

Project Title:

**Smart Healthcare System**

Submitted by:

**Iqra Muzaffar [2020-CE-45]**

**Hooria Azeem [2020-CE-46]**

**Palwasha Zulfiqar [2020-CE-48]**

Submitted to:

**Dr.Asim Rehmat**

**Ma’am Sana**

Course:

**[CMPE-311L Software Engineering]**

Semester:

**5th (Fifth)**

Date:

**02/01/2023**

**Department of Computer Engineering**

**University of Engineering and Technology, Lahore**

# **Table of contents:**

[Introduction: 3](#_Toc122493861)

[Problem Statement: 3](#_Toc122493862)

[**Objectives:** 3](#_Toc122493863)

[**Methodology:** 4](#_Toc122493865)

[Major Functionalities: 5](#_Toc122493866)

[**EMR:** 5](#_Toc122493867)

[**OCR:** 5](#_Toc122493868)

[**Tesseract:** 7](#_Toc122493869)

[Requirement Analysis: 8](#_Toc122493870)

[Feasibility: 9](#_Toc122493871)

[**Security:** 9](#_Toc122493872)

[**Performance:** 9](#_Toc122493873)

[**Technical Requirements (Tools):** 10](#_Toc122493874)

[**Product Development:** 10](#_Toc122493872)

[**Intended Audiance:** 11](#_Toc122493873)

[**Walk-Through:** 11](#_Toc122493874)

[**Visibility Plan:** 12](#_Toc122493872)

[**Resources:** 12](#_Toc122493873)

[**Usability:** 12](#_Toc122493874)

[**Risk Analysis:** 13](#_Toc122493872)

[Data Flow Models: 14](#_Toc122493875)

[**Zero Level DFD:** 14](#_Toc122493876)

[**First Level DFD:** 15](#_Toc122493877)

[**Second Level DFD:** 15](#_Toc122493878)

[Use-Case Diagrams: 16](#_Toc122493875)

[UI Designs: 19](#_Toc122493875)

[**Database (ERD-Dia):** 24](#_Toc122493878)

[Conclusion: 26](#_Toc122493881)

[*References* 26](#_Toc122493882)

# **Introduction:**

# This project is basically a Hospital Management system which includes online appointments of patients, storing their details into the system and also deals with prescription and computerized billing. This software has the facility to give a unique id for every patient and stores the details of every patient and the doctors automatically. It includes a search facility to know the current status of each appointment and prescription.

# Most importantly it is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to hospitals as it automates the manual work .It enables you to develop your organization and improve its effectiveness and quality of work.

# **Problem Statement:**

The information regarding patient's appointments and history of treatment is very difficult to retrieve and to find particular information like- E.g. To find out about the patient’s history, the user has to go through various registers. This results in inconvenience and wastage of time. Manual calculations are error prone and take a lot of time this may result in incorrect information. For example calculation of patient’s bill based on various treatments. Hospitals need efficient systems to work efficiently!

## **Objectives:**

* To design a software to manage all the areas of a hospital such as medical, financial, administrative and the corresponding processing of services.
* System lists all of the various hospital departments as well as the doctors
* A patient’s personal account can also be created through the system.
* There are three sections of this project:
* Panels for Admin
* Panel for Patients
* Panel for Doctors
* The user can simply make an appointment on this web application’s overview. In other words, each user will be required to fill out a client-side appointment form. The patient’s name, address, phone number, and department and doctor selections are all included in the form. The system also establishes personal accounts for each patient during this procedure.

### **Benefits:**

* The system will facilitate online appointments in order to replace paper appointments. This will make the process smoother and efficient.
* It will allow multiple appointments to be organized simultaneously.
* It will allow PATIENTS to choose appointment date and time of their choice.
* In addition to making and attending appointments, treatment, prescription and billing areas are made efficient.

### **Methodology:**

We are using **AGILE METHODOLOGY**. Agile project management is an iterative approach to delivering a project throughout its life cycle.

Iterative or agile life cycles are composed of **several iterations or incremental steps** towards the completion of a project. Iterative approaches are frequently used in software development projects to promote velocity and adaptability since the benefit of iteration is that you can adjust as you go along rather than following a linear path.

One of the aims of an agile or **iterative approach** is to release benefits throughout the process rather than only at the end. At the core, agile projects should exhibit central values, **flexibility**, empowerment and collaboration [1].

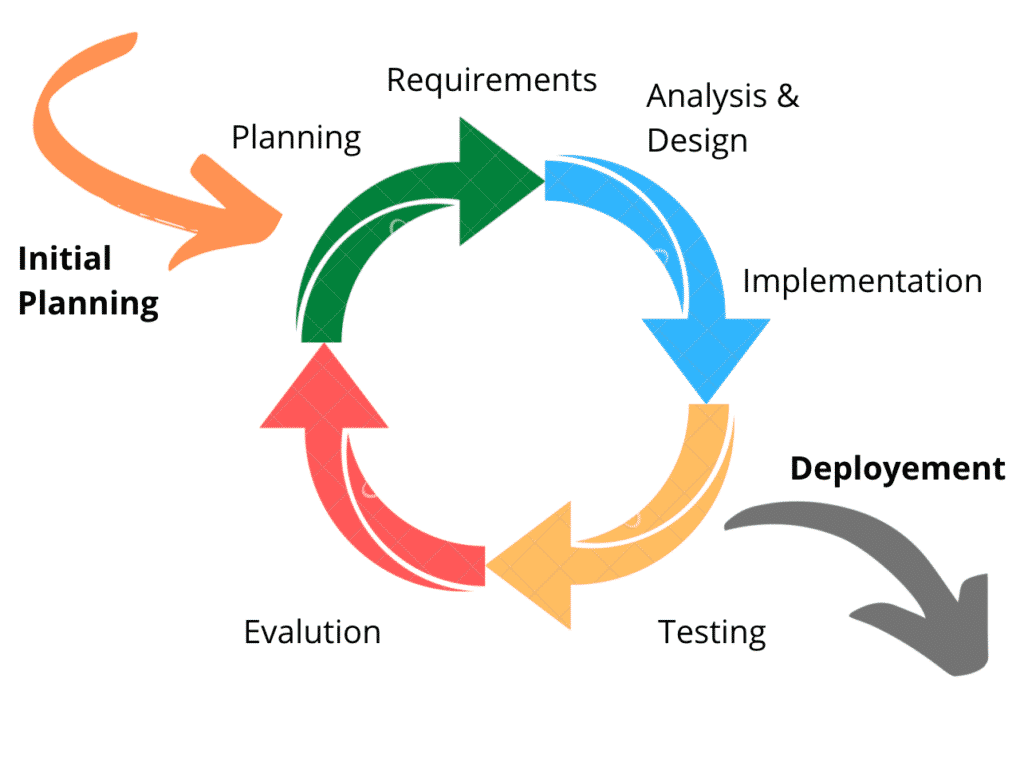


Figure 1

### **Major Functionalities:**

### **EMR:**

Electronic medical records (EMRs) are a **digital version** of the paper charts in the clinician’s office. An EMR contains the medical and treatment history of the patients in one practice. EMRs have advantages over paper records. [2] For example, EMRs allow clinicians to:

* Track data over time
* Easily identify which patients are due for preventive screenings or checkups
* Check how their patients are doing on certain parameters such as blood pressure readings or vaccinations
* Monitor and improve overall quality of care within the practice.

### **EMR TASKS IN THIS PROJECT:**

* Manage all areas of hospital.
* Creating panels for Admin, Doctors and Patients.
* Managing online appointments.
* Managing treatment, prescription and billing areas.

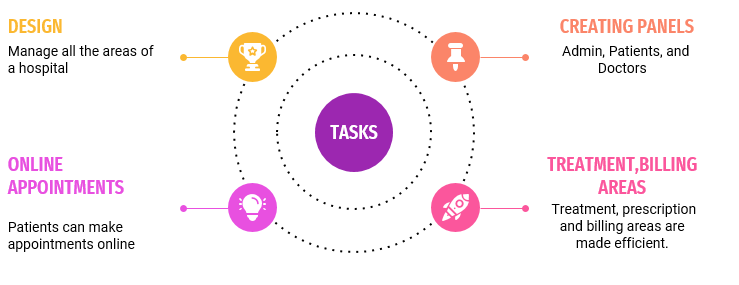


Figure 2

### **Optical Character Recognition-OCR:**

Optical Character Recognition (OCR) is the process that **converts** an image of text into a **machine-readable text format**. For example, if you scan a form or a receipt, your computer saves the scan as an image file. You cannot use a text editor to edit, search, or count the words in the image file. However, you can use OCR to convert the image into a text document with its contents stored as text data.

Moreover, **digitizing** this document content creates image files with the text hidden within it. Text in images cannot be processed by word processing software in the same way as text documents. OCR technology solves the problem by converting text images into text data that can be analyzed by other business software. You can then use the data to conduct analytics, streamline operations, **automate processes**, and improve productivity. [3]

### **How does an OCR works?**

The OCR engine or OCR software works by using the following steps:

### **Image acquisition**

A scanner reads documents and converts them to binary data. The OCR software analyzes the scanned image and classifies the **light areas** as background and the **dark areas** as text.

### **Preprocessing**

The OCR software first **cleans the image** and removes errors to prepare it for reading. These are some of its cleaning techniques:

* Tilting the scanned document slightly to fix alignment issues during the scan.
* Removing any digital image spots or smoothing the edges of text images.
* Cleaning up boxes and lines in the image.
* Script recognition for multi-language OCR technology
* Text recognition

The two main types of **OCR algorithms** or software processes that an OCR software uses for text recognition are called pattern matching and feature extraction.

### **Pattern matching:**

Pattern matching works by **isolating a character** image, called a glyph, and comparing it with a similarly stored glyph. Pattern recognition works only if the stored glyph has a similar font and scale to the input glyph. This method works well with scanned images of documents that have been typed in a known font.

### **Feature extraction:**

Feature extraction breaks down or **decomposes** the glyphs into features such as lines, closed loops, line direction, and line intersections. It then uses these features to find the best match or the nearest neighbor among its various stored glyphs.

### **Post-processing**

After analysis, the system converts the extracted text data into a computerized file. Some OCR systems can create annotated PDF files that include both the before and after versions of the scanned document.

### **Tesseract:**

### We are using Tesseract for OCR. It is an open source OCR engine that extracts printed or written text from images. It was originally developed by Hewlett-Packard, and development was later taken over by Google. This is why it is now known as “Google Tesseract OCR”.

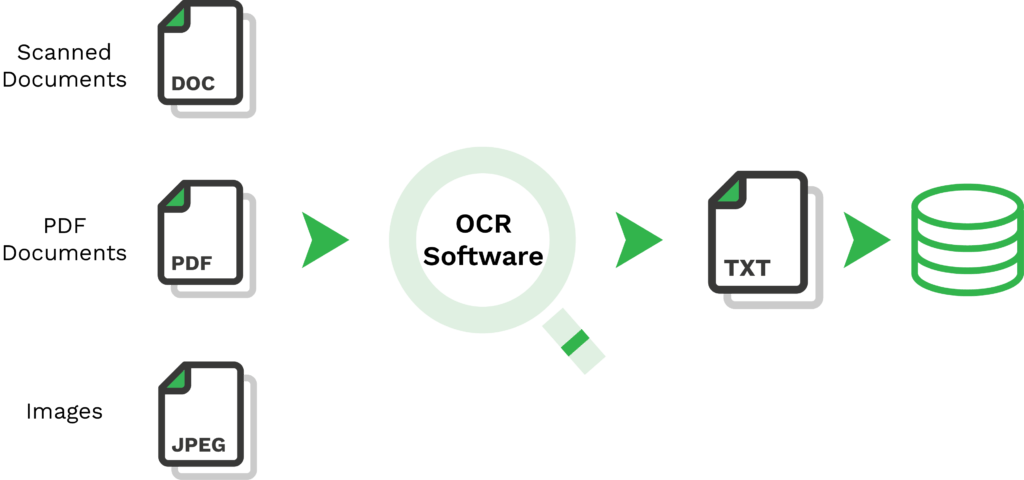


Figure 3

But what is an open-source OCR? It simply means that it is available for everyone to use freely, either directly or using an **Application Programming Interface (API)**. With Tesseract OCR, users can extract text from images with efficient in-line and character pattern recognition of the OCR engine.

As of now, Tesseract already supports language recognition for more than 100 languages “out of the box”. [4]

### 

Figure 4

### **Requirement Analysis:**

The system needs to meet the following **functional** **requirements**:

**Web Interface:**

* **Admin Panel**
* Login
* View following:
* Doctors Data
* Patients Data
* Pending Appointments
* Departments
* Treatment kinds
* Medical Records
* Patient’s Treatment
* Payment Information
* **Doctor’s Panel**
* Login
* View personal information
* Pending Appointment
* Patient’s Profile
* Appointment Record
* Treatment Record
* Prescription Area
* **Patient’s Panel**
* Login
* View personal information
* View following:
* Pending Appointment
* Treatment Record
* Prescription Record

# **Feasibility:**

The feasibility study for a proposed Management system will be wide-spread. There are several components that should be included in the study, and each will be discussed.

## **Security-Feasibility:**

* Patient Identification

The system needs the patient to recognize herself or himself using the phone

* Login ID

Any user: doctor/ patient who make use of the system need to hold a Login ID and password

* Modifications

Any modifications like insert, delete, update, etc. for the database can be synchronized quickly

* Admin Authority
* Front Desk Staff Rights to admin: The staff at the front desk (admin) can view any data in the Hospital Management system, and add new patients record to the HMS but they don’t have any rights to alter any data in it.
* The administrator can view as well as alter any information in the Hospital Management System.

## **Performance:**

* Response Time

The system provides acknowledgment in just few seconds once the ’patient’s information is checked.

* Capacity

The system needs to support at least 50 people at once.

* User-Interface

The user interface acknowledges within five seconds. It is fast, convenient and easy to use.

* Maintainability

The system offers efficiency for data backup.

* Capacity

The system needs to support at least 50 people at once.

* Errors

The system will track every mistake as well as keep a log of it.

* Reliability and Durability

The system is available all the time and it is durable in case of system errors.

## **Technical Requirements (Tools):**

Technical requirements refer to the question that how the software is built. So, basically:

* Html, CSS, Java-script for Front-end
* Xampp Server for php localhost
* MySQL database & SSMS
* PHP for backend
* Tesseract for OCR

## **Program Development:**

It basically refers to the goals and milestones of the project. So:

* Milestone 1 (Proposal (Draft))

Project Discussion-Problem analysis with client.

* Milestone 2 (Requirement Analysis)

Requirement Analysis-Requirement Discussion

* Milestone 3 (Feasibility Report (Draft)).

Feasibility study-technology, plan, budget, time division, task division

* Milestone 4 (Resources)

Learn Technologies for implementation and gather resources

* Milestone 5(Web-App)

Front-end Designs-dashboards, login, signup, contact pages

* Milestone 6(Back-end Development)

Crud operations. Appointment management, prescription, billing

* Milestone 7 (OCR)

OCR library integration

* Milestone 8 (Testing)

Integration and acceptance testing of all modules.

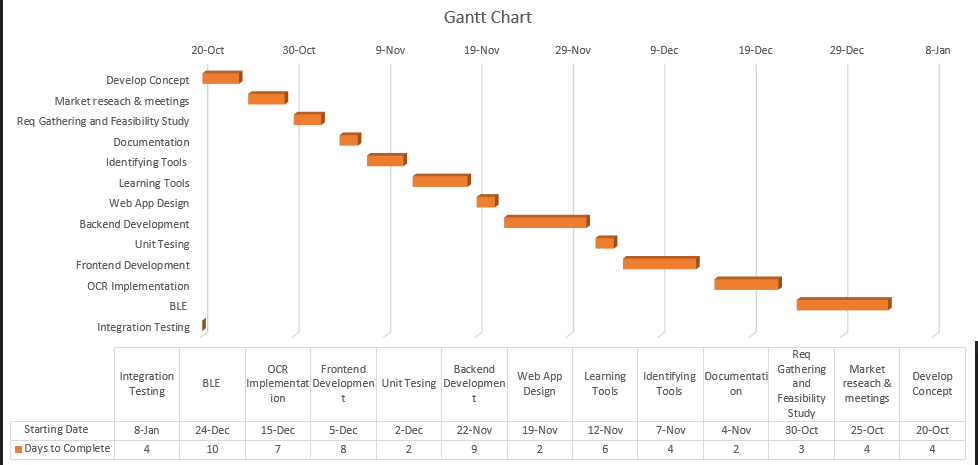


Figure 5

## **Intended Audience:**

The intended audience of this document would be the **client** and **specific employees** like Manager and Receptionist, consultants and System Operators .The SRS document can be used in any case regarding the requirements of the project and the solutions that have been taken. The document would finally provide a clear idea about the system.

**Walkthrough:**

In order to ensure that the Doctors and the Patients are on the same page, the following walkthrough has been prepared to illustrate the Group’s understanding of the product desired by the Client. The walk-through is not necessarily a reflection of the exact interactions for the final product; rather it should serve as a rough overview of the functionality required by the final product as the Group currently understands it.

### **Administrative Walkthrough:**

The administrator needs a way to login, view and manage everything in the database. To search appointment or doctors or patients the administrator will enter name of respective doctors or patients on a simple text-box, and the result will be appear if present in database. Administrator will be able to view complete data of both doctors and patients. Patients, doctors, departments, treatment kinds, and medical records may all be managed by an administrator. To view appointment schedule administrator can click on appointments scheduled.

**Client Walkthrough:**

The client needs a way to login, view appointment, book, edit and delete an appointment. To view his information the patient will click on My Info. If he has booked an appointment it will be also visible on My Info page. To modify or delete appointment the patient will click on edit or delete appointment .To view appointment schedule customer can click on appointment schedule.

## **Visibility Plan:**

* External – The Group will conduct regular biweekly meetings with the Client at the Computer

Engineering department. If situations arise or if a problem needs to be addressed between the

Meetings, The Group will conduct any further necessary communication via Zoom. As agile model will be used, a feedback will be required from the Client at the end of every step to ensure that both parties are in-sync and to minimize any miscommunication in the requirements.

* Internal – The Group will meet as per need on Zoom to discuss progress and problems.

Meeting minutes will be kept track of and sent to all members of the Group for reference. Any additional communication will be done in WhatsApp group or through other collaboration tools such as document sharing.

# **Resources:**

Our group consists of 3 members, tasks will be divided as:

•Development Member:

Her task will be to connect frontend with backend & database.

• Design Member:

Her task will be to make frontend designs, database designs.

• Social Media Member:

She will do all sort of documentation.

We are given **around 12 weeks** of time to complete the system.

# **Usability:**

## Usability issues such as **speed** of operation for the user interface, collection and storage of important quantitative data, speed and efficiency of the work flow processes through automation, and **concurrency of collected data** will be important considerations.

## **Risk Analysis:**

## 1. **Changing Requirements**:

## • Risk:

## The Client may have different ideas about the system during the course of the project. Depending on the situation, the changes that the Client wishes to have implemented may require little or major changes to the architecture.

## • Solution:

## To reduce the possibility of this occurring, the Group needs to establish a clear visibility plan with the Client.

## 2. **Incomplete Requirements**:

## • Risk:

## It is possible that requirements may be implied but not discussed or misunderstood. This frequently occurs after meetings.

## • Solution: The Group’s interpretation of the Client’s requirements will be presented back to the Client to get a confirmation on whether the Group has understood the Client. Frequent client updates and a high level of visibility will also help call attention to any misunderstandings.

## 3. **Human Resources**:

## • Risk:

## The Group is relatively small consisting of only 3 members, some members are not technically oriented and almost all members have limited knowledge of relevant web technologies.

## • Solution:

## For these reasons the Group acknowledges that a slow design and implementation phase may be inevitable, and are planning accordingly.

## 4. **Lack of Resources, Tools**:

## • Currently we will be unable to provide online payment due to lack of access and cooperation with banks.

## • Our website will only be dealing with events within Lahore.

## **Data Flow Models:**

In Software engineering DFD (data flow diagram) can be drawn to represent the system of different levels of abstraction. Higher-level DFDs are partitioned into low levels-hacking more information and functional elements. Levels in DFD are numbered 0, 1, 2 or beyond. Here, we will see mainly 3 levels in the data flow diagram, which are: 0-level DFD, 1-level DFD, and 2-level DFD.

**Zero Level DFD**:

It is also known as a context diagram. It’s designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as a single bubble with input and output data indicated by incoming/outgoing arrows.

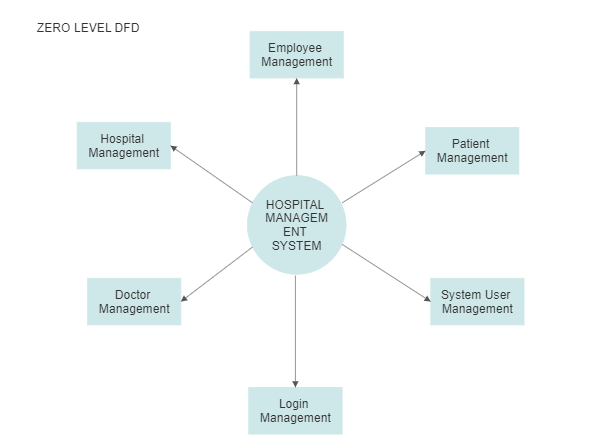


Figure 6

### **First Level DFD:**

In 1-level DFD, the context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main functions of the system and breakdown the high-level process of 0-level DFD into sub-processes.

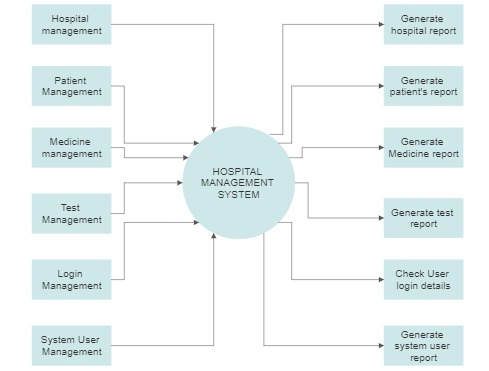


Figure 7

### **Second Level DFD:**

2-level DFD goes one step deeper into parts of 1-level DFD. It can be used to plan or record the specific/necessary detail about the system’s functioning.

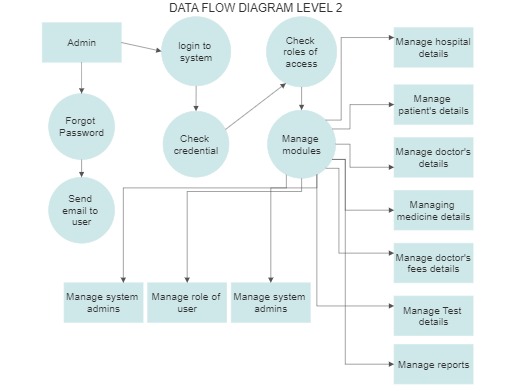


Figure 8

## **Use Case Diagrams:**

In UML, use-case diagrams model the behavior of a system and help to capture the requirements of the system. Use case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally.

Use-case diagrams illustrate and define the context and requirements of either an entire system or the important parts of the system. You can model a complex system with a single use-case diagram, or create many use-case diagrams to model the components of the system.

### **Admin:**

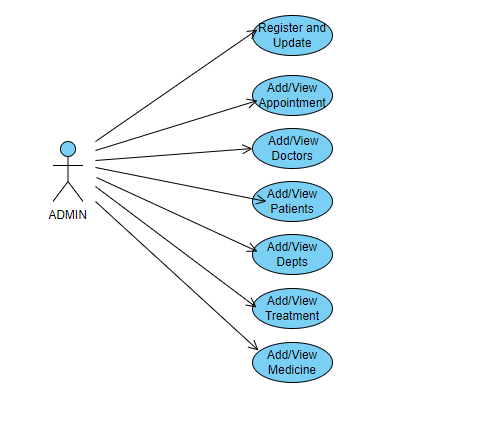


Figure 9

### **Doctor:**

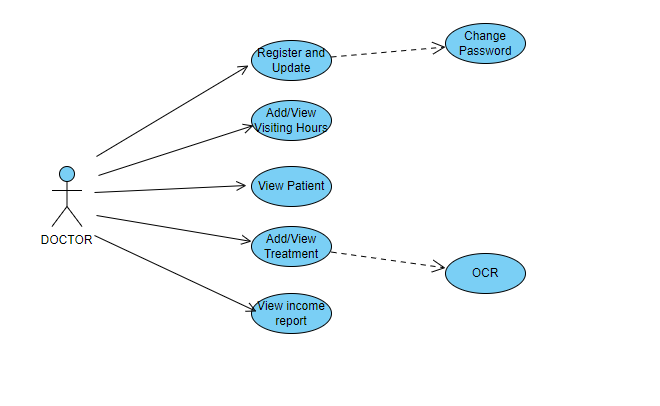


Figure 10

### **Patient:**

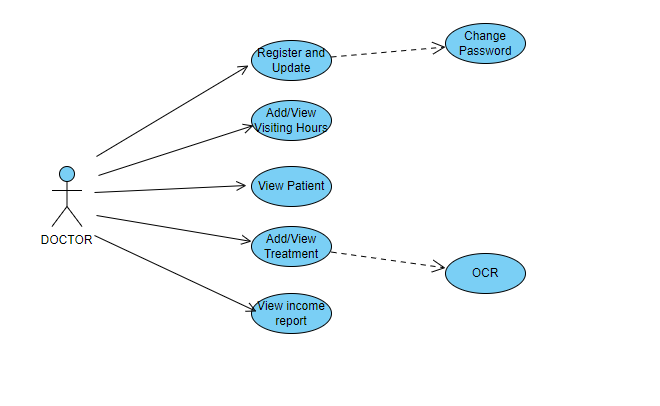


Figure 11

### **UI Designs:**

### User interface (UI) design is the process designers use to build interfaces in software or computerized devices, focusing on looks or style. Designers aim to create interfaces which users find easy to use and pleasurable. UI design refers to graphical user interfaces and other forms. [5]

### **Important Features:**

### To design UIs best, you should consider:

### Users judge designs quickly and care about **usability** and likeability.

### They don’t care about your design, but about getting their tasks done easily and with **minimum effort.**

### So, **understand** your users’ contexts and task flows (which you can find from, e.g., customer journey maps), to fine-tune the best, most intuitive UIs that deliver seamless experiences.

### UIs should also **be enjoyable** (or at least satisfying and frustration-free).

### When your design **predicts users’ needs**, they can enjoy more personalized and immersive experiences. Delight them, and they’ll keep returning.

### UIs should **communicate brand values** and reinforce **users’ trust**.

### Good design is **emotional design**. User associate good feelings with brands that speak to them at all levels and keep the magic of pleasurable, seamless experiences alive.

### **UI Design of Our Project:**

### **Homepage:**

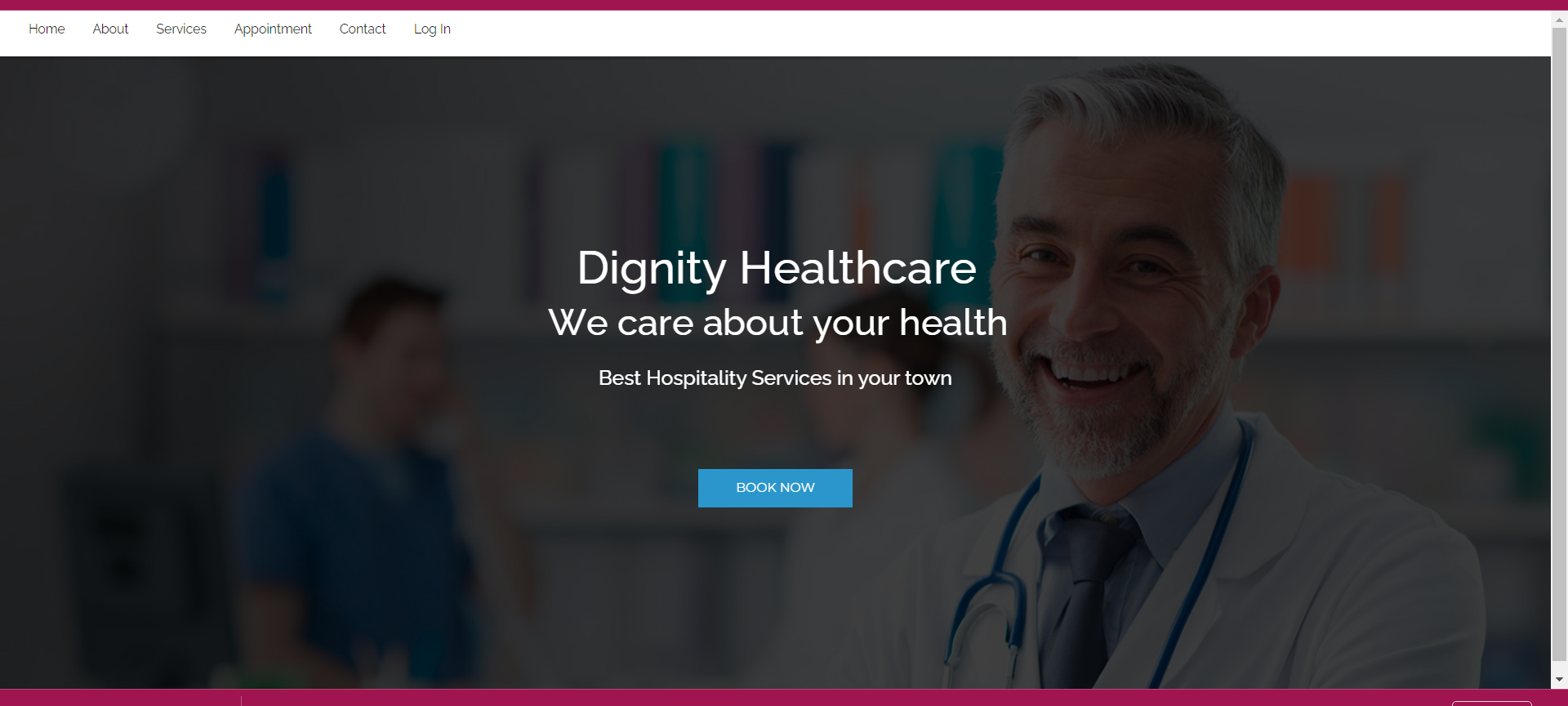


Figure 12

### **Contact Page:**

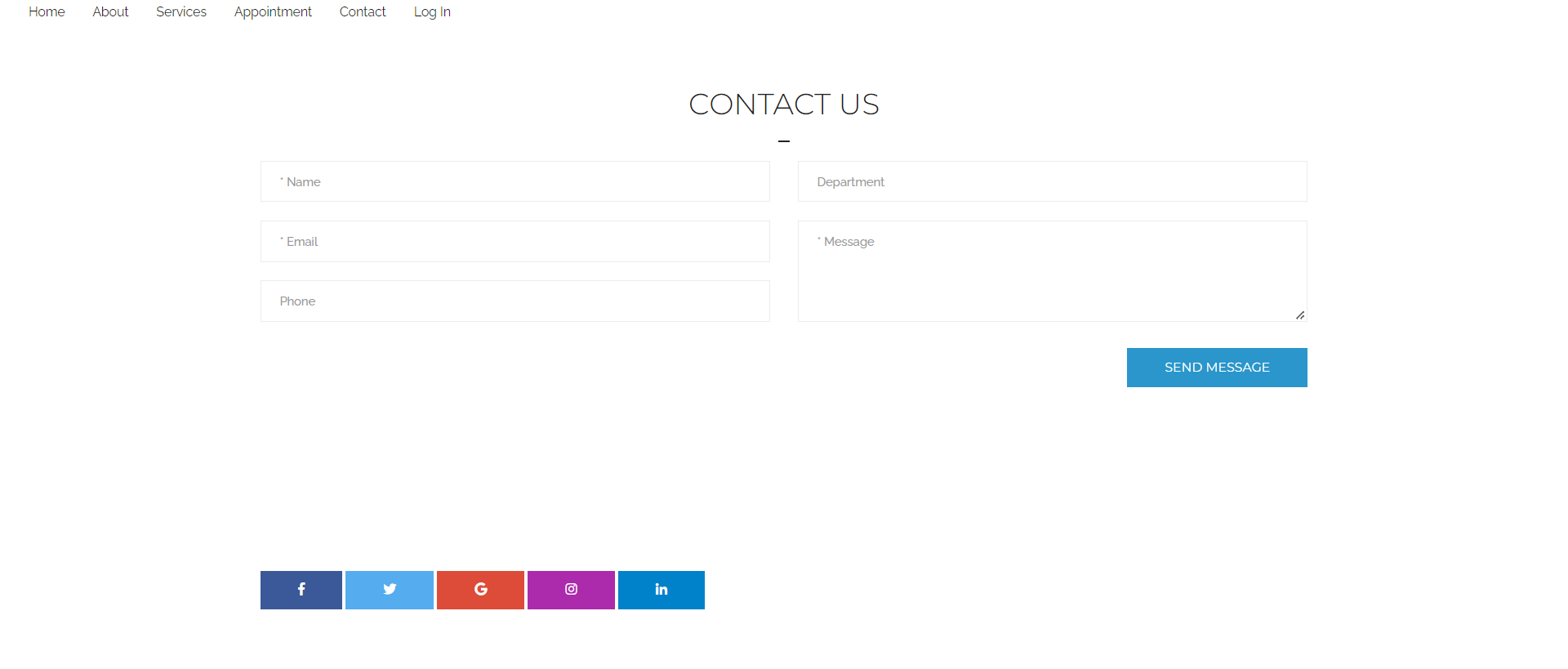


Figure 13

### **Services:**

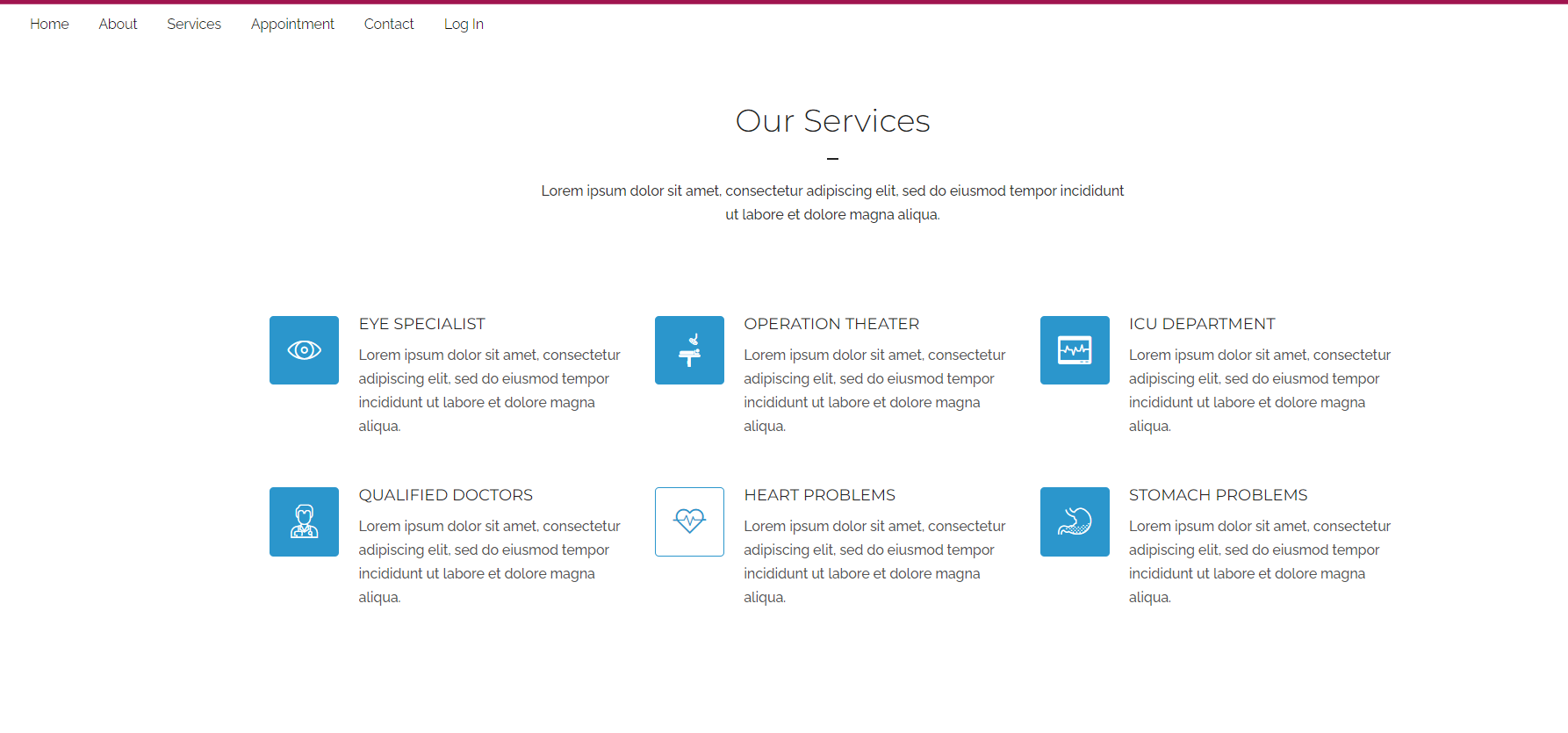


Figure 14

### **Appointment:**

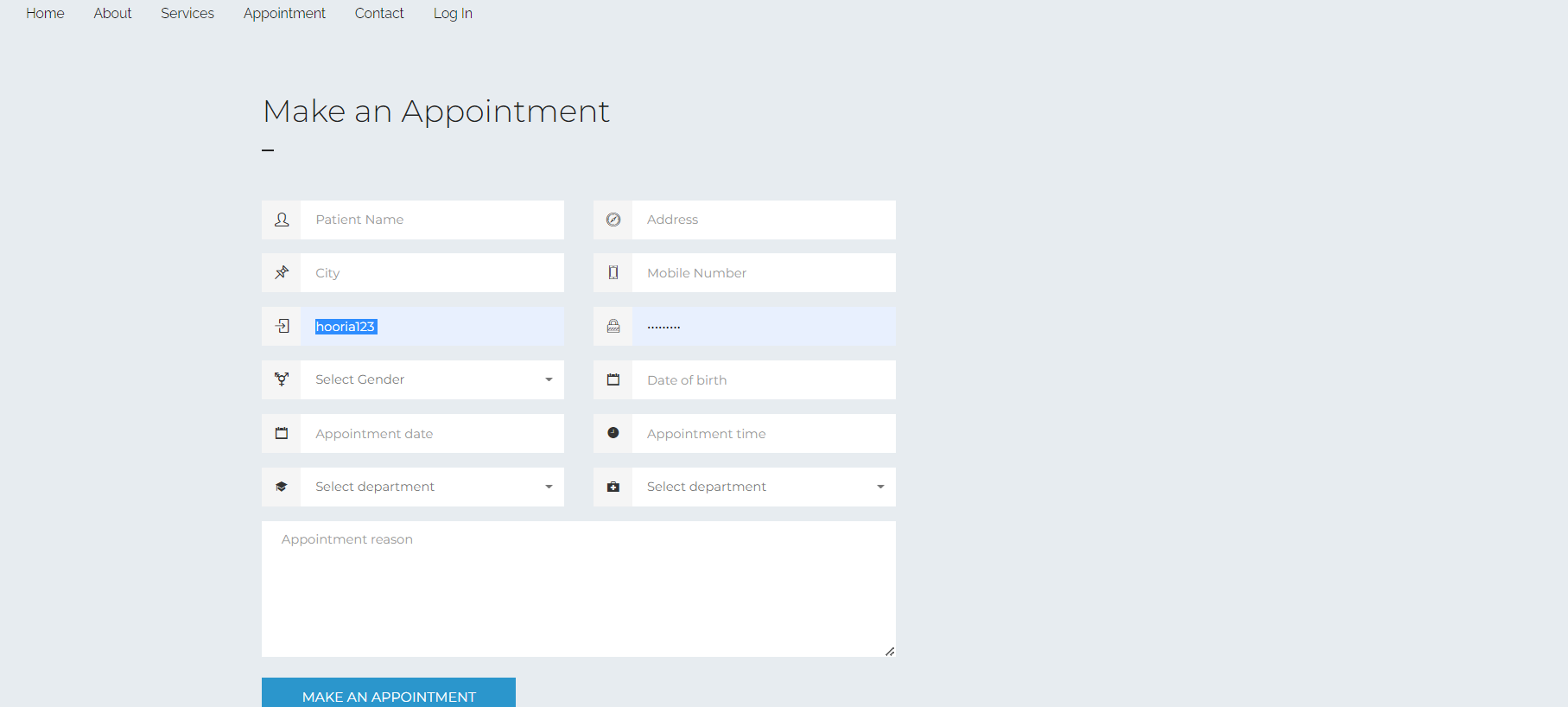


Figure 15

### **Doctor’s Dashboard:**

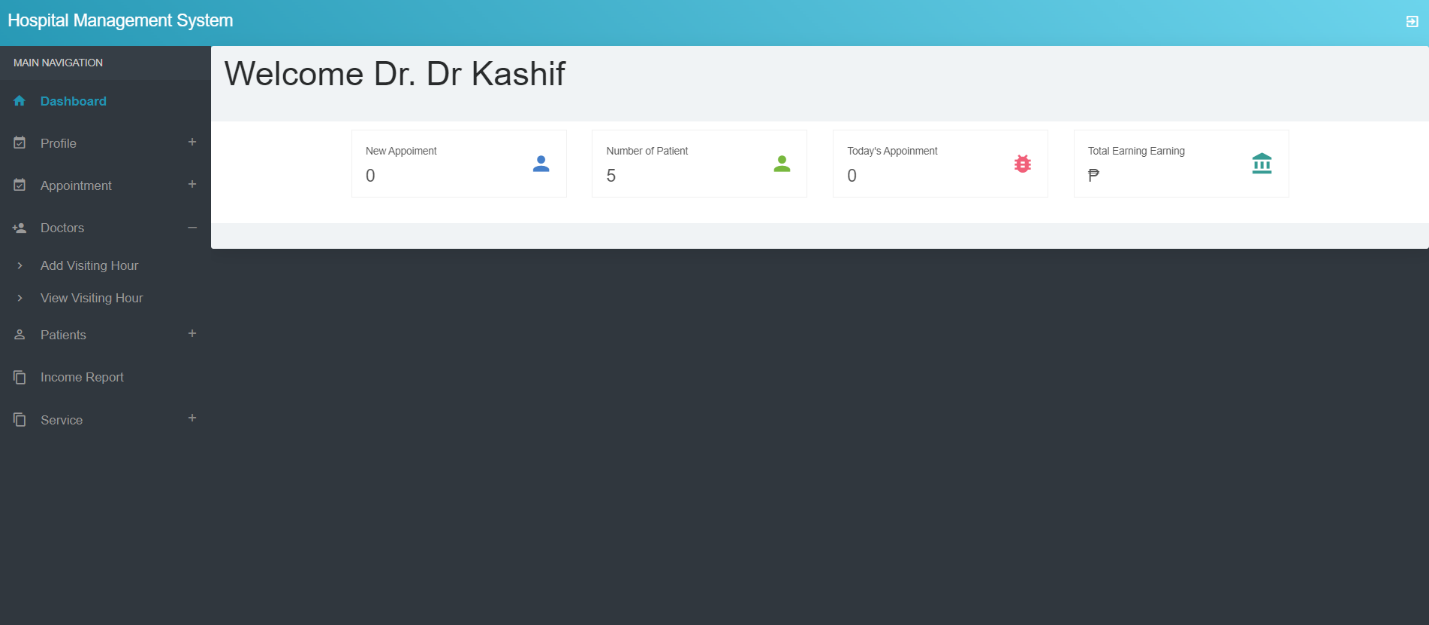


Figure 16

### **Patient’s Dashboard:**

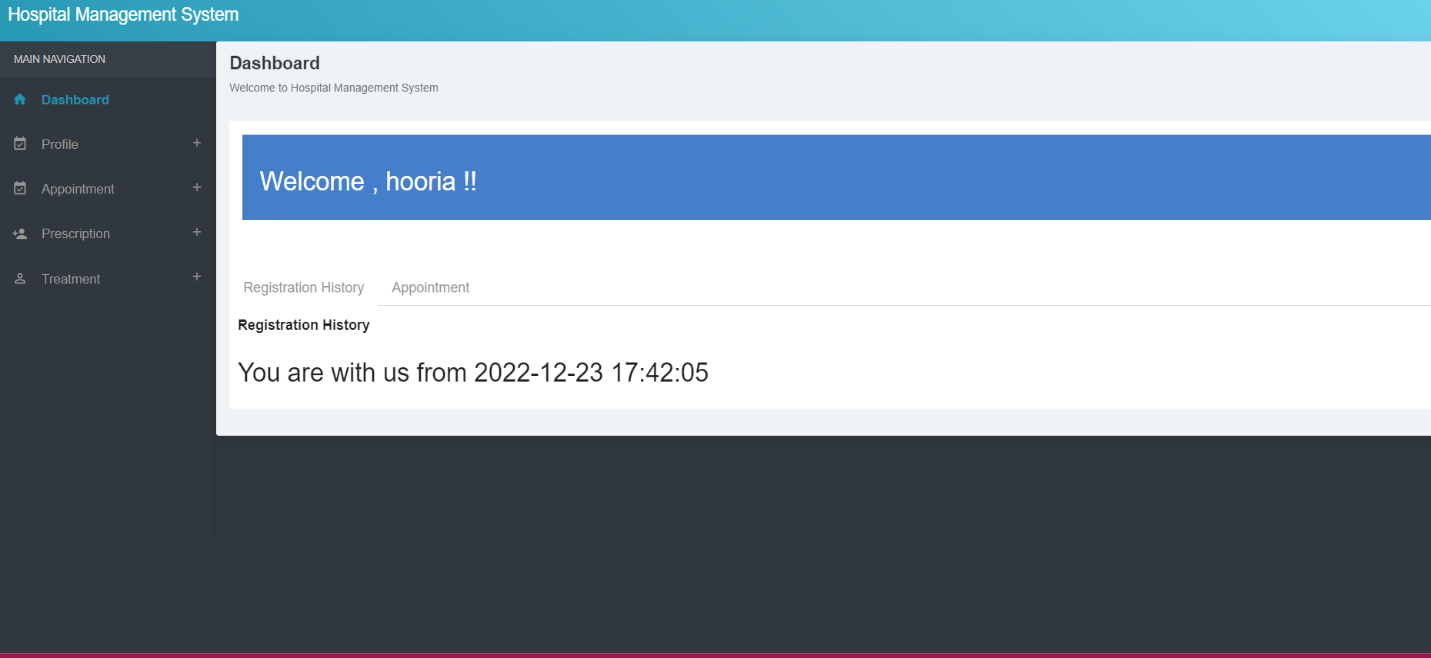


Figure 17

### **Admin’s Dashboard:**

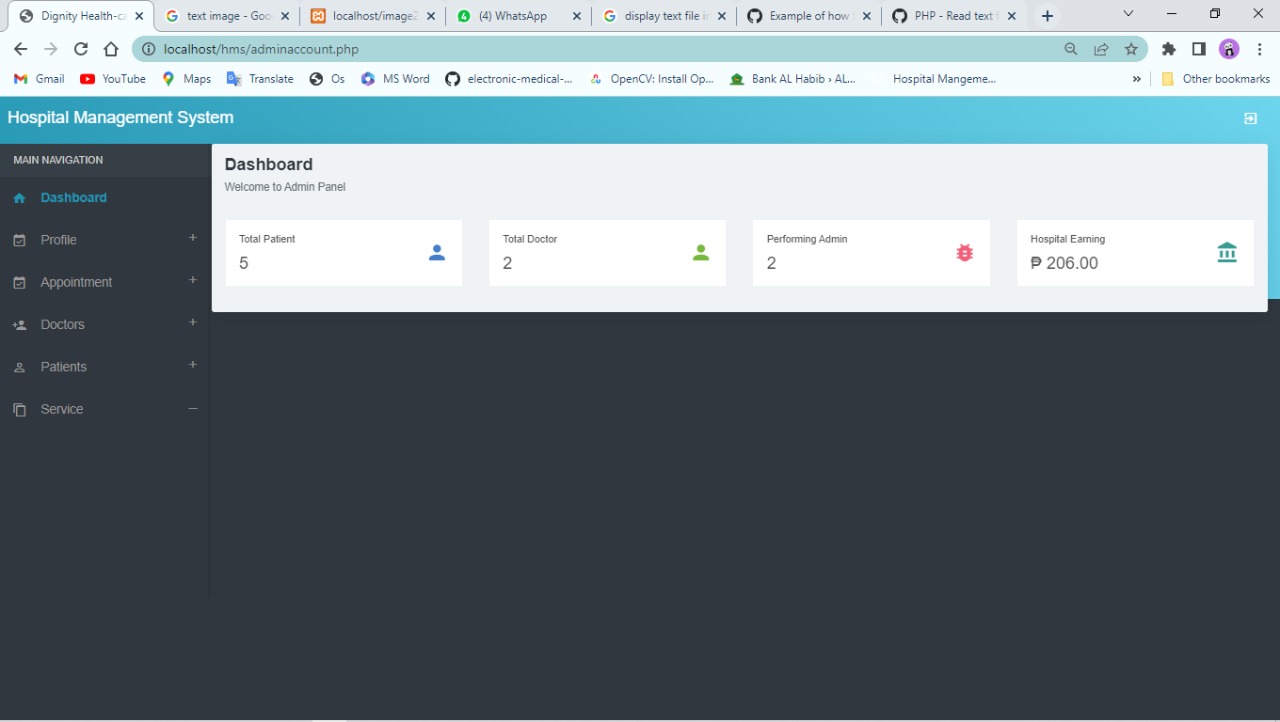


Figure 18

# **Data-Base:**

We have used MySQL for this project. And the relationships can be seen below in the figure which is known as ERD-Diagram (**Entity Relationship Diagram**).

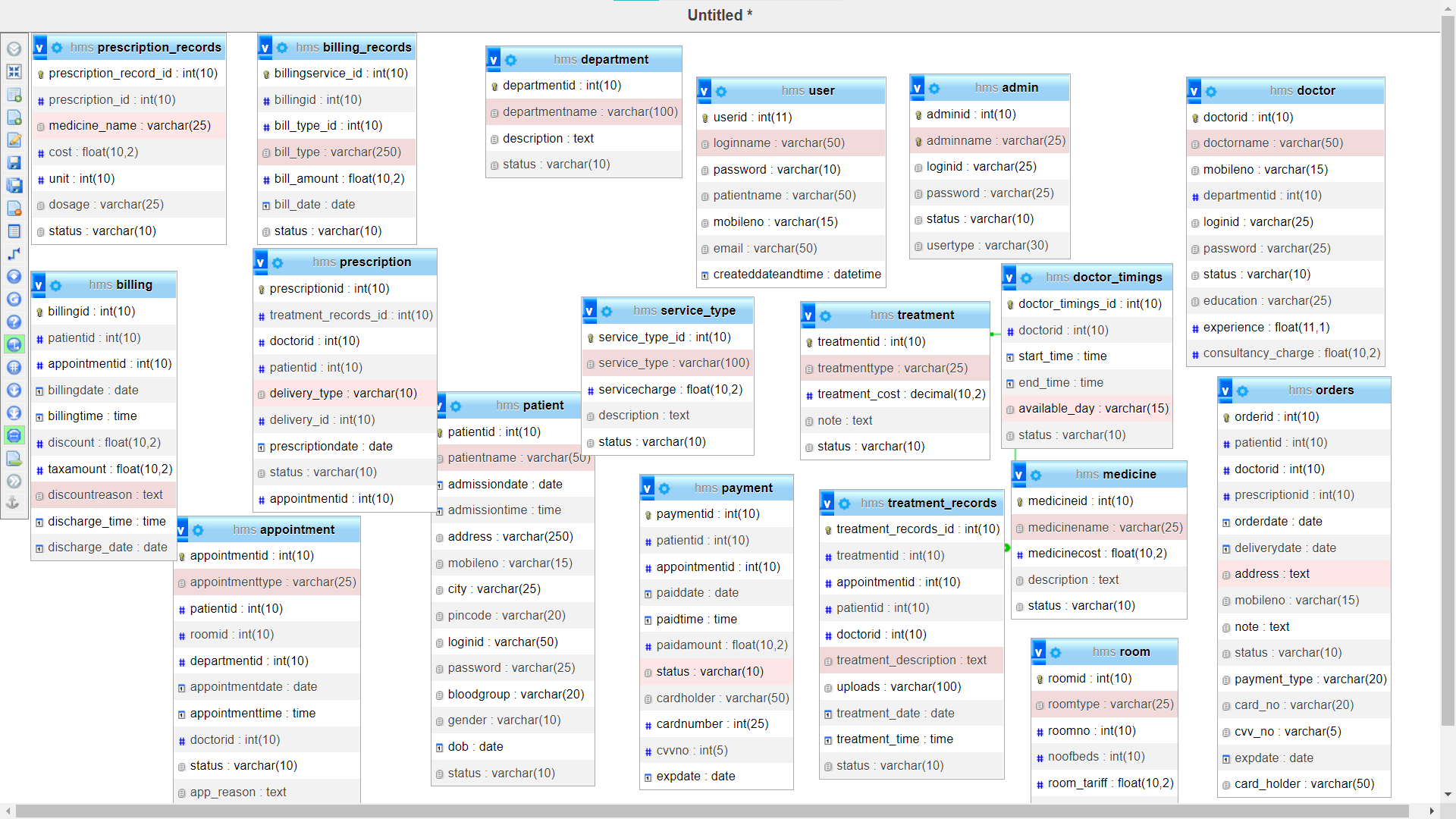


Figure 19

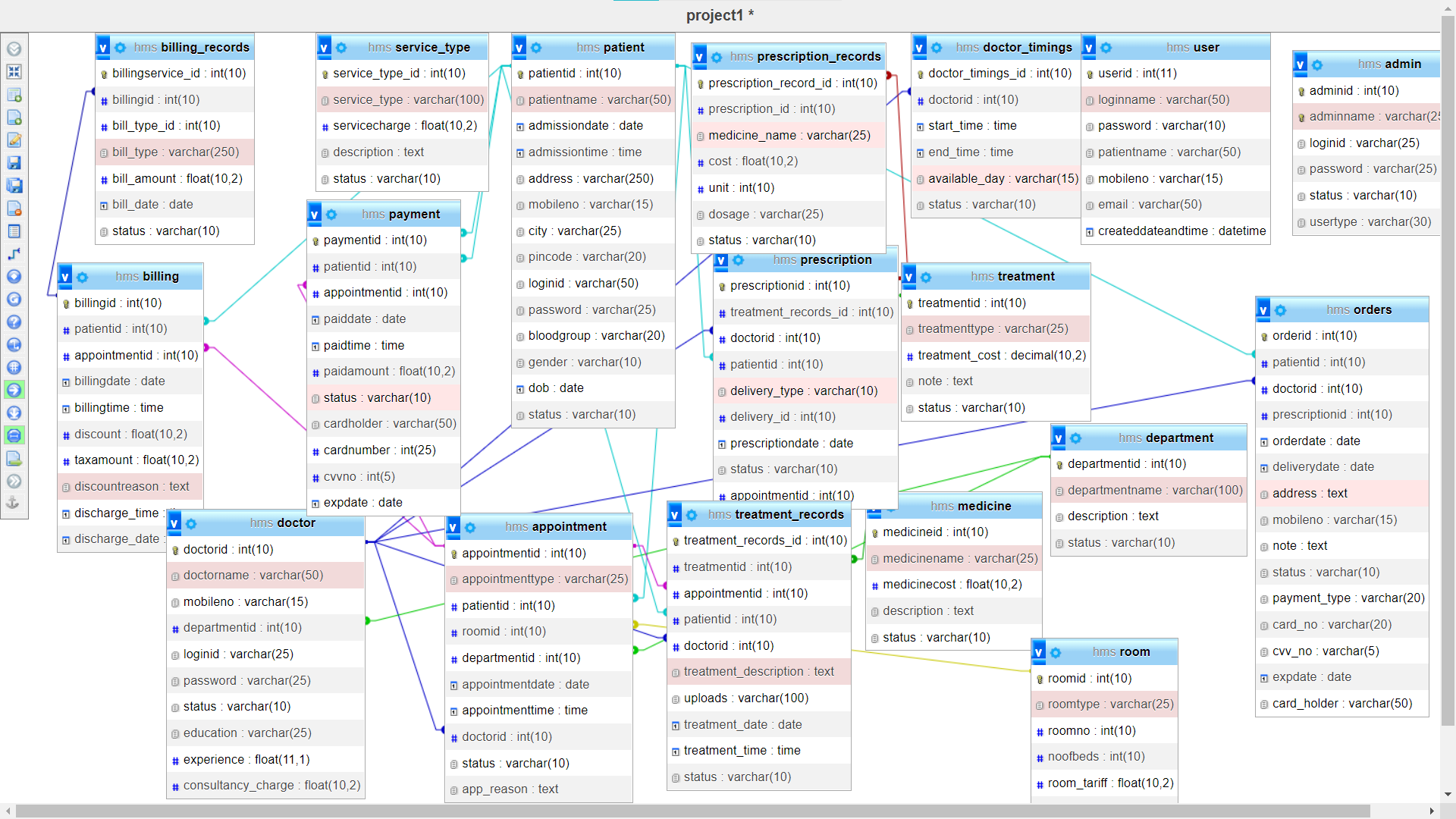


Figure 20

### **Database Table:**

We have 18 tables in our Database.

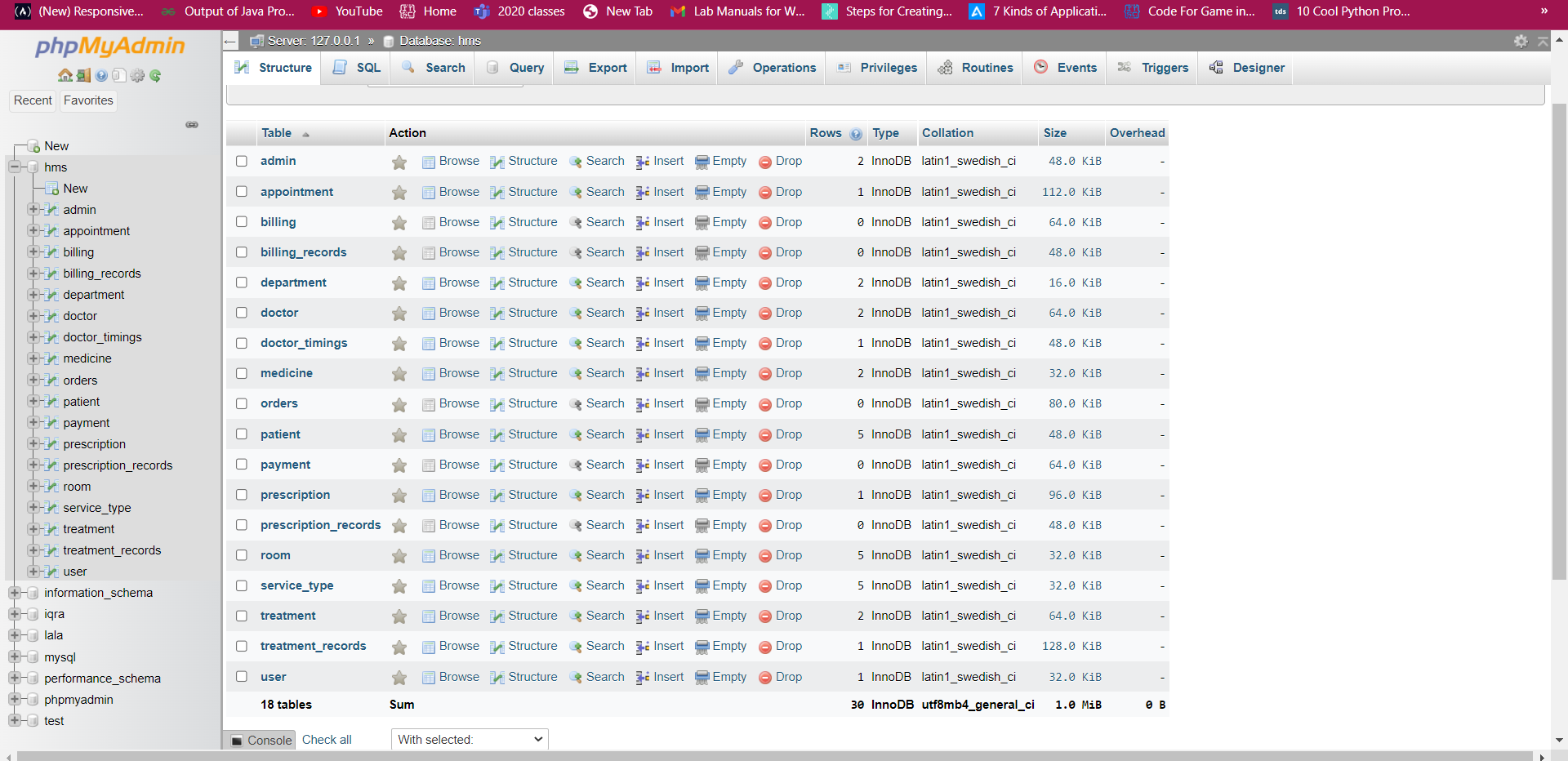


Figure 21

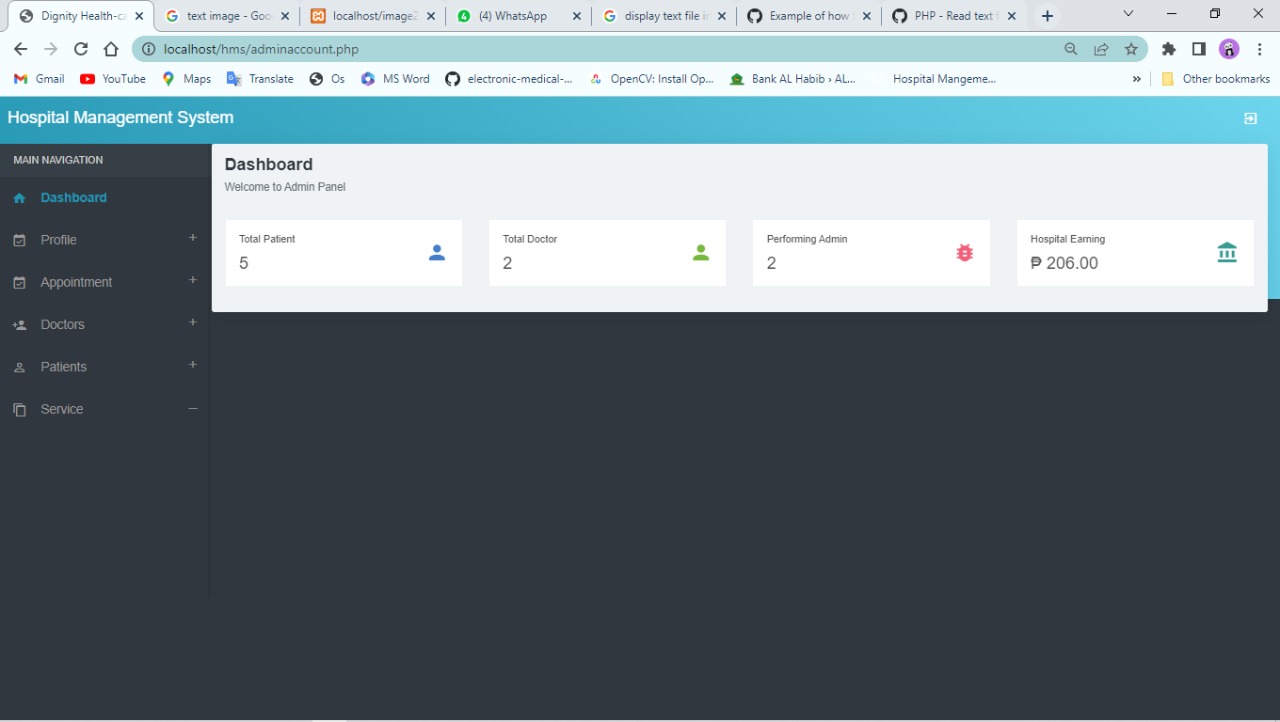


Figure 22

# **Conclusion:**

We have basically built a software system that surely can help in automating a hospital management system but there’s always a room for betterment in everything. We can implement BLE module or we can make this work for android systems too.

### **References:**

[1] <https://www.apm.org.uk/resources/find-a-resource/agile-project-management/>

[2]<https://www.healthit.gov/buzz-blog/electronic-health-and-medical-records/emr-vs-ehr-difference#:~:text=Electronic%20medical%20records%20(EMRs)%20are,have%20advantages%20over%20paper%20records>.

[3] <https://aws.amazon.com/what-is/ocr/>

[4] <https://www.klippa.com/en/blog/information/tesseract-ocr/?cn-reloaded=1>

[5]<https://www.interaction-design.org/literature/topics/ui-design#:~:text=User%20interface%20(UI)%20design%20is,e.g.%2C%20voice%2Dcontrolled%20interfaces>