems and databases to ensure multiple applications can work together. To manage these components, software engineers devised web application architecture to logically define relationships and manner of interactions between all of these components for a web app.

Every Web application consists of both a front-end and a back-end.

The front-end, otherwise known as the client-side of the architecture, is whole thing that the remote user sees and interacts with in their browser. The main purpose of the client-side is to collect data from users and parse it to the server or database [2]. The front-end is programmed in variants of HTML, CSS, and JavaScript.

Then, we have the back-end, which is also called the server-side of the web application. It is the part, which cannot be seen or accessed by users. This is where the data is stored and manipulated. The HTTP request is also processed at the back-end. It fetches the data – text, files, images, etc. – requested for by the users. In contrast with the front-end, there are several programming languages (PHP, Java, Python, Perl, Ruby, etc.) that can be used in coding the back-end of a Web Application. [3]

Below is a picture to describe the features of web architecture:

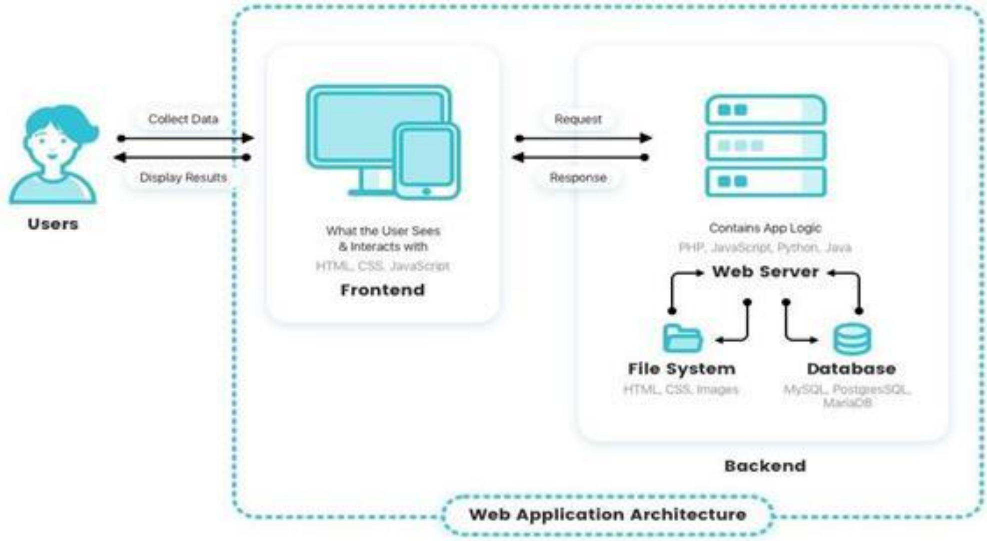


Figure 2..1: A web architecture (reinvently.com/blog/fundamentals-web-application- architecture)

2.2 Current Trends in Web Architecture

The two most popular Web Application Architectures are Server-Side Rendering and Client-Side Rendering

Server-Side Rendering: When a user visit a webpage by clicking a URL link, a request is sent to the server. After the request is done processing, the browser receives the files (HTML, CSS, and JavaScript) and then renders the content of the page. Another request is made each time the user visit or clicks another page on the website.

Client-Side Rendering: The main difference between the Server-Side and Client-Side rendering is that once you visit a Website that uses CSR, just a single request will be made to the server to load the main skeletal part of the app. Content is then dynamically generated using JavaScript.

Ethical Aspect for Good Web Application Architecture

Having a working Web Application is not good enough, one must consider the ethical aspect by following best practices in order to ensure the best experience forall users.

Security: If the source code is unsecured or unprotected, the web app can be vulnerable. This means the risk of inserting malicious code by any hacker increases considerably. Additionally, a web app must protect the private data of the users as required by privacy laws which includes Health Insurance Portability and Accountability Act of 1996 (HIPAA) for medical applications in the US and General Data Protection Regulation (GDPR) for the EU. [4]

Scalability: Developers must ensure the architecture can be scaled across several regions and through multiple servers to handle potentially major variations or changes in traffic. Today, this often results to relying on Cloud Servers. A major benefit of employing cloud services is that you pay only in proportion to the consumed bandwidth.

Separation of Services: The web application should be built with self-contained components with modularity in mind. This enables an easy addition of new features and fixing of probable errors. The app architecture should also be simple, offer fast responses, solve problems consistently and ideally have a very good up-time.

There are different kinds of web server architecture depending on the programming language used in the development: Java Web Application Architecture, Cloud-based Web Application Architecture, Node.js Web Application Architecture, PHP Web Application Architecture, Angular.JS Web Application Architecture, Laravel Web Application Architecture, .NET Web Application Architecture etc. Since PHP is the language used in this project, we will now look at the diagram of PHP web application architecture.

A typical PHP based web application looks like the diagram below;

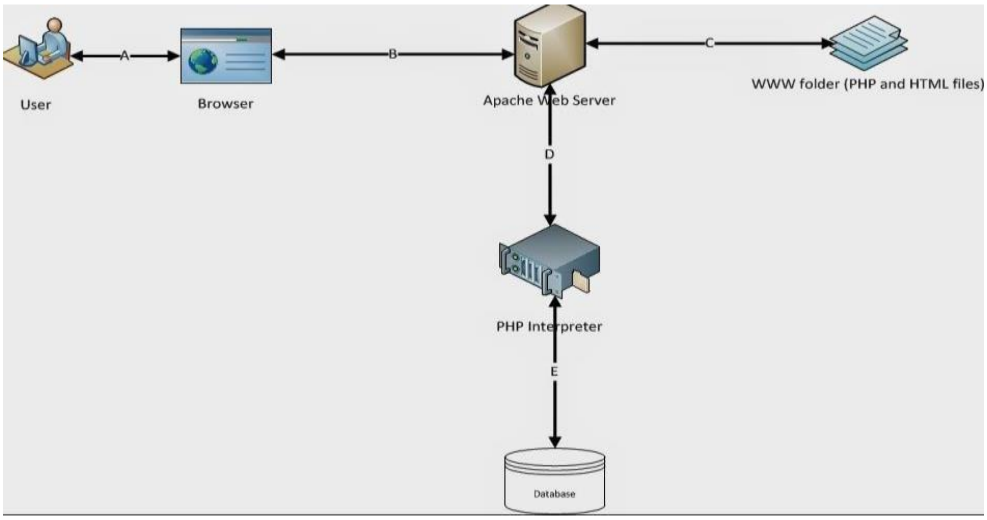


Fig. 2.2: PHP web architecture (letsdophp.blogspot.com/p/architect-ture-diagram-of-php-based-web.html)

When an internet user visit or click on a web-page, what they get to see is a specific well formatted page displayed on their browser but underneath, all these are made possible through a series of interaction between various system components of applications, databases, middle-ware systems and servers. The framework which ties up this relation and interaction together is the Web Application Architecture. The sum of what we are saying is, there is a flow of processes between these systems connected together which starts with the user clicking on a web link or URL. This follows with the browser triggering a search. Consequently, the network parses data from the server to the browser and the pagerequested for by the user is displayed. [5]

From the explanations above, we have been able to make it quite plain that a Web Application Architecture is made up of components, internal and external applications communicating together. Recently, there have been new developments in the web community. There is an ongoing progressive transition to a new domain in applications which has resulted in better, transformed capabilities in both back-end and front-end processes.

With majority of end users now migrating to the mobile domain, it is currently the preferred choice used for most internet searches. This has invariably, necessitated the need to develop web app and architecture that meets the requirement across all platform.

We will now consider the web technologies used in the project. In web development, there is a design technique that is known as modularity. This is the separation of the functionality of a programme into independent parts. This approach makes the code more organized, reusable and maintainable. Therefore, for ease of debugging and future enhancement or upgrade, the system is developed in a way as to be separated into two parts: the front-end and the back- end.

Front end: The front-end also known as client-side is built with HTML, CSS3, and JavaScript;

HTML: Hyper Text Markup Language is without a doubt the most popular or prevalent scripting language that is employed by web programmers to design web content. It is the mainstay language of the Web. HTML is actually a document just like a word processor. Open source word processor like LibreOffice Writer is used to view word documents due to the fact that it knows how to display them in a readable and editable format. In the same vein, when we talk about the web, it is the web browser that can read and display the documents built with HTML. HTML is formatted for the web browser through elements with comes with tags that are enclosed in angular brackets within the web page content. An example is the HTML opening tag (<html>) and closing tag (</html>). All scripts must within this opening and closing tag. Sometimes, though the tags may represent an empty element. They are thus, unpaired. For instance, the <img> tag. The web browser is designed to read HTML documents and transform the scripts into visible, readable or even audible web pages. The tags in the HTML documents are not shown or displayed to the viewer but are used to interpret the content that is seen by the client. The foundation and block of all web pages are formed by the HTML elements.

Images and objects can be embedded into the HTML elements in order to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages. [6]

CSS: CSS stands for Cascading Style Sheets. It is the standard style sheet designed specifically for the purpose of formatting the layout of web pages. CSS3 is the latest iteration/version of the CSS specification. It adds many new styling features and improvement to enhance the capabilities of web presentation. It primarily complements HTML by giving it a beautiful and formatted look and feel to the web pages. The HTML elements we explained in the previous section will appear quite drab and plain with default font type, size and style. But incorporating both scripting together, spices up the look with rich and colourful background images, drawing border areas, changing the page's layout, font sizes and style.

Though CSS is a completely different scripting language from HTML, yet it is impossible to use the former without the latter. In other words, HTML could be used alone to display web page to the browser – even though it will be dull – but CSS cannot be used on its own. It is impossible to write a CSS page. Rather, a CSS page is written to style and format a pre-written HTML page. [7]

JavaScript: JavaScript is a very powerful scripting language and arguably the most popular and widely used client-side language. It is designed to add great features such as interactivity and dynamic effects to web pages in order to enhance the user experience when browsing.

Originally built by Netscape Navigation in the 1990s as LiveScript, it was the changed in 1995 to JavaScript, and then became an ECMA (European Computer Manufacturers Association) standard in 1997. JavaScript is now the standard client-side scripting language for Web-based applications. It is supported by all currently available web browsers.

JavaScript is not related in any way to Java, though they share some similarities in syntax and are both object-oriented languages. It is a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles. The application of JavaScript to use outside of web pages—for example, in PDF documents, site-specific browsers, and desktop widgets—is also significant. Newer and faster JavaScript VMs and platforms built upon them (notably Node.js) have also increased the popularity of JavaScript for server-side web applications. On the client side, JavaScript was traditionally implemented as an interpreted language but just-in-time compilation is now performed by recent browsers. [8]

Back End: The back-end also known as the server-side consists of the scripting languages and the database.

PHP: PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. It stands for PHP: Hypertext Preprocessor, a recursive acronym. It actually started as a small open source project by Rasmus Lerdorf around 1994.

PHP code is interpreted by a web server. Its commands can be embedded directly into an HTML source document rather than calling an external file to process data. It is very useful in designing, managing and interacting with databases (such as MySQL, MongoDB etc.), dynamic content, session tracking and building large scale, heavy duty commercial site such as e-commerce, social networking, online streaming sites etc.

It supports huge and diverse number of major email protocols such as IMAP, LDAP and POP3. PHP can also be integrated with other major databases including but not limited to PostgreSQL, Oracle, Microsoft SQL server, Sybase and the likes. PHP is free software released under the PHP License. Though it is deployed on most web servers, yet it can also be used as a standalone shell on almost every operating system and platform. [9]

MySQL: MySQL (pronounced as: My-Es-Kyoo-El), is a relational management database system (RDBMS) with a client-server model. It stands for My Structured Query Language and was originally managed by MySQL AB, a Swedish company. It was developed Michael "Monty" Widenius, who named it after his daughter called My. Though it has since been acquired by Sun Microsystems, who was later bought by Oracle Corporation, yet it remains open source.

A database simply means a collection of structured data or a place where data is stored, managed, manipulated and organized. There are different approaches in building a database but the designer of MySQL opted for the relational database model which means that the data is broken into several separate storage areas which are called tables. All the tables in the database relate and are connected in some ways. This is

far more robust and efficient than combining all the data into one big storage unit. MySQL as a RDBMS allows the possibilities of data query (requesting information from the database), data access control (securing the data and giving permissions to who can access the data), data manipulation (performing all forms of editing or modification) and data identity (defining the data types).
Several popular web apps like YouTube, Facebook, Twitter, Yahoo!, and Google makes use of MySQL and it is now also compatible with major operating Systems such as Windows, Linux and MacOS. [10]

2.3 Related works

Huge bodies of works have been done in the same sphere of this project I am presenting. Haven taken a thorough study of similar existing systems, I discovered some drawbacks and limitations on their design and implementations. For the purpose of brevity, I have chosen three (3) similar projects for comparison with mine. These related works in focus are;

1. O. O. Lawal, B. O. Afeni, and J. O. Mebawondu, "Development of Hospital Information Management Systems". [11]

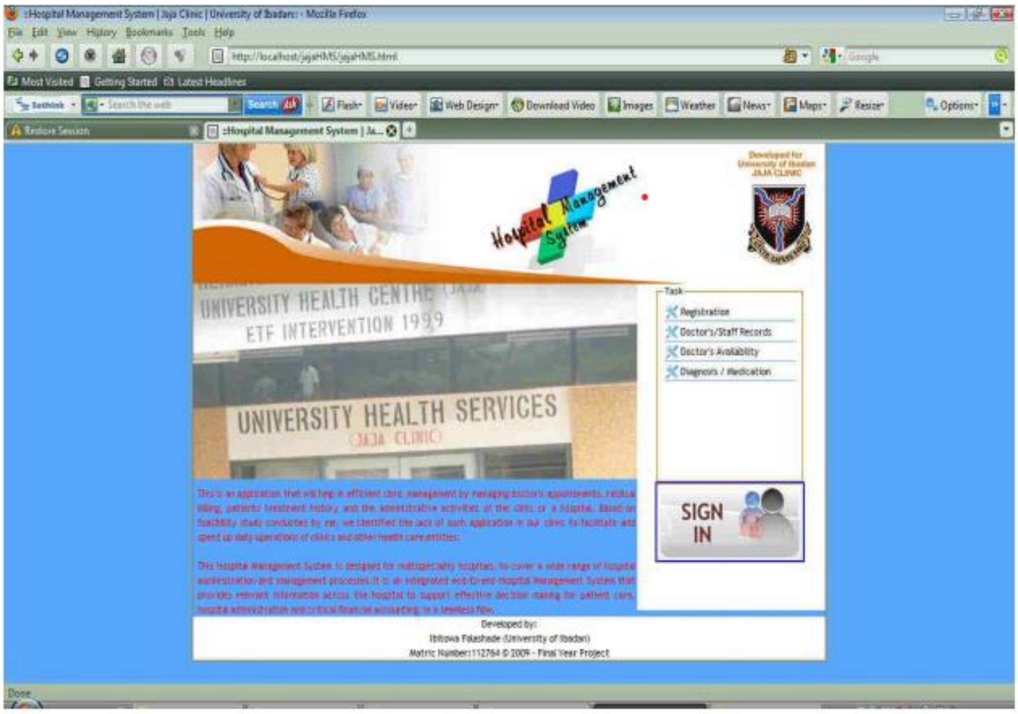
2. Ahmed Shamel Noori and Ali Fadhil Najem, "Hospital Magenememt System: Design and Implementation". [12]

3. Saad Alami, "Hospital Management Control System". [13]

2.3.1 Example of a related work

O. O. Lawal, B. O. Afeni, and J. O. Mebawondu, "Development of Hospital Information Management Systems. [11]

This is quite a simple and basic HMS implementation similar to what I am working on but the drawback in the system is lack of functionality to book for appointment remotely. This is a very important feature in a modern HMS. The ability to remotely book an appointment removes the inconvenience of having to go the hospital to do this, especially if the hospital is situated in a populated area. Though this project was also built with similar technologies as mine, yet it lacks the aesthetics and appeal found in a modern HMS.


:Hospital Management System | Jaja Clinic | University of Ibadan: - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/jajaHMS/jajaHMS.html
Most Visited Getting Started Latest Headlines
Restore Session
:Hospital Management System | Ja...
Hospital Management System
Developed for University of Ibadan JAJA CLINIC
UNIVERSITY HEALTH CENTRE
ETF INTERVENTION 1999
UNIVERSITY HEALTH SERVICES
(JAJA CLINIC)
Task
Registration
Doctor's/Staff Records
Doctor's Availability
Diagnosis / Medication
SIGN IN
Done

This is an application that will help in efficient clinic management by managing doctor's appointments, medica billing, patients' treatment history and the administrative activities of the clinic or a hospital. Based on feasibility study conducted by me, we identified the lack of such application in our clinic to facilitate and speed up daily operations of clinics and other health care entities.

This Hospital Management System is designed for multispeciality hospitals, to cover a wide range of hospital administration and management processes. It is an integrated end-to-end Hospital Management System that provides relevant information across the hospital to support effective decision making for patient care, hospital administration and critical financial accounting, in a seamless flow.

Developed by:
Ibitowa Folashade (University of Ibadan)
Matric Number:112764 © 2009 - Final Year Project

Fig. 2.3: The welcome page of the HMS built by O. O. Lawal, B. O. Afeni, and J. O. Mebawondu.

2.3.2 Another example of related work

Hospital Management System: by M. Sowmya, D. Anilchhandra Varma, M. Venugopalarao, M. Sailaja and T. Prasanth. [14]

This is a well built and implemented HMS system with many features and functionalities similar to what I am proposing. Their project is relatively of a good quality and adds some other functionality such as the ability for remote bed allotments for patients. However, as a tailored made HMS specifically for hospitals, it comes a bit short of my project. There’s no functionality such as Management and staffs receiving patients’ feedbacks. Furthermore, the ability for Doctors to maintain record history for patients is severely limited.

All these functionalities are important and added to my system.

3 Methodology

The methodology describes the steps taken from the initial project conception to the finished project. Thus, in this chapter, I will discuss and explain the methods used in attaining my results. These are outlined and elucidated below;

3.1 Interviews/Survey

After several discussion, consultation and round-table meetings with the staffs and authorities of WeCare Hospital, it was agreed that the design should meet the basic functionalities of a good and modern web application. Issues such as data integrity, accessibility, efficiency and fluidity (fast and easy loading of the web app) were to be paramount. Considering the varying background of users and the fact that many will likely access the web page using mobile web browsers, the system should be designed to be inter-operable without any lag. We then settled on using latest web technologies (PHP, JavaScript, JQuery, BOOTSTRAP, CSS3, HTML5 and MYSQL). This is due to the fact that these are the most convenient ways of building cross-platform app.

3.2 Literature Review

In order to understand the concepts associated with records management and or computer-based records management systems, it is imperative to examine and analyze published material from experts regarding the field. The purpose of this review is to analyze and examine and obtain experience as regards the creation and archival processing of electronic records. The review is based on an exhaustive assessment of the literature on computerized electronic management and electronic records, and contains an overview of the main concepts associated with the creation of an electronic records management system from the perspective of published experts.

Several excellent literary text books, articles, journals, sites and past student works and projects were consulted in designing and developing this project.

Books like "Beginning PHP and MySQL: From Novice to Professional" by W. Jason Gilmore and "PHP, MySQL, JavaScript, HTML5: All in one for Dummies" by Janet Valade and Steve Suehring were particularly every helpful. [15][16]

3.3 Paper based Prototyping

A paper base prototyping is an act of drawing or sketching the user interface of the proposed system with a paper and pen. It is a rough drawing of the whole inter-connected systems. This gives the opportunity to users to have an overview of the interface of the application and its features and also contributing to the improvement if and where necessary.

3.4 Feasibility Study

Depending on the results of the initial investigation, my initial survey after meeting with the management staffs was now expanded to a more detailed feasibility study. Feasibility study is a test of system proposal in relation to its work-ability, impact on the organization, ability to meet needs and effective use of the resources. It focuses on these major questions:

1. What are the user's demonstrable needs and how does the proposed system meet them?
2. What resources are available for the proposed system?
3. What are the likely impacts of the system on the organization?
4. Whether it is worth to solve the problem?

During feasibility analysis for this project, the following primary areas of interest were considered and some steps were taken to meet them:

Seven steps involved in the feasibility analysis are:

- Set a time line for the system development life cycle (SDLC).
- Prepare system flowcharts.
- Decide the technology to use for the proposed system.
- Define and identify characteristics of proposed system.
- Determine and evaluate performance and cost effective of each proposed system.
- Weigh system performance and cost data.
- Prepare and report final project directive to management. [17]

3.4.1 Technical feasibility

This deals with the systematic study of resource availability that may affect the ability to achieve the required or expected result. Here, I evaluate in order to determine whether the technology needed for the proposed system is available or not. The following questions were asked:

- Can the work for the project be done with the available software technology?
- Can the system be upgraded if developed?

The major area of concern under technical feasibility is with specifying equipment and software that will successfully satisfy the user requirement. The technical needs of the system include:

Front-end and back-end selection

An important issue for the development of a web-based system project is the selection of suitable front-end and back-end. When we decided to develop the project we went through an extensive study to determine the most suitable platform that suits the needs of the organization as well as helps in development of the project.

The aspects of our study included the following factors for the system:

Front-end selection:

1. It must have a graphical user interface that assists employees that are not from IT background.
2. Scalability and extensibility.
3. Flexibility.
4. Robustness.
5. Platform independent.
6. Easy to debug and maintain.
7. Event driven programming.
8. Front end to be built HTML5, JavaScript, Bootstrap and CSS3. [17]

Back-end Selection:

1. Multiple user support.
2. Efficient data handling.
3. Provide inherent features for security.

4. Efficient data retrieval and maintenance.

5. Stored procedures.

6. Operating System compatible.

According to above stated features we selected MySQL as the RDBMS coupled with PHP as the scripting language (server-side). [18]

3.4.2 Economical feasibility

Economic justification is generally the "Bottom Line" consideration for most systems. Economic justification includes a broad range of concerns that includes cost benefit analysis. In this we weigh the cost and the benefits associated with the proposed system and if it suits the basic purpose of the organization, the project then progresses to the analysis and design phase.

The financial and the economic questions during the preliminary investigation are verified to estimate the following;

- The cost to conduct a full system investigation.
- The cost of hardware and software for the type of application being considered.
- The benefits in the form of reduced cost of operation for the organization.
- This feasibility checks whether the system can be developed with the available funds. The Hospital Management System does not require enormous amount of money to be developed depending on the features. The cost of project depends upon the functionalities being required from the hospital management [19].

3.4.3 Operational Feasibility

This is mainly related to human organizations and political aspects. The points to be considered include:

- What changes will be brought to the organization with the system?
- What organization structures are disturbed?
- What new skills will be required? Do the existing staff members have these skills? If not, can they be trained in due course of time?

The system is operationally feasible as it very easy for the end users to operate considering since it is built with simple web technology with the ease of use being the paramount consideration during development [20].

4 System Design

Under this section, we will consider the design philosophy which was employed in the development and implementation of this project while presenting pictorial diagrams that details the system functionalities and interaction between the modules.

4.1 Architecture Design

A layered architectural style which partition the application into different layers has been adopted in designing this system. This necessary for the purpose of modularity. These layers which constitute the architecture namely: infrastructure layer, data layer, application layer, communication layer and user layer as illustrated in Fig.4.6.

Below is brief explanation for each layer:

1. Network/ Infrastructure Layer: this layer include the computer hardware (desktop, Server,peripheral devices, modem, UPS, etc.). These are basically the systems that ensures constant internet connection in the facility considering the unavailability of supply of constant power in the hospitals.
2. Data Layer: this is composed of database to store logs and records of past and ongoing records;
3. Application layer: this is the HMS, the web application that provides the users with all the functionalities and modules that users of the system interacts with.
4. Communication layer: define communications carriers.
5. User layer: represents the user of the systems [21].

4.2 Use Case Diagram (UCD)

A use case diagram is a Unified Modelling Language (UML) that gives a pictorial representation of the functionalities of the system available to the different users and how they interact with it. This is defined by and created from a use-case analysis. It gives us a graphical over-all view of the inter-connection of the modules in the HMS.

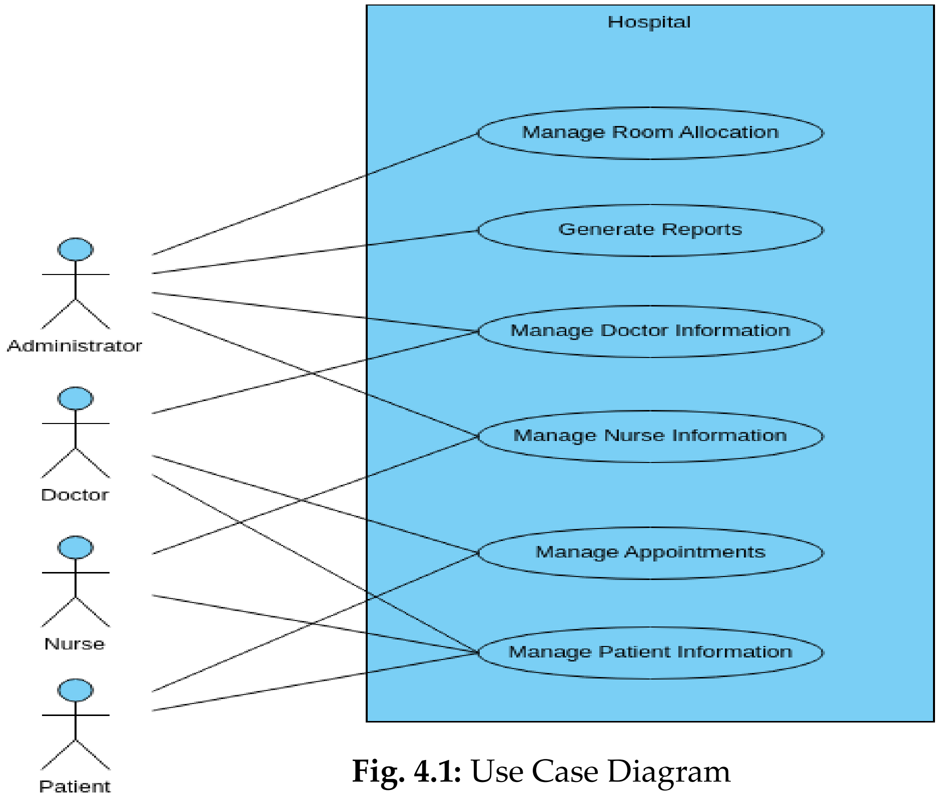


Fig. 4.1: Use Case Diagram

4.3 Data Flow Diagram (DFD)

This is a graphical image of the flow of HMS platform. It is used for visualization of data processing. It displays the communication between a system and its foreign bodies. Of which the context-level DFD is broken-down to display more information of the proposed HMS platform. DFD is generally used during problem inquiry or analysis. Below is a DFD showing the channel of flow of data through different processes in the HMS platform.

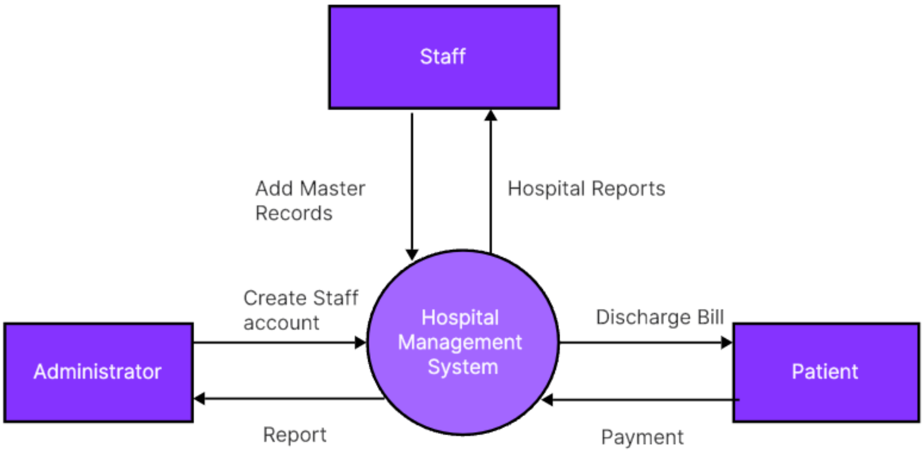


Fig. 4.2: Context or Zero level DFD

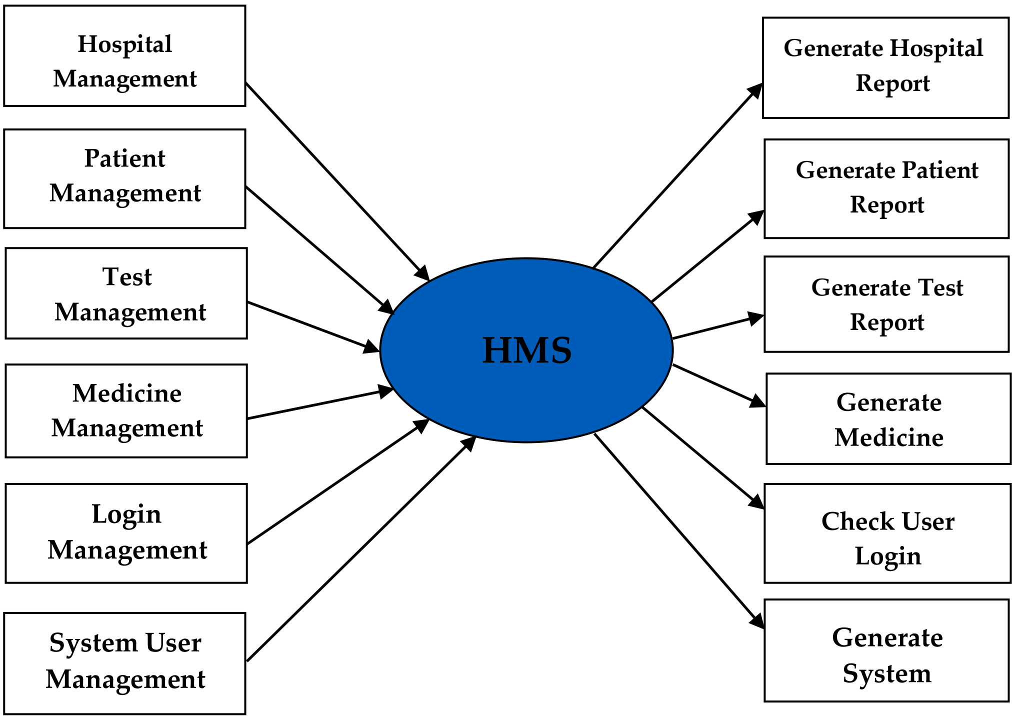


Fig. 4.3: First Level DFD

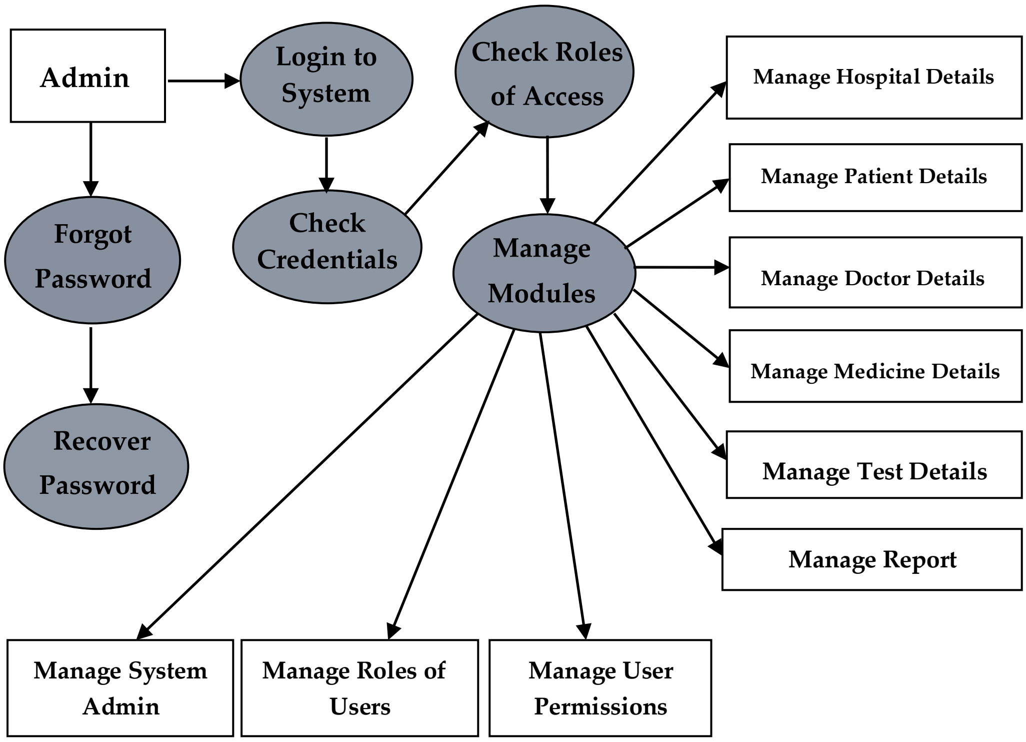


Fig. 4.4: Second Level DFD

4.4 User Design Interface

There are many factors that must be considered when designing the user interface of a soft-ware because the user must be able to interact with the system in a way that the system will understand whatever the input given by the user. Therefore, the quality of the interface and software in general must pass the usability testing standard. Some usability factors, such as fit for use, ease of learning, task efficiency, ease to remember, and subjective satisfaction must all be put into consideration when designing the user interface. [22]

Fit for use deals with the system's task support that the user has in the real life. The other factors only deal with the ease of use or how user friendly the system is. Making user interaction as simple as possible when designing the user interface is very important and this must be considered during functional requirement phase of a software design. To create an operational, usable and user friendly interface, the technical functionality with visual element must be put into consideration.

In actual fact, there are ten fundamental "heuristics" principles that must be followed when a user interface program is to be designed. The following are some of the general principles stated by Jakob Nielsen in his book titled "The ten usability heuristics". [23] We will only discuss seven (7) that are more suitable for the purpose of our project. The usability heuristics are;

1. User Control and Freedom

User often make mistake by clicking buttons which are not relevant to his/her current task, therefore a clearly mark "exit" or "cancel" button should designed on the interface without having to go through a tedious or lengthy process just to leave an unwanted page. Or better still, "undo" and "redo" should to place on the interface.

2. Consistency and Standards

Platform conventions are important part of user interface design because user should be able to know that a word that is read before still means the same thing when the same is encountered on the site or web page. Failing to follow or maintain consistency in usage of words may increase the users' cognitive load thereby leading to confusion. Thus, when a user interacts with your system, they should have no doubts about the meaning of icons, symbols or words used by the developer.

3. Error Prevention

It is better to carefully design software where there is no error in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option through a dialog box before they click the button. For instance, if a user mistakenly hit a "delete" button, he/she should receive a warning message to confirm the decision before going through with the action.

4. Recognition rather than Recall

The user should not be forced to remember what he/she had seen before from one part of the dialogue to another. Objects, actions and many other options should be made very visible so that the user will easily remember while instructions for use of the system should be visible or easily retrievable whenever appropriate.

5. Aesthetic and Minimalist Design

Information which is not relevant or rarely needed to the dialogue should not be placed on the interface. Unnecessary information might overload the page and compete with the relevant information thereby diminishing their visibility. In order words, the content and visual elements of the interface should focus on the essentials and support the users' primary goal of visiting the page.

6. Help users Recognize, Diagnose, and Recover from Errors

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

7. Help and Documentation

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large. [23]

ER Diagram

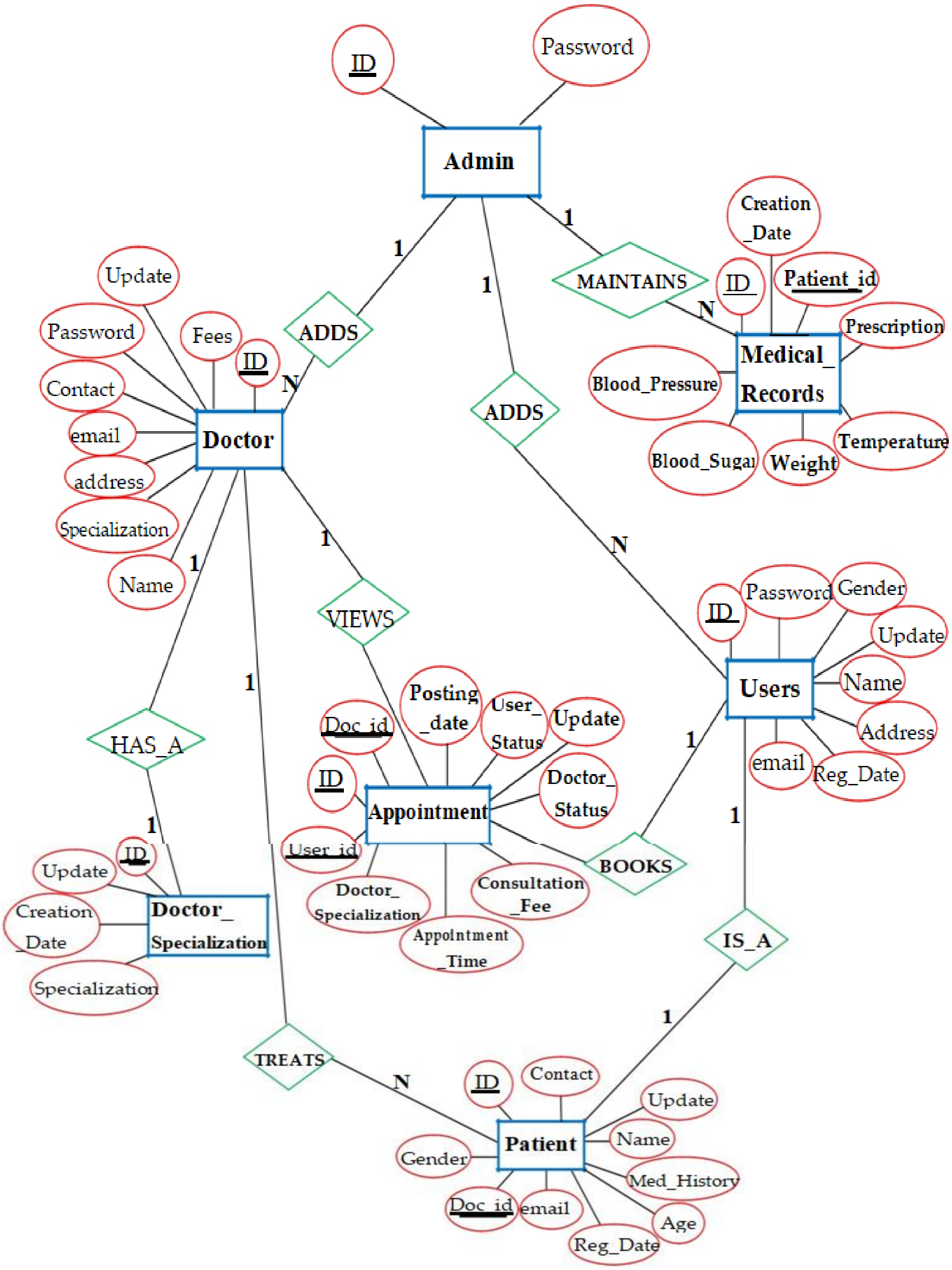


Fig. 4.5 displays the entity relationship diagram between entities and their attributes

Symbols	Meaning
—	Primary Key
▭	Entity
◇	Relationship
○	Attributes

ER Diagram Descriptions

There are three major types of relationship between entities as seen in the ER diagram.

These includes one-to-one (1 to 1), one-to-many (1 to N), many-to-many (N to N).

- One-to-one (1 to 1): This type of relationship define the relation between one instance of an entity (A) that is related to a particular other instance of another entity

(B). For example for a hospital with many patients, every patient has a name and each name (A) is related with only one identification number (id) (B).

- One-to-many (1 to N): This type of relationship define the relation between one instance of an entity (A) that is related with one, zero or many instances of another entity (B), however for a particular instance of entity B there is only one instance of entity A. For instance, a hospital may have one admin but many doctors and patients. Thus, the admin alone serves as the system administrator and therefore has the privileges and permission to adds/registers several doctors and patients and manage their schedule.

- Many-to-many(N to N): This relationship type define the relation between one instance of an entity(A) that is related with one, zero or many instances of another entity(B), and one instance of entity B is related with one, zero or many instances of entity A. In large hospital settings, for instance, it is possible for many doctors to have several patients, vice versa.

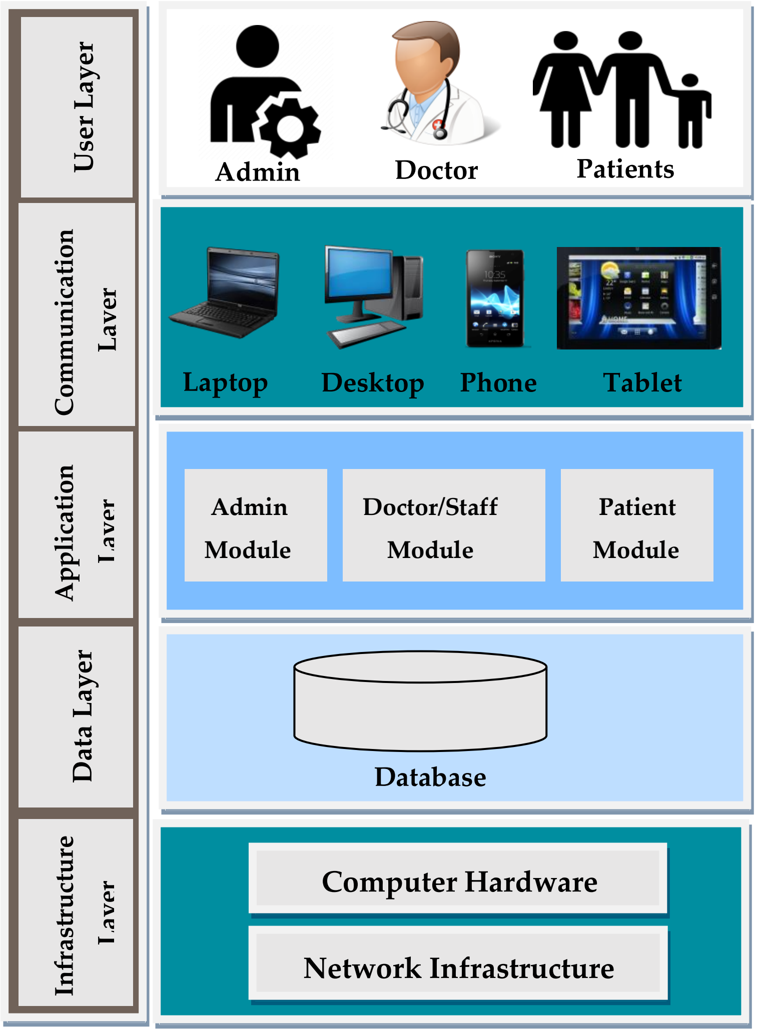


Fig. 4.6: Architectural Design

5 Implementation

Important information we put into consideration while designing the system is the fact that most users will certainly access the site through their mobile devices due to dearth of broadband connectivity and since the mobile data providers in Nigeria can sometimes be unreliable, it is necessary therefore that we use modern software development tools. With these, the users will be able to access the site without much problem unlike the system currently available in the healthcare industry in my city. For the purpose of modularity and ease of developing this system, we decided to separate the various users into the following parts:

User Groups

There are four categories for users. The features and functionalities available to a specific user will vary depending on the user type. They will be able to perform certain task in a category which may not be available for other categories. For instance, the admin has higher privileges and permissions to all sensitive information which is not available to other users. The Doctor can view patients' medical history; such data is not available for the patient users.

Below are the various categories of user groups;

Administrator: The admin has the full privilege and permissions to all the system's functionalities. They have the ability to create user profile both for doctors, patients, and staffs.

Doctor: Doctors can check appointments and also set appointment by their own. They can start prescribing drugs to their patients by clicking the appointment list. It will directly lead them to the prescription page. The prescription page has a lot of automated features for the doctor to set things and assign to their patients. Every field that is necessary for a doctor to make a perfect prescription is available. Which will obviously reduce the workload of doctors and they can give proper time to patients.

Staffs: These are other personalities in the hospital such as nurses, auxiliary nurses, receptionist, pay-roll staffs etc. They will have the permission to view patients' details and be responsible for approving appointments and monitor the progress of the whole process.

Patients: Patients are normal users like us. They will be able to browse through available doctors and book for appointments. They can also check their medical history. The patient's functionality is limited based on their need. They will only be able to change their sensitive information without the permission of the admin.

Naturally, there will be 3 modules which correspond to the size of the user-group. These are the;

- Admin module
- Doctor/Staff module
- Patient module

5.1 Admin Module

Dashboard: In this section, admin can view the Patients, Doctors, Appointments and New queries.

Doctors: In this section, admin can add doctor's specialization and mange doctors (Add/Update).

Staffs: In this section, admin can add staffs to the system.

Patients: In this section, admin can view patient's details.

Appointment History: In this section, admin can view appointment history.

Contact us Queries: In this section, admin can view queries which are sent by users.

Doctor Session Logs: In this section, admin can see login and logout time of doctor.

User Session Logs: In this section, admin can see login and logout time of user.

Reports: In this section, admin can view reports of patients in particular periods.

Patient Search: In this section, admin can search patient with the help of patient name and mobile number.

Admin can also change his/her own password.

5.2 Doctor/Staff Module

Dashboard: In this section, doctor can view his/her own profile and online appointments.

Appointment History: In this section, Doctor can see patient's appointment history. Patients: In this section, doctor can manage patients (Add/Update).

Search: In this section, doctor can search patient with the help of patient name and mobile number.

5.3 Patient Module

Dashboard: In this section, patients can view his/her profile, Appointments and Book Appointment.

Book Appointment: In this section, Patient can book his/her appointment.

Appointment History: In this section, Patients can see his/her own appointment history.

Medical History: In this section, Patients can see his/her own appointment history.

User can update his/her profile, change the password and recover the password

6 Results

Developing a Hospital Management System (HMS) could be a formidable task depending on the size of the project and the scale of information it contains. Putting these into consideration, I decided to employ a minimalist design philosophy that caters for the need of all users. This was achieved by focusing on the necessary information needed by each user group. The admin module is the most comprehensive of the three modules. This is not far-fetched since it is he/she who manages the account of other users (Doctors and patients). Despite the foregoing assertion, the whole project remains patient-centric.

The goal of this HMS was to build a robust, efficient and high-performing application that easily meet the needs of all the users, with the use of powerful scripting languages such as HMTL5, CSS3, PHP, and JavaScript using XAMPP as the web-server. The seamless interoperability and performance of these scripting languages proved productive in achieving the set goal of this project.

I will now show the final end result of the whole project with all the modules by displaying pictures, so we can see how it was built and designed and how the various categories of users practically interact with the system. I will now dive into the different features in each module, while considering in-depth the available functionalities;

I will start with the homepage, which is the beginning point of every webpage;

Homepage:

The Homepage is always the first point of contact when users want to access a web page. True to our minimalist design philosophy, the homepage contains very few information with just five buttons – Home, Contact, Admin Login, Doctor’s Login and Patient’s Login.

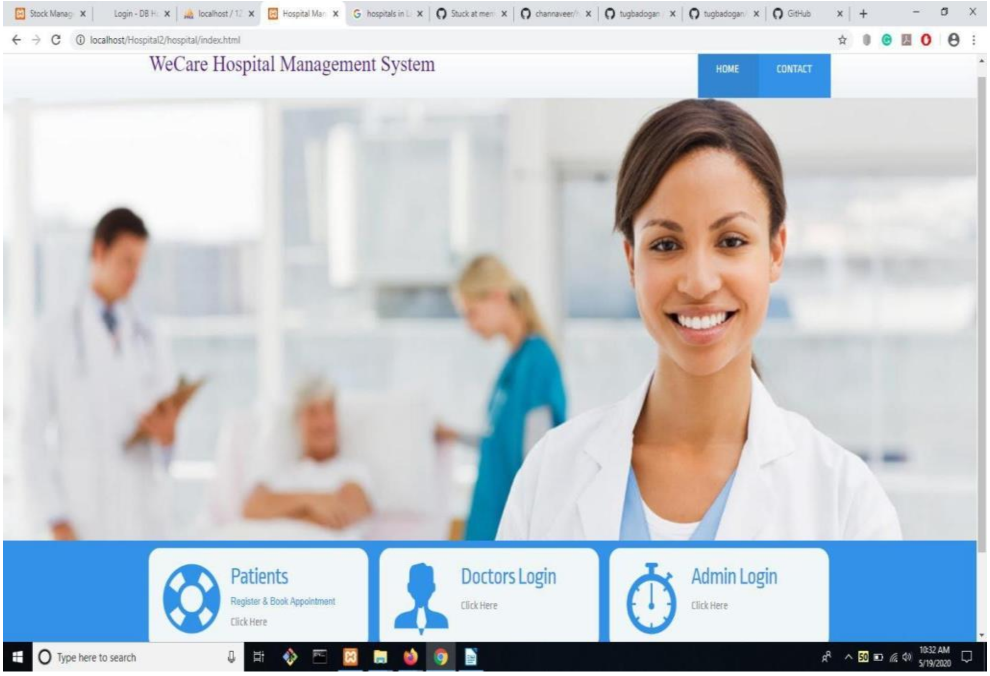
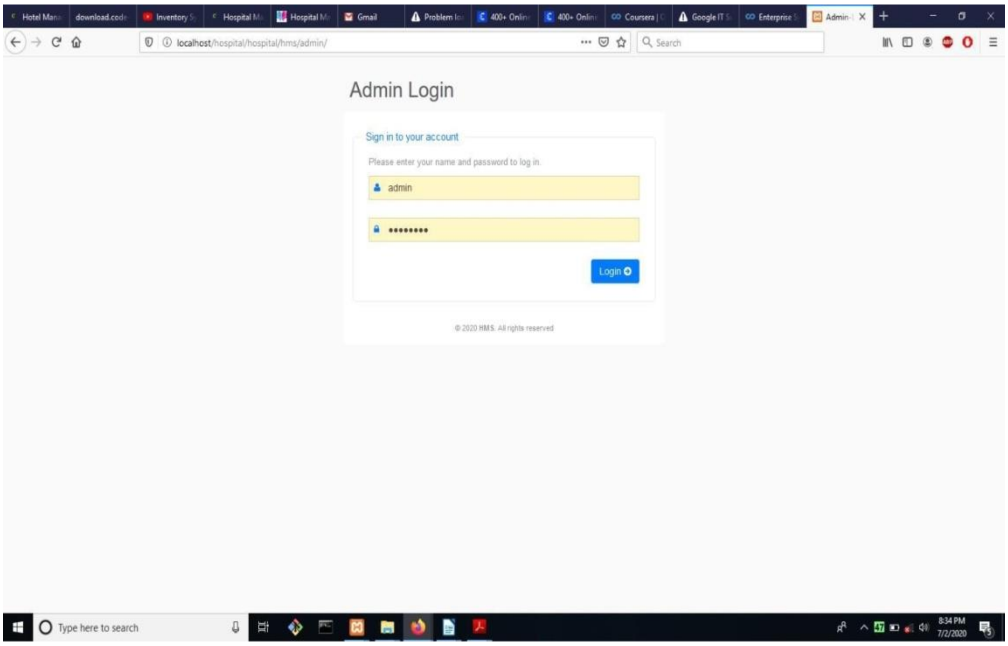


Fig. 6.1: The Homepage of our System

Admin Module

The admin is the super-user with all privileges. He oversees and manages the other Users.



Admin Login

Sign in to your account

Please enter your name and password to log in.

admin

••••••••

Login

© 2020 HMS. All rights reserved

Fig. 6.2: The Admin login page

From the homepage, the admin logs into his administrative page. From here he carries out his varied duties of administration. The page consists of the dashboard, which contains all the information. From the here, the admin is able to see the current number of doctors employed by the hospital, users registered, total number of appointments logged, the doctors working on the patients etc.

Also, he/she is able to manage all users such as adding or modifying Doctor's and patient's profile, check session logs and reports from both Doctors and patients, check queries and also a little search engine to search for patients and so much more.

The dashboard also contains a style selector, where the admin can change the pattern of the background as he sees fit. There are several built-in aesthetic background programmed into the dashboard.

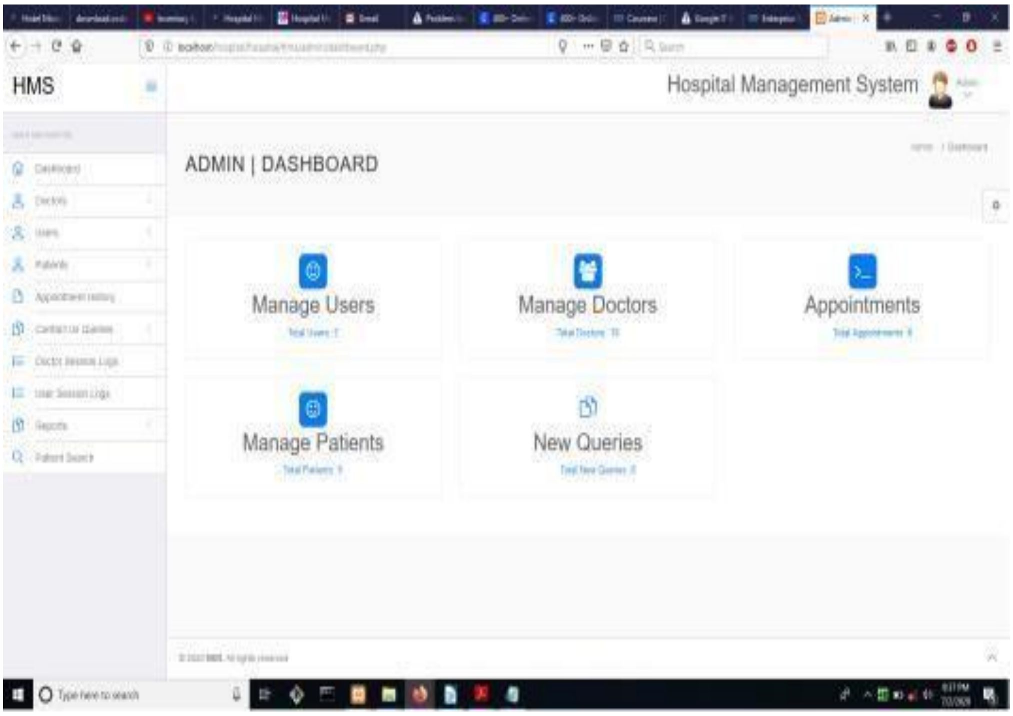


Fig. 6.3: The administrator's dashboard

In the above image, it is clear that the admin is managing 7 Users, 5 patients and 10 Doctors. There are 6 appointments and 0 queries.

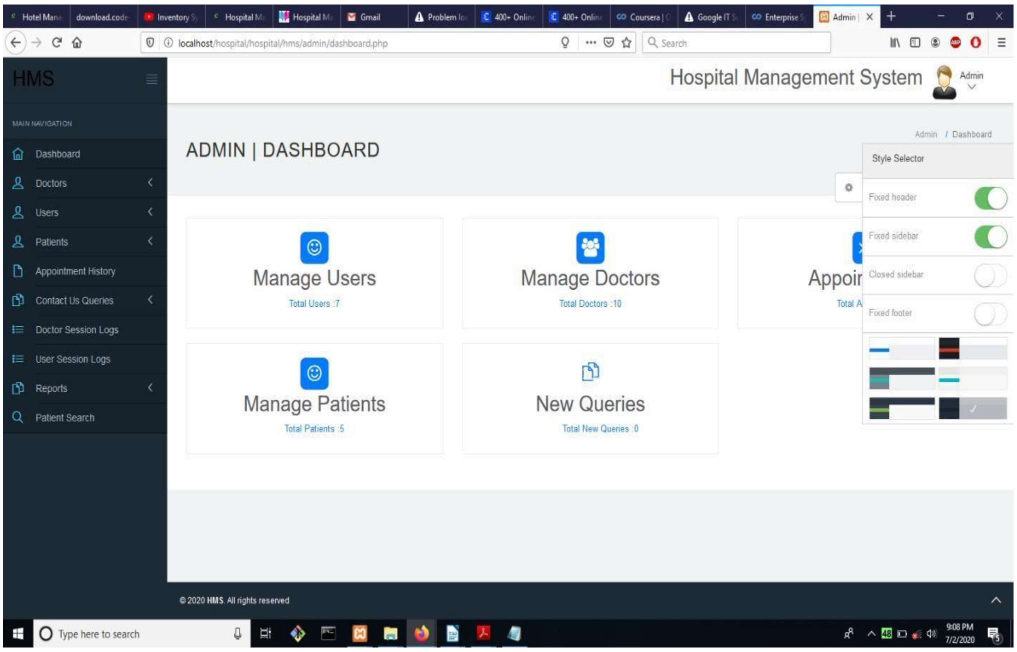


Fig. 6.4: Using the style selector, the admin can change the background foraesthetic appeal and feel.

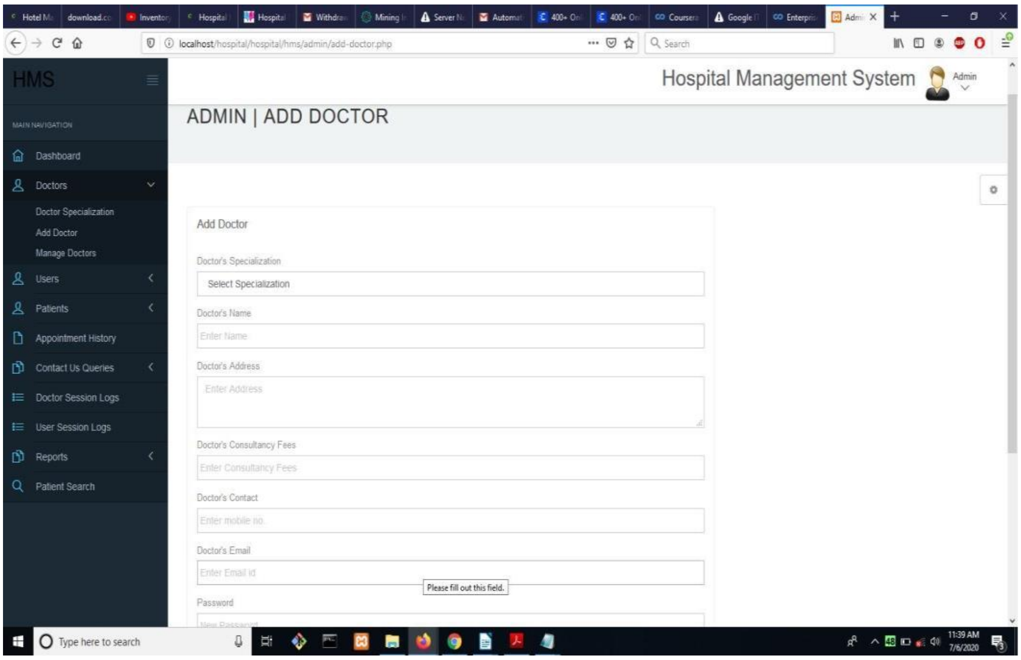
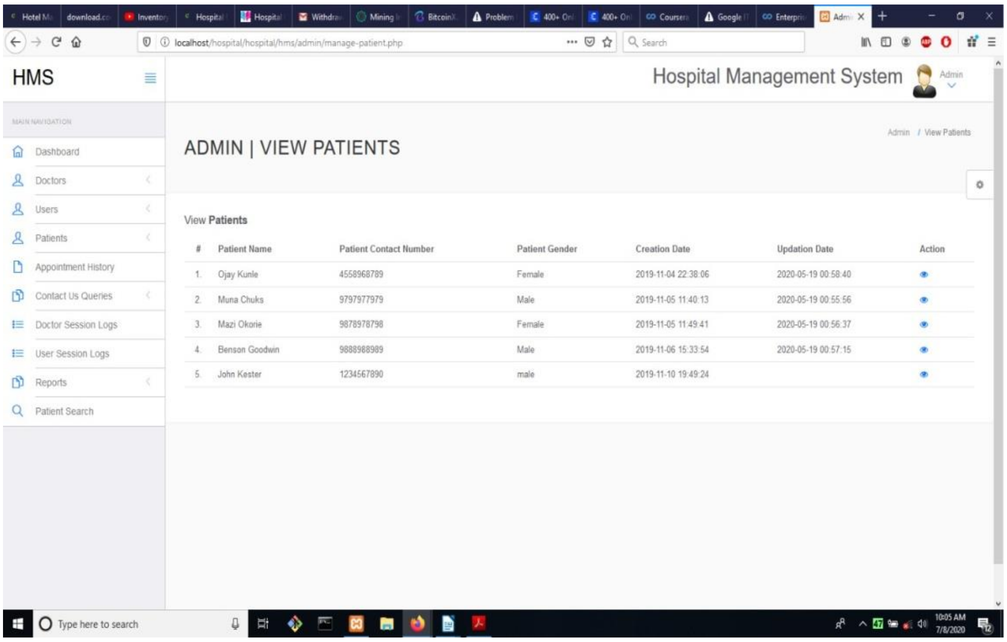


Fig. 6.5: shows the page where the admin adds the data of new doctors to the system.

This is another important responsibility for the admin. When a doctor is employed by the hospital, the admin have to add the doctor's account in details. There are various things for a doctor. Under personal details, doctor information such as specialization, name, address, email, password, contact number and other details for the doctor must be added. After successful insertion of a doctor, the admin can assign them patients and appoint them for requests. All those information from every tab needs to be inserted carefully as it will show up with the doctors details. If they insert less information then the patients will not be able to see all information of the doctor. This can lead to a situation where the system may not provide the required benefits for the patient. So the admin must exercise utmost caution when registering a doctor into the system.

Patient Management

The admin also manages the patients' details and history. He maintains their records and ensures security and confidentiality. Important personal details such as name, email, contact number gender, age, contact address can be retrieved here. The records here can also be edited and updated when there is a change of information from the patient.



ADMIN | VIEW PATIENTS

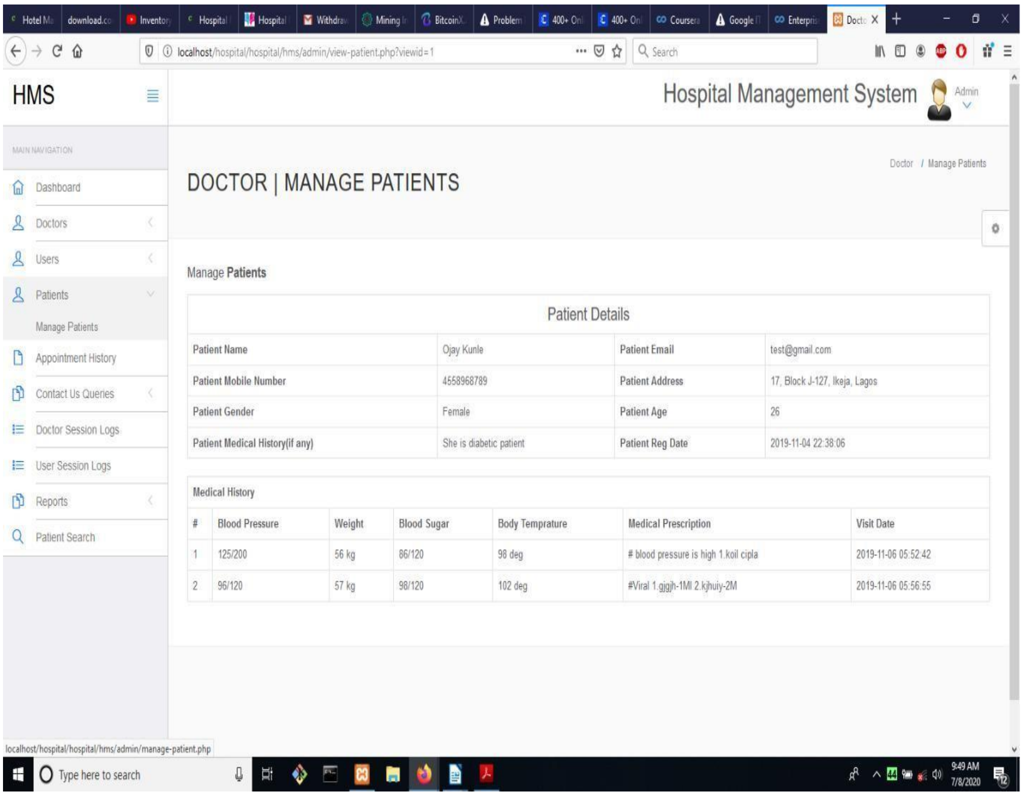
View Patients

#	Patient Name	Patient Contact Number	Patient Gender	Creation Date	Updation Date	Action
1.	Ojay Kunle	4558968789	Female	2019-11-04 22:38:06	2020-05-19 00:58:40	👁
2.	Muna Chuks	9797977979	Male	2019-11-05 11:40:13	2020-05-19 00:55:56	👁
3.	Mazi Okorie	9878978798	Female	2019-11-05 11:49:41	2020-05-19 00:56:37	👁
4.	Benson Goodwin	9888988989	Male	2019-11-06 15:33:54	2020-05-19 00:57:15	👁
5.	John Kester	1234567890	male	2019-11-10 19:49:24		👁

Fig. 6.6: shows the admin viewing and monitoring patients' data.

The admin also manages the patients' details and history. He maintains their records and ensures security and confidentiality. Important personal details such as name, email, contact number gender, age, contact address can be retrieved here. The records here can also be edited and updated when there is a change of information from the patient.

Others details are the medical history of each patient such as blood pressure, weight, blood sugar, body temperature, medical prescription and visit date.



DOCTOR | MANAGE PATIENTS

Manage **Patients**

Patient Details

Patient Name	Ojay Kunle	**Patient Email**	test@gmail.com
Patient Mobile Number	4558968789	**Patient Address**	17, Block J-127, Ikeja, Lagos
Patient Gender	Female	**Patient Age**	26
Patient Medical History(if any)	She is diabetic patient	**Patient Reg Date**	2019-11-04 22:38:06

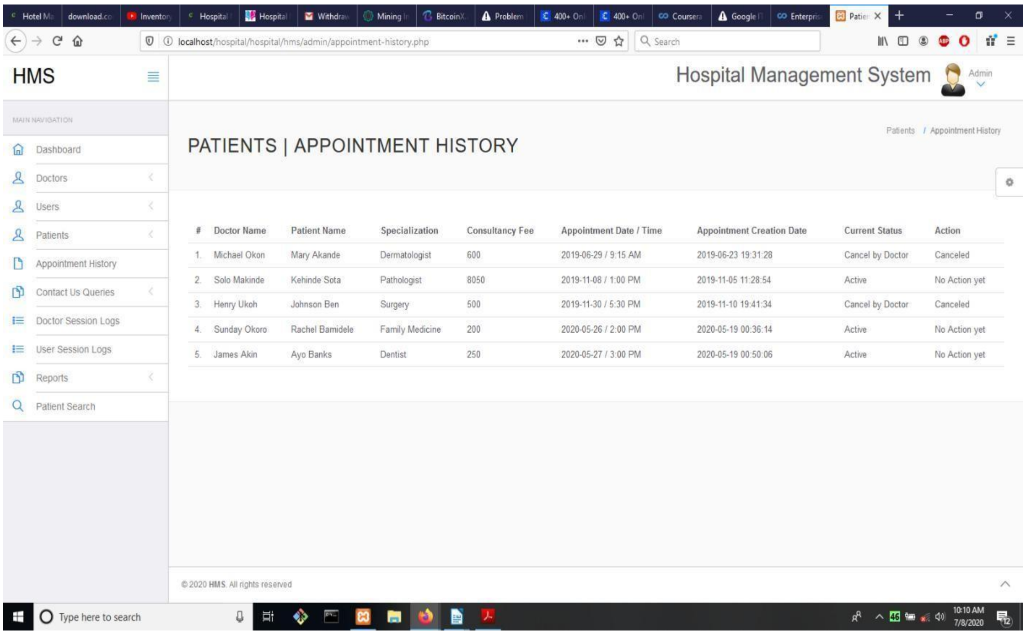
Medical History

#	Blood Pressure	Weight	Blood Sugar	Body Temprature	Medical Prescription	Visit Date
1	125/200	56 kg	86/120	98 deg	# blood pressure is high 1.koil cipla	2019-11-06 05:52:42
2	96/120	57 kg	98/120	102 deg	#Viral 1.gjgjh-1Ml 2.kjhuiy-2M	2019-11-06 05:56:55

Fig. 6.7: The admin monitors patient's medical records.

The admin also maintains and monitors all appointments. He ensures that appointments are kept by the patients by contacting them and also reminds the doctors to meet with their patients. Any appointment missed, kept or canceled is indicated under the **status** bar. With this the hospital is able to manage the performance of their staffs and take measures to improve the services rendered.

The doctor responsible for each patient, their consultancy fee and appointment date and time are seen in the image below:



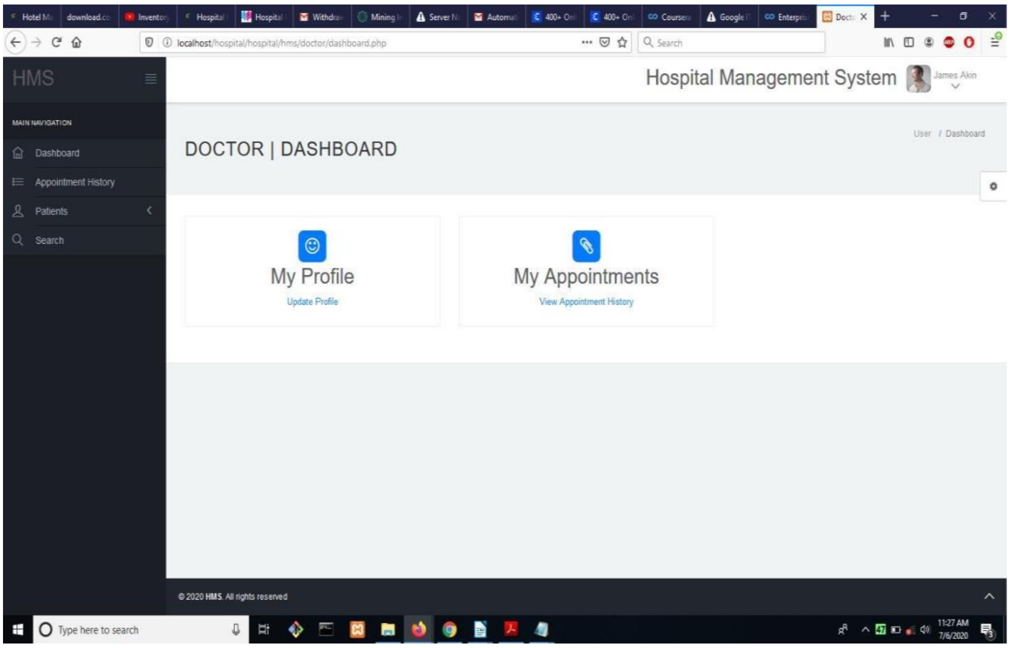
PATIENTS | APPOINTMENT HISTORY

#	Doctor Name	Patient Name	Specialization	Consultancy Fee	Appointment Date / Time	Appointment Creation Date	Current Status	Action
1.	Michael Okon	Mary Akande	Dermatologist	600	2019-06-29 / 9:15 AM	2019-06-23 19:31:28	Cancel by Doctor	Canceled
2.	Solo Makinde	Kehinde Sota	Pathologist	8050	2019-11-08 / 1:00 PM	2019-11-05 11:28:54	Active	No Action yet
3.	Henry Ukoh	Johnson Ben	Surgery	500	2019-11-30 / 5:30 PM	2019-11-10 19:41:34	Cancel by Doctor	Canceled
4.	Sunday Okoro	Rachel Bamidele	Family Medicine	200	2020-05-26 / 2:00 PM	2020-05-19 00:36:14	Active	No Action yet
5.	James Akin	Ayo Banks	Dentist	250	2020-05-27 / 3:00 PM	2020-05-19 00:50:06	Active	No Action yet

Fig. 6.8: The image above displays the admin's view of patients' appointments with their Doctors.

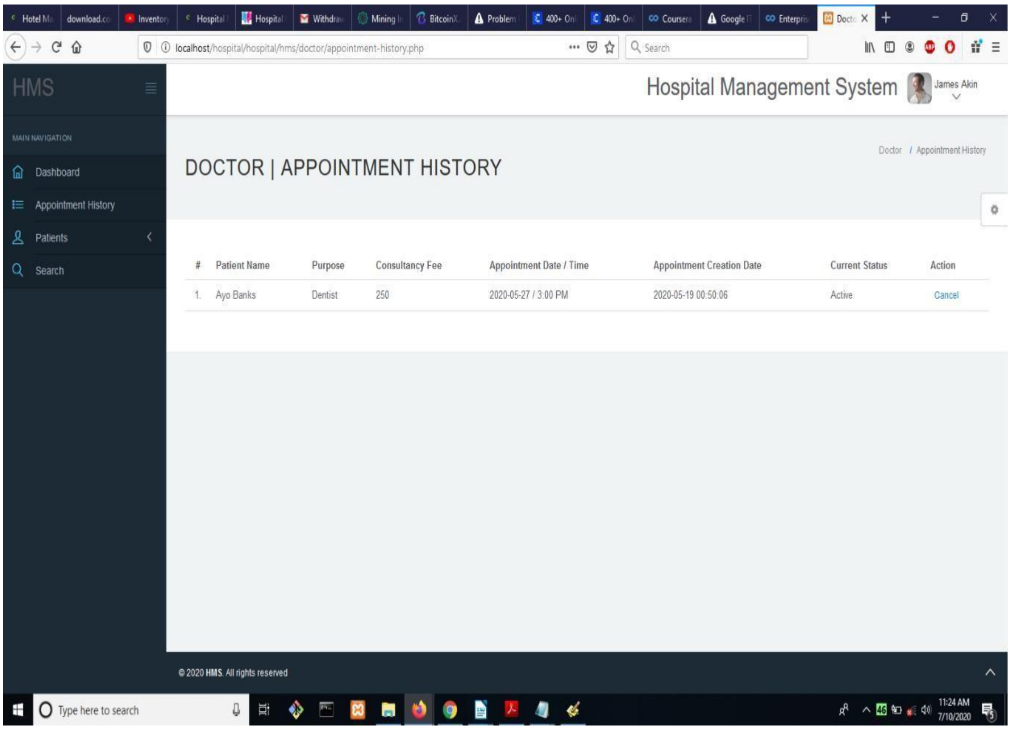
There are other functionalities in the admin dashboard, but we I restricted to explaining the most important for brevity.

Doctor's Module



DOCTOR | DASHBOARD

After logging in, doctors will directly see their profile and appointments in the dashboard. The above dashboard displays the page of a Doctor.


localhost/hospital/hospital/hms/doctor/appointment-history.php
HMS
MAIN NAVIGATION
Dashboard
Appointment History
Patients
Search
Hospital Management System
James Akin
Doctor / Appointment History
© 2020 HMS. All rights reserved
Type here to search
11:24 AM
7/10/2020

DOCTOR | APPOINTMENT HISTORY

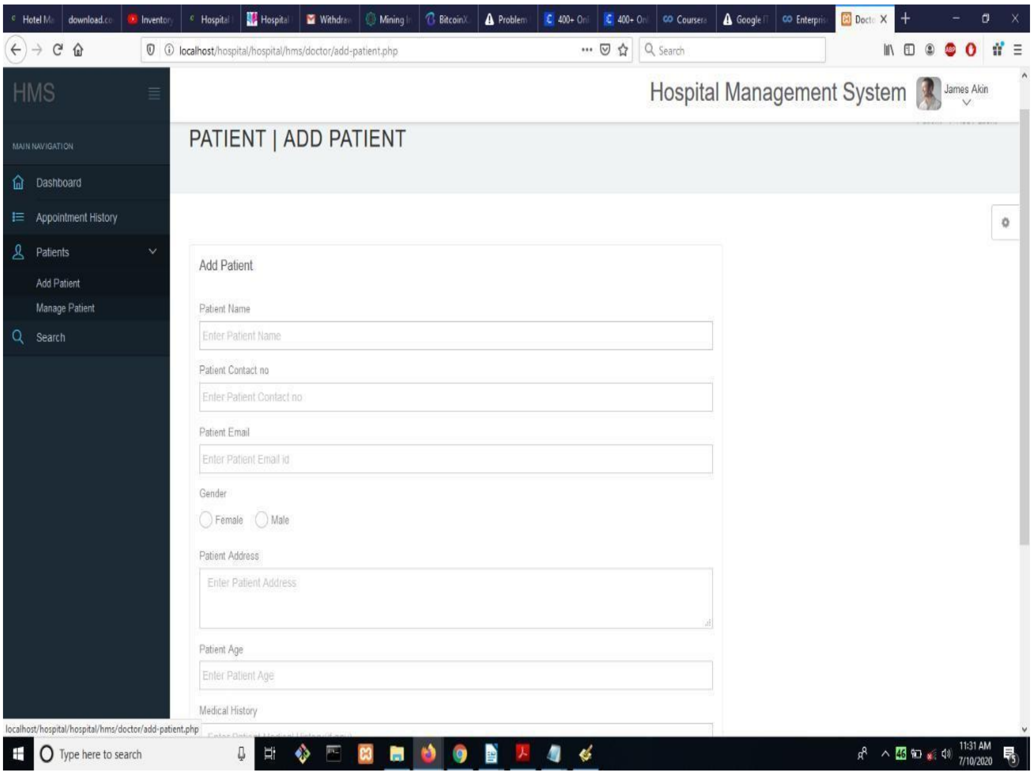
#	Patient Name	Purpose	Consultancy Fee	Appointment Date / Time	Appointment Creation Date	Current Status	Action
1.	Ayo Banks	Dentist	250	2020-05-27 / 3:00 PM	2020-05-19 00:50:06	Active	Cancel

Fig. 6.10: A Doctor's appointment page with a patient.

The second navigation button in the left pane in the Doctor's dashboard is Appointment History. From here the doctor is able the check his/her next patient with the date and time and other necessary information needed by the doctor as regards the patient as seen in Fig. 4.10 above.

By clicking on the patient's pane, the doctor is able to add a patient to add and/or manage patients. The doctor needs to add patients he/she is currently consulting with or treating for future reference.

The image below shows how doctors add patients:



PATIENT | ADD PATIENT

Add Patient

Patient Name

Enter Patient Name

Patient Contact no

Enter Patient Contact no

Patient Email

Enter Patient Email id

Gender

○ Female ○ Male

Patient Address

Enter Patient Address

Patient Age

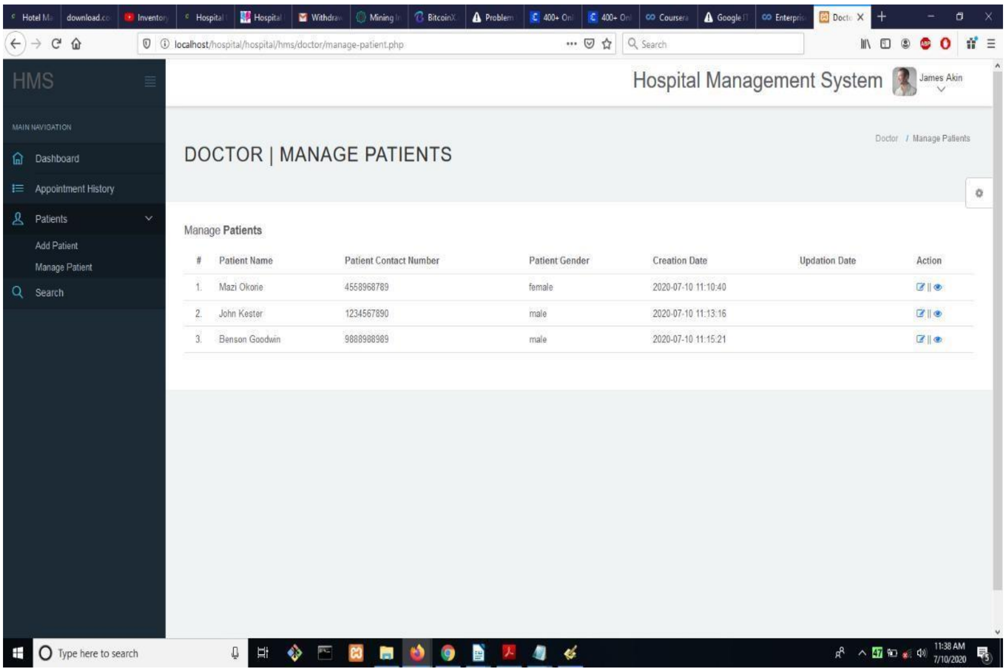
Enter Patient Age

Medical History

Fig. 6.11 shows the functionality used in adding patients.

The add button ensures the doctor will also be able to set appointment by himself like the admin. If a doctor sets appointment by himself then it will not remain pending. It does not have anything to do with the management to approve or anything else. This will make future meeting between them straight forward since the doctor is already being consulted by the patient.

From the image below, it is obvious that a doctor is now managing three (3) patients. The doctor can also check the current patients he/she is currently administering treatments by clicking on the **Manage Patient** button. Again in the following image, the Doctor can perform two actions – view and/or update patient's information. To do either of these tasks, the doctor simply click on one of the button under the Action. To view the information of the first patient for instance, the following page will display:

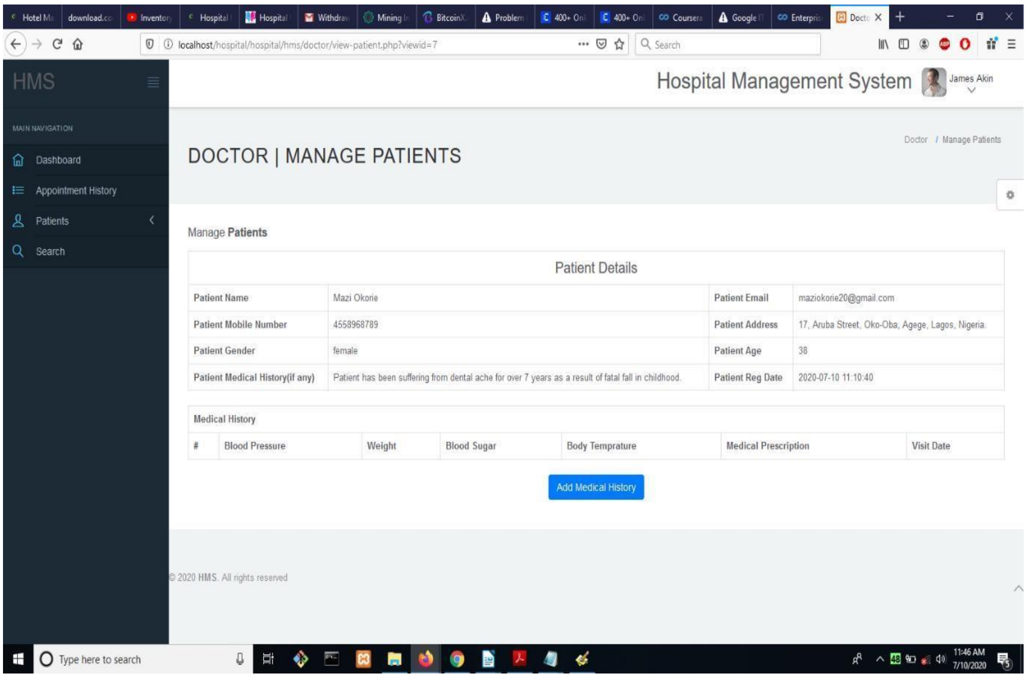


DOCTOR | MANAGE PATIENTS

Manage **Patients**

#	Patient Name	Patient Contact Number	Patient Gender	Creation Date	Updation Date	Action
1.	Mazi Okorie	4558968789	female	2020-07-10 11:10:40		✎ || 👁
2.	John Kester	1234567890	male	2020-07-10 11:13:16		✎ || 👁
3.	Benson Goodwin	9888988989	male	2020-07-10 11:15:21		✎ || 👁

Fig. 6.12 shows the page doctors manages the data of their patients.



DOCTOR | MANAGE PATIENTS

Manage **Patients**

Patient Details

Patient Name	Mazi Okorie	**Patient Email**	maziokorie20@gmail.com
Patient Mobile Number	4558968789	**Patient Address**	17, Aruba Street, Oko-Oba, Agege, Lagos, Nigeria.
Patient Gender	female	**Patient Age**	38
Patient Medical History(if any)	Patient has been suffering from dental ache for over 7 years as a result of fatal fall in childhood.	**Patient Reg Date**	2020-07-10 11:10:40

Medical History

#	Blood Pressure	Weight	Blood Sugar	Body Temprature	Medical Prescription	Visit Date

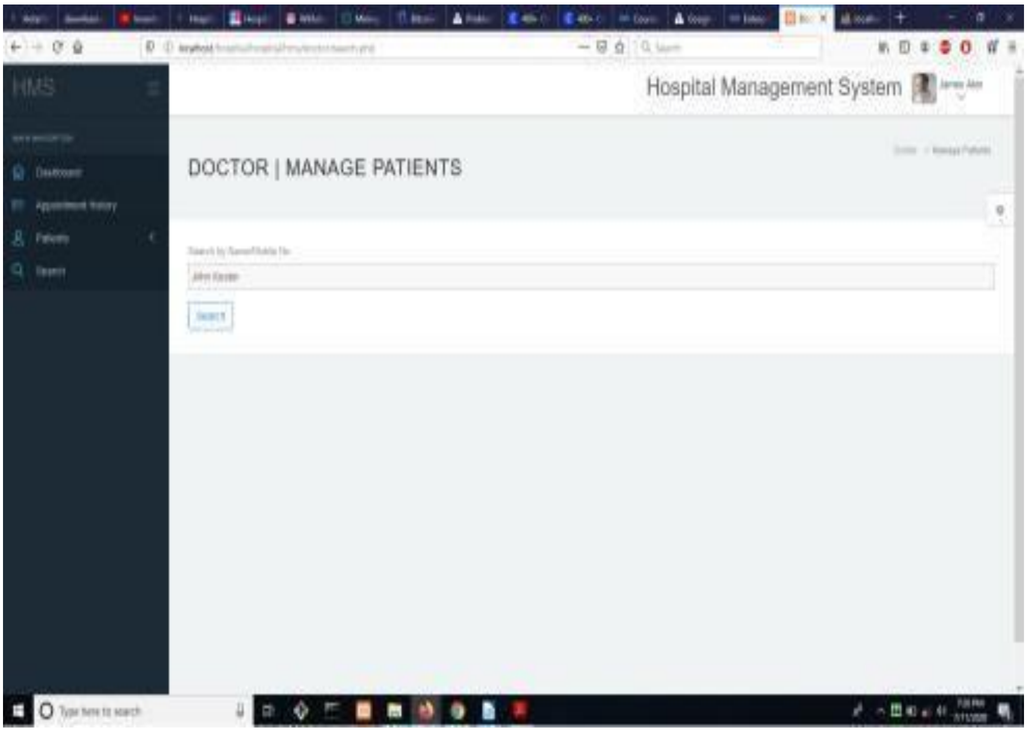
© 2020 HMS. All rights reserved

Fig. 6.13 shows an expanded view of a patient's data

The above image shows the information of a patient. Important information is the medical history which serves as a reminder for the doctor when dealing with each patient in their future meetings.

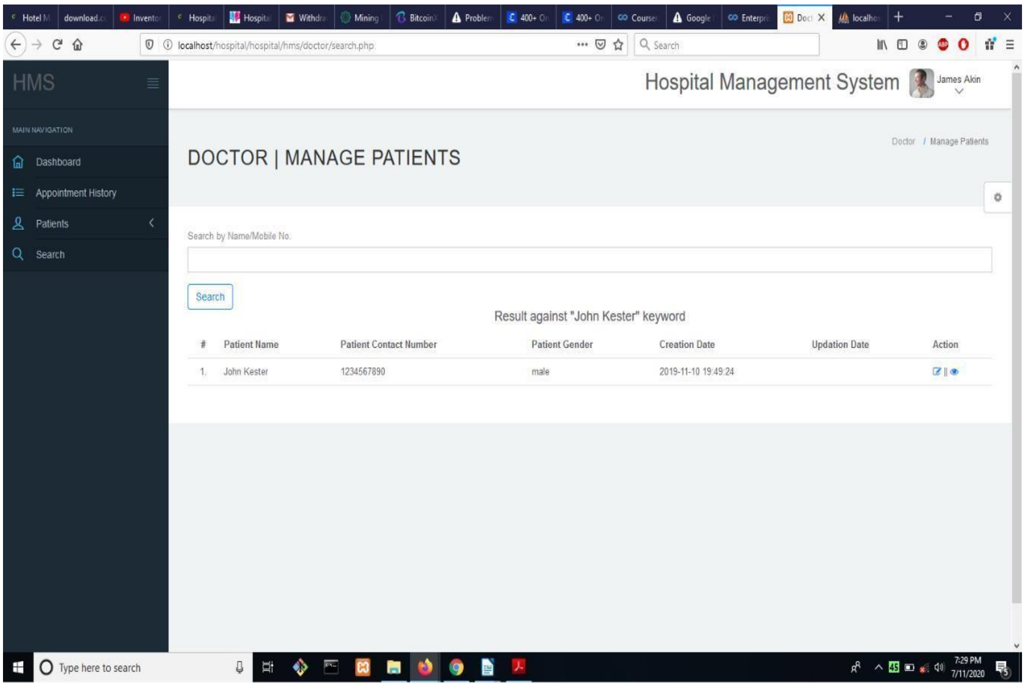
The last button on the left pane is the Search button. When the number of patients being managed by the doctor or admin becomes too large, it will be tedious and time consuming to get a patient's information by browsing or glancing through the list.

The best method of dealing with such kind of dilemma is to use an in-built search engine. This tool makes it very convenient for the doctor to quickly check the history of a particular patient in case of an emergency.



Search by Name/Mobile No.
John Kester

Fig. 6.14 above displays a handy tool for the Doctor to search for any patient. The Doctor uses a built-in search engine to get a patient called "John Kester".



DOCTOR | MANAGE PATIENTS

Search by Name/Mobile No.

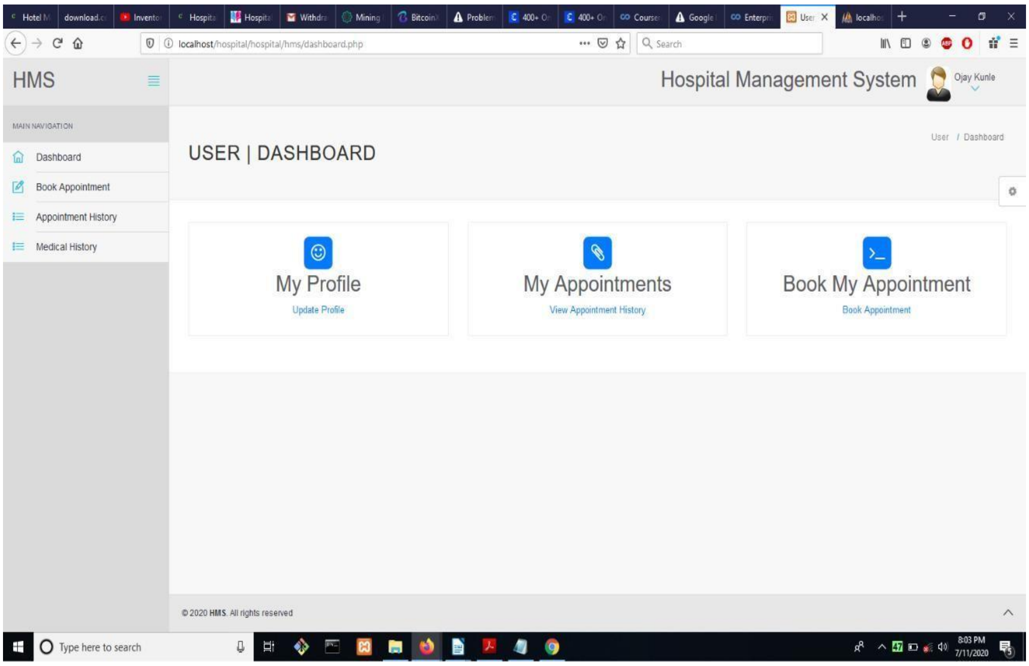
Search

Result against "John Kester" keyword

#	Patient Name	Patient Contact Number	Patient Gender	Creation Date	Updation Date	Action
1.	John Kester	1234567890	male	2019-11-10 19:49:24		✎ || 👁

Fig. 6.15: shows that the search engine has displayed the entry information concerning the patient. From here, the Doctor can edit/update or view the patient's medical history.

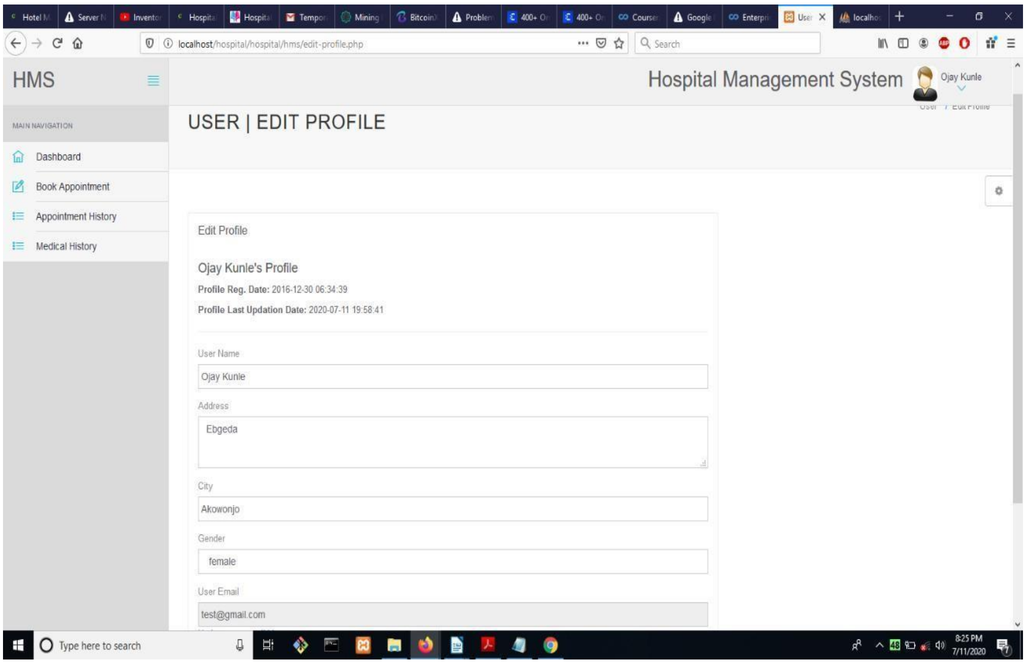
Patient's Module



USER | DASHBOARD

Fig. 6.16: Patient's dashboard

After logging in, registered patients will be ushered to their dashboard where they will be able to see/edit/update their profile, view/cancel their appointment and also book another appointment if need be. For the purpose of explaining the functionalities of this system, we will use the dashboard of a patient whose name is **Ojay Kunle**.



USER | EDIT PROFILE

Edit Profile

Ojay Kunle's Profile

Profile Reg. Date: 2016-12-30 06:34:39

Profile Last Updation Date: 2020-07-11 19:58:41

User Name: Ojay Kunle

Address: Ebgeda

City: Akowonjo

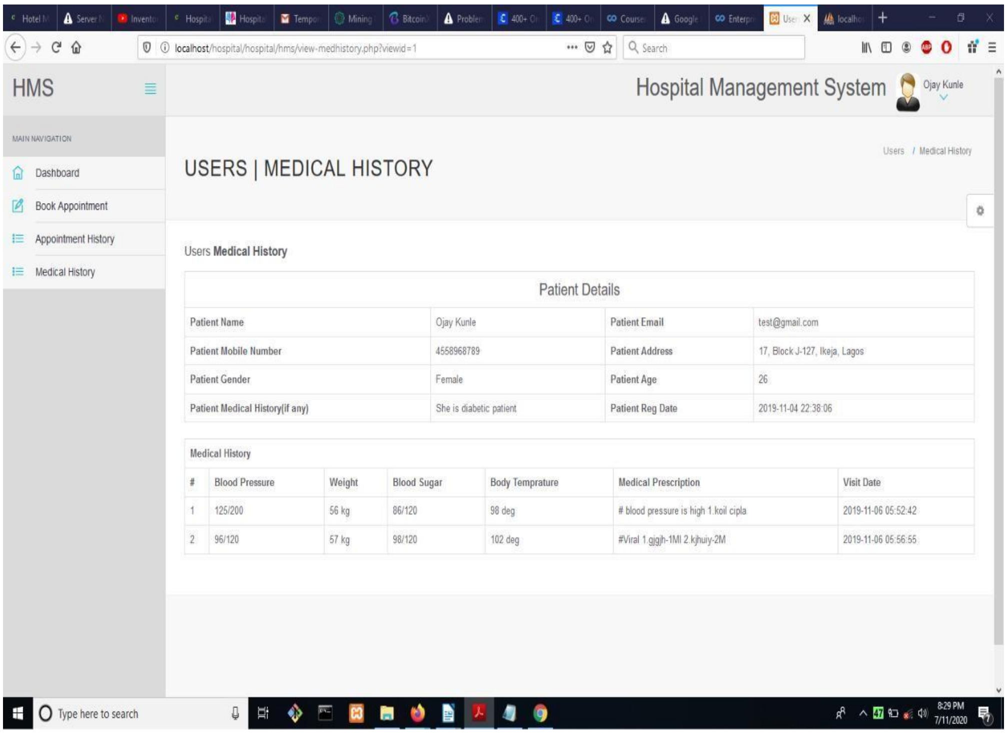
Gender: female

User Email: test@gmail.com

Fig. 6.17: Patient's profile

Patient Profile: The patient's profile contains name, gender, address, username and email. Every single important detail that is necessary will be there for them. This is just the profile page for a patient. They are to check and update their information in this page.

Medical History: In the medical history section of the patient dashboard, they will be able to check their previous prescriptions if they have one. This will come handy for a lot of people. It is very common for people to lose hand-written prescriptions all the time. Through this system, patients can check their prescriptions from a list where prescriptions will appear by date when they were assigned to them. They will be able to check old prescriptions or new prescriptions whenever they wish without any hassle. This is a novel functionality that brings you the convenience to stay close to information that is needed and useful.



USERS | MEDICAL HISTORY

Users **Medical History**

Patient Details

Patient Name	Ojay Kunle	**Patient Email**	test@gmail.com
Patient Mobile Number	4558968789	**Patient Address**	17, Block J-127, Ikeja, Lagos
Patient Gender	Female	**Patient Age**	26
Patient Medical History(if any)	She is diabetic patient	**Patient Reg Date**	2019-11-04 22:38:06

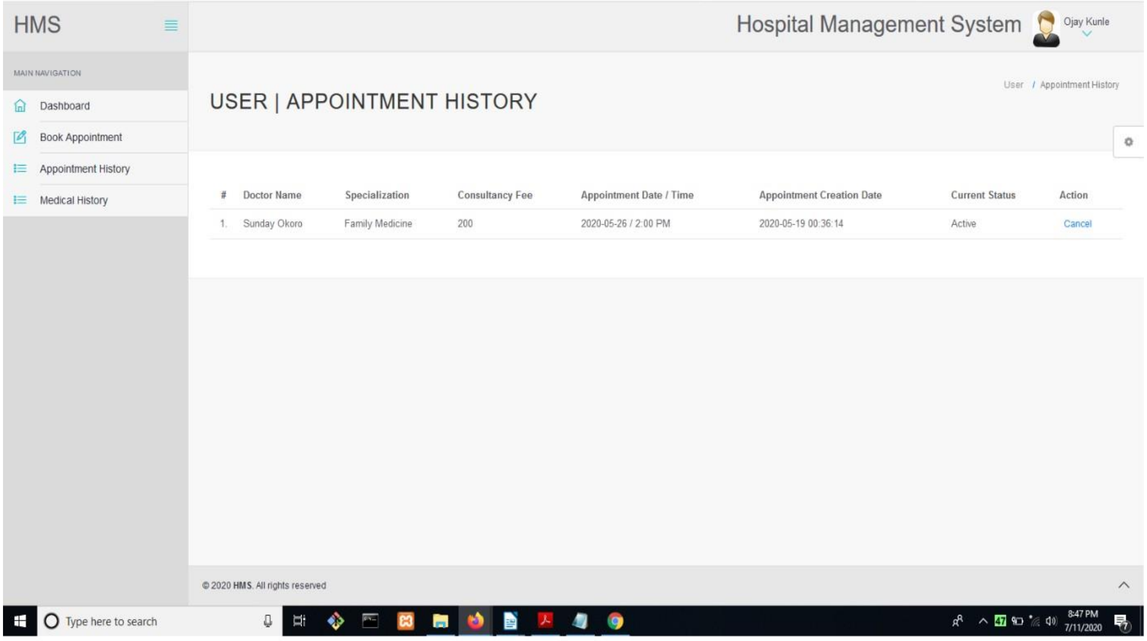
Medical History

#	Blood Pressure	Weight	Blood Sugar	Body Temprature	Medical Prescription	Visit Date
1	125/200	56 kg	86/120	98 deg	# blood pressure is high 1.koil cipla	2019-11-06 05:52:42
2	96/120	57 kg	98/120	102 deg	#Viral 1.gjgh-1Ml 2.kjhuiy-2M	2019-11-06 05:56:55

Fig. 6.18: Medical history and records as viewed by the patient

Appointment History: The patient is able to view his/her current appointments with a Doctor in this section. Relevant information such as the Doctor's name, the specialization, appointment date and time, consultancy fee, status (active or canceled) are displayed here.

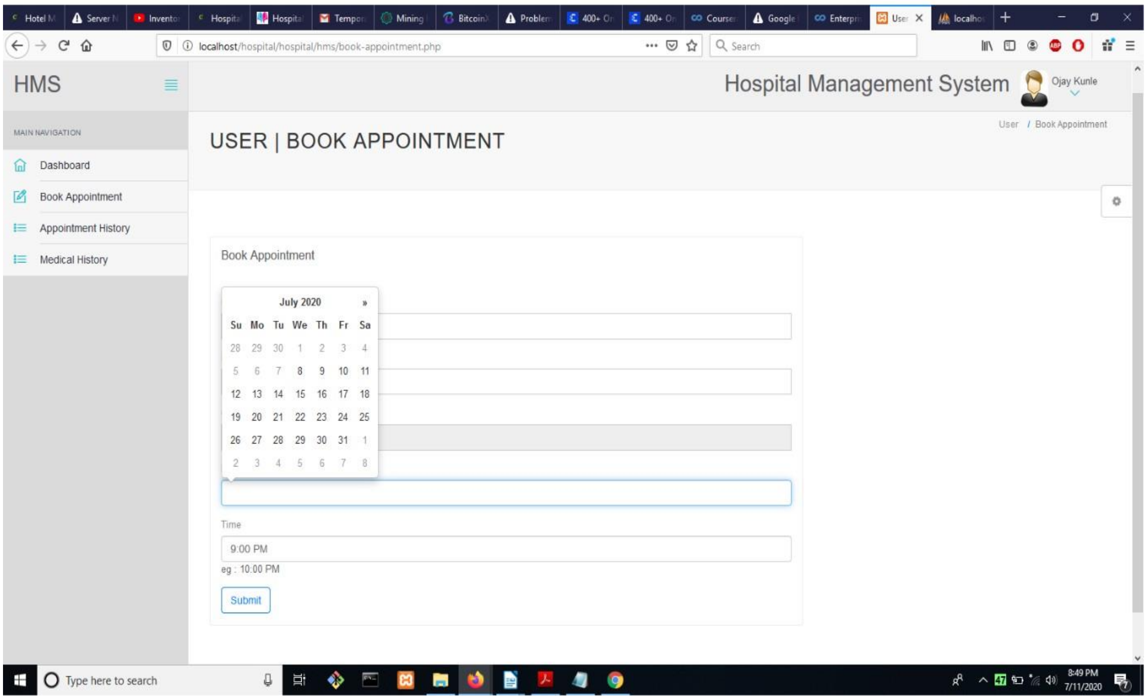
Book Appointment: This functionality gives the patient the ability to book an appointment with a Doctor. In this section a patient will be able to browse throw all the doctors available and their specialization and book for an appointment with their desired doctor. Another important feature is the luxury of doing this while sitting at home. This is quite an advanced tool if we juxtapose this functionality with the system currently being used.



USER | APPOINTMENT HISTORY

#	Doctor Name	Specialization	Consultancy Fee	Appointment Date / Time	Appointment Creation Date	Current Status	Action
1.	Sunday Okoro	Family Medicine	200	2020-05-26 / 2:00 PM	2020-05-19 00:36:14	Active	Cancel

Fig. 6.19 shows how a patient views his appointment.



USER | BOOK APPOINTMENT

Book Appointment

July 2020 »

Su	Mo	Tu	We	Th	Fr	Sa
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

Time

9:00 PM

eg : 10:00 PM

Submit

Fig. 6.20 shows a patient booking an appointment with a calendar and time functions.

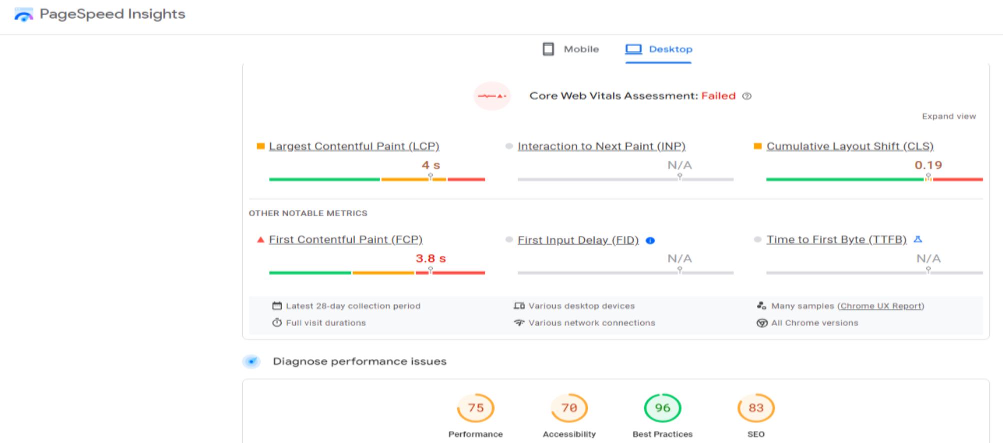
6.1 Measurement Results

The whole project turned out to meet the intended goals. The management, admin, doctor and staffs of the hospital were satisfied. The users gave a favorable response as regards the speed with which they access the system in comparison with the old system.

The site was tested with different browsers and platforms yet the pages displayed correctly without any loss of performance and graphics.

We decided to use **Google's PageSpeed Insights** website analyzer to check and compare the performance of the old and new HMS when accessed using both desktop and mobile platforms. We used PageSpeed due to its simplicity and clear analysis. The site is quite famous for its notable rigour in holding websites up to a high standard in terms of over-all performance and it is generally considered the best web tool for website performance check [24]. The test was carried out by inputting the URL of the old and new HMS into the search link of PageSpeed Insights.

Google's PageSpeed Insights website analyzer accesses web pages under four (4) categories: Performance, Accessibility, Best Practices and SEO. Performance analysis measures how well a website performs under different network conditions; Accessibility test measures how easily and efficiently web pages are navigated; Best Practices test measures the interoperability of the web technology used in designing the webpage; SEO measures the position of a website in search engine result page. Each category is calculated against a metric score of 100% with 90% being the passing score. I started with the old system:



PageSpeed Insights

Mobile | Desktop

Core Web Vitals Assessment: Failed

Expand view

Largest Contentful Paint (LCP) 4 s

Interaction to Next Paint (INP) N/A

Cumulative Layout Shift (CLS) 0.19

OTHER NOTABLE METRICS

First Contentful Paint (FCP) 3.8 s

First Input Delay (FID) N/A

Time to First Byte (TTFB) N/A

Latest 28-day collection period
Full visit durations

Various desktop devices
Various network connections

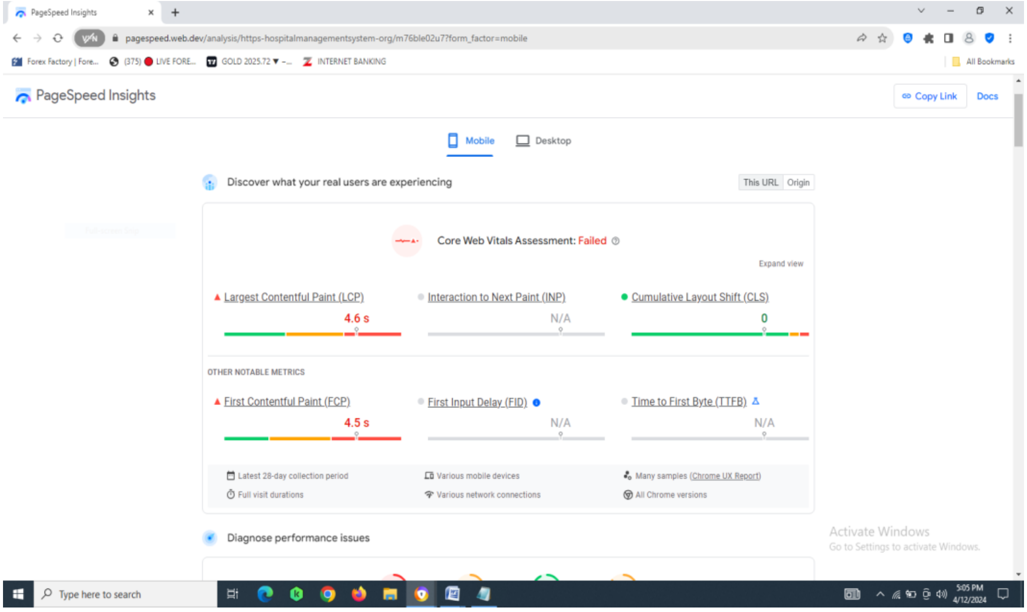
Many samples (Chrome UX Report)
All Chrome versions

Diagnose performance issues

Performance	Accessibility	Best Practices	SEO
75	70	96	83

Fig. 6.21: A web assessment of the old system indicated a **"failed"** result when accessed using a Desktop.

Secondly, we carried out a performance analysis of the old system when accessed on a mobile platform and below is the image of the result:



Discover what your real users are experiencing

Core Web Vitals Assessment: Failed

Largest Contentful Paint (LCP) 4.6 s

Interaction to Next Paint (INP) N/A

Cumulative Layout Shift (CLS) 0

OTHER NOTABLE METRICS

First Contentful Paint (FCP) 4.5 s

First Input Delay (FID) N/A

Time to First Byte (TTFB) N/A

Latest 28-day collection period

Full visit durations

Various mobile devices

Various network connections

Many samples (Chrome UX Report)

All Chrome versions

Diagnose performance issues

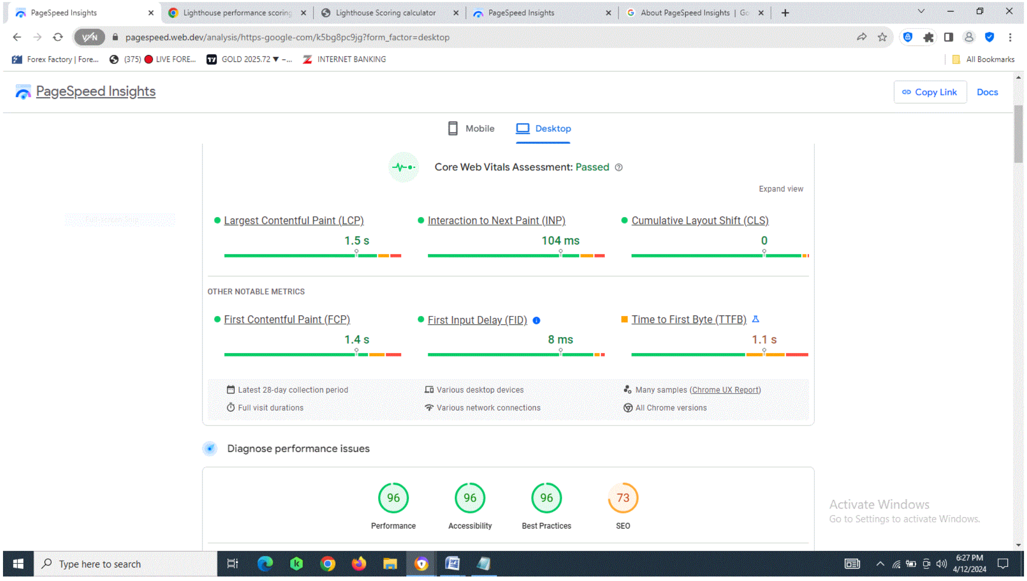
Fig. 6.22: A web assessment analysis of the old system delivers a **"failed"** result when accessed using a Mobile platform.

From the screenshot above it is plain that the old system currently in use failed the fundamental web assessment tests as indicated from the results of the performance analysis. On both desktop and mobile platforms the outcome was deemed insufficient due to the inability of the system to meet the base target point.

On Desktop platform, the over-all performance score was 75% as against the base metric score of 90%; while the Accessibility score was 70% against the base score of 95%; it got a good score of 96% in Best Practice against the baseline score of 90%. And finally an 83% score in SEO. The latter score is generally not significant for the purpose of this system.

Using a Mobile platform, the over-all performance score was 52% as against the base score of 80%; while the Accessibility score was 73% against the base score of 80%; it got a good score of 93% in Best Practice against the baseline score of 90%. And an 86% score in SEO. The latter score is generally not significant or needed for the purpose of this system.

The performance analysis for our system shows good results in all the important and necessary metrics. Below are the results:



PageSpeed Insights

Core Web Vitals Assessment: Passed

● Largest Contentful Paint (LCP) 1.5 s
● Interaction to Next Paint (INP) 104 ms
● Cumulative Layout Shift (CLS) 0

OTHER NOTABLE METRICS

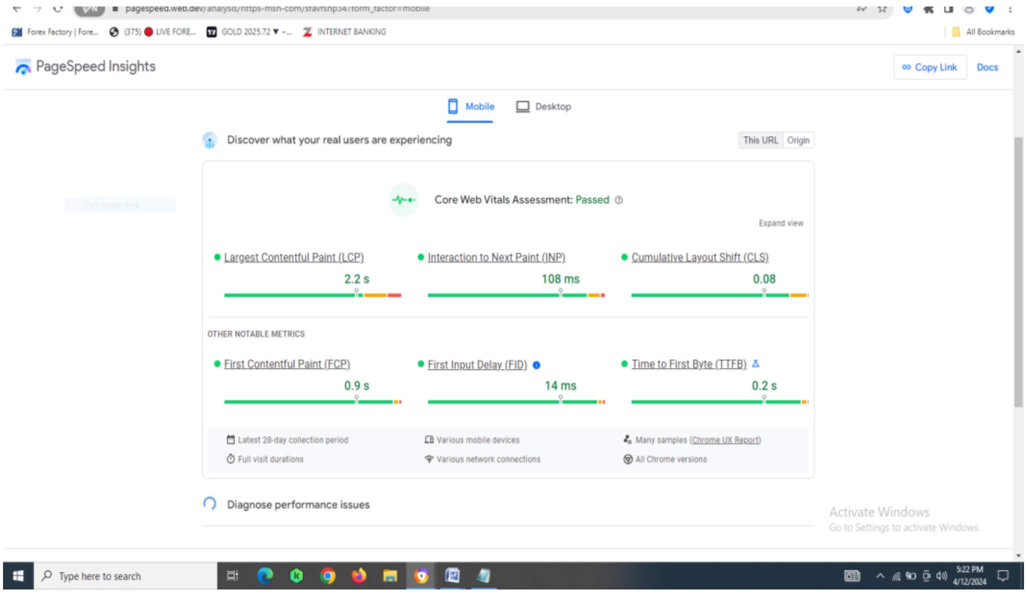
● First Contentful Paint (FCP) 1.4 s
● First Input Delay (FID) 8 ms
■ Time to First Byte (TTFB) 1.1 s

Latest 28-day collection period
Full visit durations
Various desktop devices
Various network connections
Many samples (Chrome UX Report)
All Chrome versions

Diagnose performance issues

96	96	96	73
Performance	Accessibility	Best Practices	SEO

Fig. 6.23: A web assessment analysis of our system delivers a **“passed”** result when accessed on a Desktop platform.



PageSpeed Insights

Discover what your real users are experiencing

Core Web Vitals Assessment: Passed

● Largest Contentful Paint (LCP) 2.2 s
● Interaction to Next Paint (INP) 108 ms
● Cumulative Layout Shift (CLS) 0.08

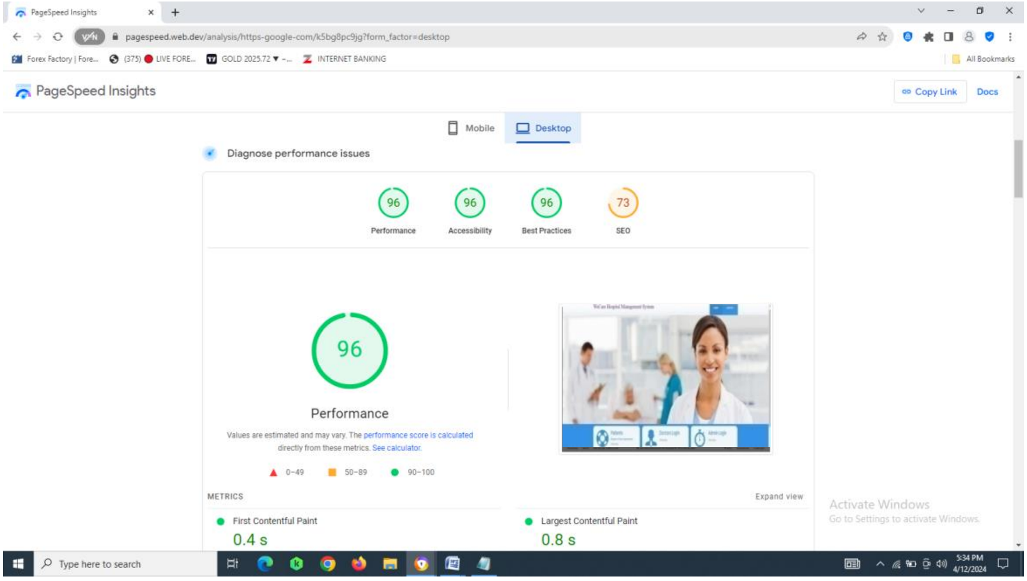
OTHER NOTABLE METRICS

● First Contentful Paint (FCP) 0.9 s
● First Input Delay (FID) 14 ms
● Time to First Byte (TTFB) 0.2 s

Latest 28-day collection period
Full visit durations
Various mobile devices
Various network connections
Many samples (Chrome UX Report)
All Chrome versions

Diagnose performance issues

Fig. 6.24: A web assessment analysis of our system delivers a **“passed”** result when accessed on a Mobile platform.



Diagnose performance issues

Performance

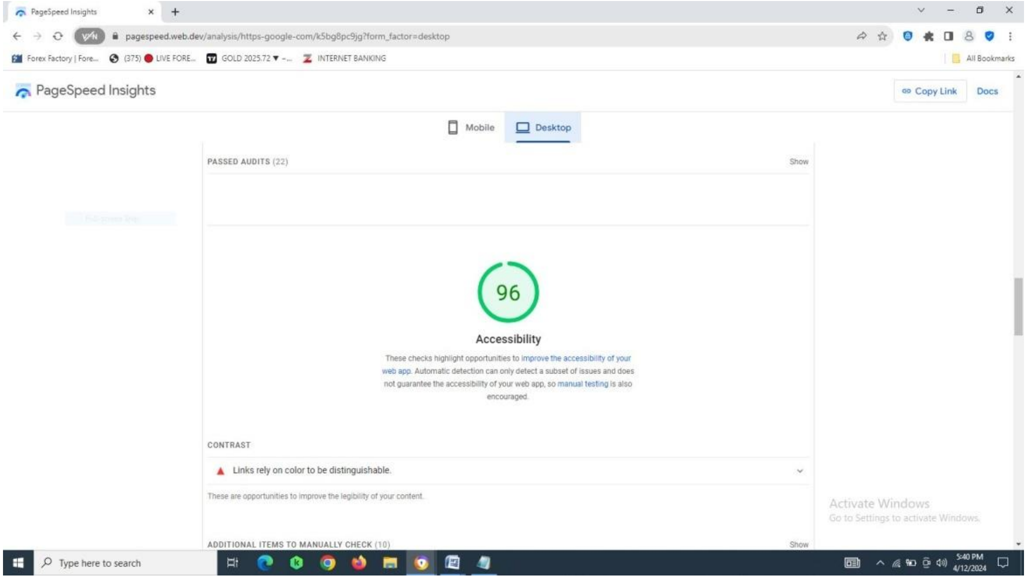
Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

METRICS

First Contentful Paint 0.4 s

Largest Contentful Paint 0.8 s

Fig. 6.25: A performance analysis of our system delivers a 96% result.


PageSpeed Insights
pagespeed.web.dev/analysis/https-google-com/k5bg8pc9jg?form_factor=desktop
Copy Link
Docs
Mobile
Desktop
Show
96
Show
Activate Windows
Go to Settings to activate Windows.
Type here to search
5:40 PM
4/12/2024

PASSED AUDITS (22)

Accessibility

These checks highlight opportunities to improve the accessibility of your web app. Automatic detection can only detect a subset of issues and does not guarantee the accessibility of your web app, so manual testing is also encouraged.

CONTRAST

Links rely on color to be distinguishable.

These are opportunities to improve the legibility of your content.

ADDITIONAL ITEMS TO MANUALLY CHECK (10)

Fig. 6.26: An accessibility analysis of our system delivers a 96% result.

Other assessments such as web "Best Practices" also delivered a 96% results. Our system failed in the "SEO" category, getting a score of 73% against a passing score of 95%. But this is not important or crucial in this project. SEO is not part of the original requirements so the system is not optimized for it.

In addition to the survey conducted and explained in section 3.6, we needed to know if our concrete goals of designing an effective, efficient and patient-centric system have been achieved.

We in conjunction with the hospital admin introduced a review by means of questionnaire that staffs and patients answered. In the summary of the survey, a total number of Sixty-five (combining Doctors and Patients) people participated with about a total of fifty-six people (56) answering questions online while nine (9) others participated via Zoom video call.

The results from the respondents established the fact that the new proposed system will have a positively tremendous impact on the way patients relate with their doctors and vice versa.

Below is the order of the list of data and the summary of result obtained from the survey. We used the following formula to calculate the average score;

Average score = mean – opinion score

0	Not Interested	=	P0
1	Least Interested	=	P1
2	Interested	=	P2
3	Highly Interested	=	P3

Average score = 0P1 + 1P1 + 2P2 + 3P3

The average score in this case, is the total number given from the list divided by the size of the list, while the opinion score is the result we got from respondents who answered all the questions from the survey. The order of the grading was from Not Interested, Least Interested, Interested and Highly Interested.

Usability Satisfaction: We enquired from the three categories of users about their impression of the new system as regards the efficiency, robustness, ease of use and speed of interaction with the system, which were the original goals we set out to achieve.

In summary, by **efficiency**, we mean how optimized and accessible it is using varied platforms and screen size (Laptop, Desktop, Mobile); by **ease of use**, we mean a system that is easily operable with minimal effort or training without wasting unnecessary time. While **Robustness** means a system that functions properly without producing erroneous of false information, and **speed** means that the system can be accessed and still load quickly even under a not-so-good network environment.

We garnered the test results from the respondents. From which we did a rating in the range 1 – 6 (Where 1 = very poor, 2 = poor, 3 = average, 4 = good, 5 = very good and 6 = excellent). Below is a table that captures the usability test results as gathered;

RESPONDENTS	**Efficiency**	**Speed**	**Robustness**	**Ease of Use**
Admin	4	5	5	6
Staffs	5	6	5	5
Users	6	5	5	6

Table 6.1 displays the average rating from the testers of the system

From the table above, we computed that the average rating from the admin is 5. The average from the staffs is 5.25; while from the public users, the rating is 5.5. The total average rating from the three groups gives us 5.1. This directly translates to the fact that the majority of those that participated in the test believes the new system is very good.

7 Discussion

Hospital management software is software meant to computerize the day to day average small hospital management activities and capable of providing easy and very effective storage information including admin management, patient registration, patient medical records, and doctors' information. Understanding the complexity of software development process and life circle was quite challenging and demanding, therefore a lot of man power, coding and research were done for this project to be completed. The main scope of this project is to develop complete package Hospital management software for Valmarn Hospital, which was one of the reason development and designed took so long but what is shown in this thesis is limited to few modules because it is contract between me and the company.

At the beginning of this project, everything look really simple and easy but as time went by, it was realized that there is more to the development of the software especially the coding part, though at the end of it all it was worth it because the result is satisfying as the main objective was realized. There are still limitations to the development of this software though, due to the fact that at the moment, the software is stand-alone and cannot communicate with other another computer from another branch of the hospital. Improvement is still needed to further enhance it so that in future it can communicate with other end systems which have same software installed on them. Furthermore, it must be stated at this point that the project is still on-going since some bugs are being fixed as testing goes on. Obviously, judging from the test-run of the system and the feedbacks received from all stakeholders, we are confident we have been able to meet and achieve the initial goals we set out to accomplish, which was to build a modern, fluid, efficient and effective Hospital Management System (HMS) that is patient-centric and meets the expectations and requirements of all three categories of users.

7.1 Ethical and societal discussion

Considering the utmost importance of the healthcare industry in every nation and the critical service they render, the matter of adequate access to healthcare services is a major policy concern. All over the world, especially in developing countries like Nigeria, there is a growing public concern as regards the poor ethical standard and conduct of the staffs and professionals in the industry. Thus, the major issues dealing with ethics in the industry can be divided into two parts; Firstly, the staffs or doctors must engage the patients in every level of decision making. That is, there must be an informed consent or shared decision making where the doctors are bounded by the oath of their profession to be truthful and accurate in the information disseminated to their patients and the latter giving the consent before receiving treatment. [25]

Secondly, there is the issue of security and confidentiality of the patients' medical history, report or information. Under no circumstances are they to be revealed to a third party or the public as doing so will constitute an attack on the patient's dignity. If the aforementioned rule is breached, the patient reserve to right to recourse to the law. [29]

7.2 Future Work

Some updates we are working on include; test reports, medication prescription details which include diet advice. The billing facility of either inpatient or outpatient will also be an attribute of the future software and most importantly, a backup facility in the software system that will save important information in case of unexpected crash.

Another aspect of important update for is enhancing the security of the system as discussed before. At the moment we encouraged all categories of users to use strong password with 12-20 word length composed of alpha-numeric characters in order to make it harder for would-be hackers to guess.

It is clear that this program does not yet offer 100% functionality. But considering the current landscape of archaic HMS systems available in our hospitals, our system can be said categorically to have mind blowing features. We are still working on making this more advanced depending on feedbacks from the management and what their requirements might be in the future.

All the functionalities obtainable in our software were what we set out to achieve from the onset due to the budget constraint of Valmarn Hospital. As they grow and become larger, better system with greater features will be required; there will new negotiations on the way forward depending on the agreement reached with the management.

References

[1] Shanika Wickramasinghe, "*Choosing the Best Web Application Architecture*", October 6, 2020. [Online], Available: https://crowdbotics.com/posts/blog/how-to-choose-the-best-architecture-for-your-web-application/ [Accessed March 12, 2021].

[2] Chandrapal Singh, "*Fundamentals Of Web Application Architecture*", July 14, 2020. [Online], Available: https://www.appventurez.com/blog/web-application-architecture [Accessed March 12, 2021].

[3] Chandrapal Singh, "*Fundamentals Of Web Application Architecture*", July 14, 2020. [Online], Available: https://www.appventurez.com/blog/web-application-architecture [Accessed March 12, 2021].

[4] Omer Imran Malik, "*What is Sensitive Data Exposure Vulnerability & How to Avoid It?*" August 12, 2019. [Online], Available: https://securiti.ai/blog/sensitive-data-exposure/ [Accessed March 14, 2021].

[5] Eugene Makieiev, "*How to Create a Modern Web Application Architecture?*" January 7, 2021. [Online], Available: https://integrio.net/blog/modern-web-application-architecture [Accessed March 14, 2021].

[6] Adam Hyde, "*The Case for HTML Word Processors*" December 11, 2016. [Online], Available: https://www.adamhyde.net/html-word-processors/ [Accessed March 15, 2021].

[7] G. Domantas, "*What Is CSS and How Does It Work?*" May 2, 2018. [Online], Available: https://www.hostinger.com/tutorials/what-is-css/ [Accessed March 15, 2021].

[8] Rock Content Writer, "*What is a Scripting Language and What are the Most Common Ones?*" December 15, 2020. [Online], Available: https://rockcontent.com/blog/scripting-languages/ [Accessed March 15, 2021].

[9] Kolade Chris, "*What is PHP? The PHP Programming Language Meaning Explained*" August 30, 2019. [Online], Available: https://www.freecodecamp.org/news/what-is-php-the-php-programming-language-meaning-explained/ [Accessed March 17, 2021].

[10] Serverhub, "*MySQL: the most popular open-source database*" March 13, 2021. [Online], Available: https://serverhub.com/kb/mysql-the-most-popular-open-source-database-system/ [Accessed March 17, 2021].

[11] O. O. Lawal, B. O. Afeni, and J. O. Mebawondu, "*Development of Hospital Information Management Systems,*" B.Sc. thesis, Dept. Comp. Sci., Joseph Ayo Babalola Univ., Ado-Ekiti, Nigeria, 2016.

[12] Ahmed Shamel Noori and Ali Fadhil Najem, "*Hospital Magenememt System: Design and Implementation,*" B.Eng. thesis, Dept. Comp. Eng., University of Technology, Johor Bahru, Malaysia, 2021.

[13] Saad Alami, "*Hospital Management Control System,*" B.Sc. thesis, Dept. Comp. Sci., Mohammed V University of Rabat, Morocco, 2018.

[14] M. Sowmya, D. Anilchhandra Varma, M. Venugopalarao, M. Sailaja and T. Prasanth, "*The Hospital Magenememt System,*" B.Eng. thesis, Dept. Software Eng., Sri Lanka Institute of Information Technology, SRI LANKA, 2021.

[15] W. Jason Gilmore, *Beginning PHP and MySQL: From Novice to Professional.* Ohio, USA: Apress, 2010.

[16] Janet Valade and Steve Suehring, *PHP, MySQL, JavaScript, HTML5: All in one for Dummies.* California, USA: Wiley, 2014.

[17] Prof. Dharminder Kumar, "*Overview of System Analysis & Design*" Lesson no. 1, pp. 23-37, 2013.

[18] B. Rajanisade, "*Hospital Management System,*" February 3, 2016. [Online], Available: https://www.coursehero.com/file/p2r99h3/1-

What-are-the-users-demonstrable-needs-and-how-does-a-candidate-system-meet/ [Accessed March 19, 2021].

[19] Prof. Dharminder Kumar, "*Overview of System Analysis & Design*" Lesson no. 1, pp. 42-45, 2013.

[20] Prof. Dharminder Kumar, "*Overview of System Analysis & Design*" Lesson no. 1, pp. 51-56, 2013.

[21] Dr. Mohammed Najm Abdulla, Dr. Intisar Al-Mejibli, Shaimaa Khamees Ahmed, "*An Investigation Study of Hospital Management Information System,"* Comp. Eng. Dept., University of Technology, Iraq, IJARCCE, 2017.

[22] Michal Malewicz, "*The User Interface (UI) Design*" June 5, 2020. [Online], Available: https://www.interaction-design.org/literature/topics/ui-design [Accessed March 19, 2021].

[23] Felipe Guimaraes e Equipe Aela, "*Nielsen's Heuristics: 10 Usability Principles To Improve UI Design*" June 29, 2019. [Online], Available: https://aelaschool.com/en/interactiondesign/10-usability-heuristics-ui-design [Accessed March 19, 2021].

[24] Google developer's guide, "*About PageSpeed Insights*" September 21, 2020. [Online], Available: https://developers.google.com/speed/docs/insights/v5/about [Accessed March 19, 2021].

[25] Michael Chika Diyoke, "*Healthcare Ethics Issues and Challenges in Nigeria*" August 16, 2019. [Online], Available: https://www.researchgate.net/publication/340536886_Healthcare_Ethics_Issues_and_Challenges_in_Nigeria [Accessed March 20, 2021].

[26] USAID, "*Health Management Information System (HMIS) Facilitator's Guide for Training of Trainers,"* February 2010, Version 1.1. [Online], Available:https://www.measureevaluation.org/resources/publications/ms-13-74/at_download/document [Accessed March 20, 2021]