

CS 232

INTRODUCTION TO DATABASES



HOSPITAL MANAGEMENT SYSTEM

FACULTY OF COMPUTER SCIENCE

DATABASE PROJECT SUBMISSION

Ahmed Nadeem 2018047

Hamza Najib 2018134

July 7, 2020

Contents

[Introduction 2](#_Toc45081064)

[Overview 2](#_Toc45081065)

[Technology used 2](#_Toc45081066)

[Testing 2](#_Toc45081067)

[Database design 2](#_Toc45081068)

[Cardinalities 2](#_Toc45081069)

[Figure 3](#_Toc45081070)

[Relationship 3](#_Toc45081071)

[Figure 4](#_Toc45081072)

[Normalization 4](#_Toc45081073)

[Figure 5](#_Toc45081074)

[Queries 5](#_Toc45081075)

# Introduction

With the increase in advent of technology and data storage requirement, data base management systems were created which are abbreviated as DBMS. They are used for records keeping as a soft as an alternative to keeping the data in hard form in which searching for a specific item can become very tedious. DBMS solve this kind of problems and provide with faster access to data.

## Overview

The Hospital Management System created for GIKI provides a way to store data online and create queries online and schedule of appointment. The records of the student can be catered to and solves the problem for writeups. This system also provides the ability to track the inventories and supplies to regulate the medical center and keep it updated.

# Technology used

For the number of options available in the market PHP, HTML 5, CSS 3 and MySQL were used. PHP was used to connect to MySQL databases for queries. HTML was used for front end. Bootstrap framework was used which is made of HTML, CSS and has premade components ready to deploy.

## Testing

The project was tested on XAMPP. Localhost server was created. XAMPP was preferred as it comes with important tools as MySQL, Apache server etc. It acted as an aid for testing the development of the source code.

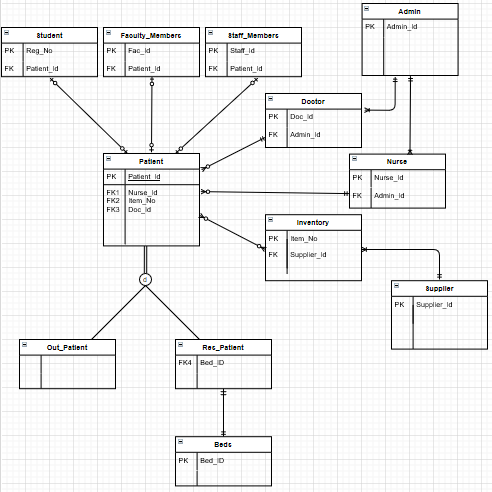
# Database design

As the focus of the project was on database design, conventions and rules were followed for creation of clean and a responsive database design that is compliant to standards and norms.

## Cardinalities

* A patient can be either of two types: out-patient or resident patient.
* A patient may be a student, a faculty member or a staff member and vice versa.
* One doctor can attend zero to many patients, where as a patient is attended by only one doctor.
* One nurse can attend zero to many patients, where as a patient is attended by only one nurse.
* An item may be given to many patients and a patient may use many items.
* An admin member may manage many doctors or nurses, but doctors and nurses are managed by only one admin member.
* An item is provided by exactly one supplier, but a supplier can provide many items.
* Exactly one bed is provided to exactly one resident patient.

### Figure

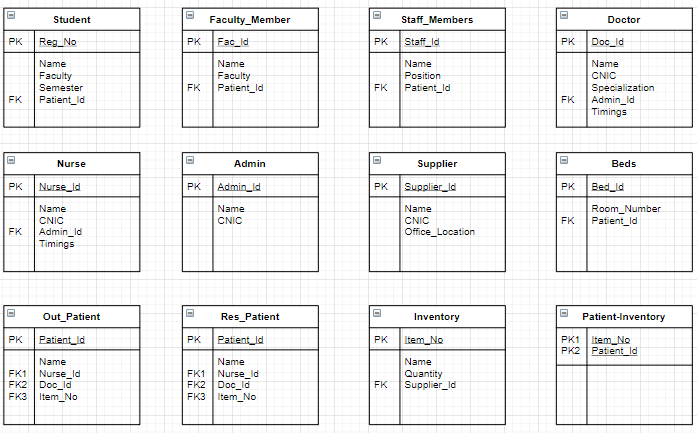


## Relationship

The relationship designed was as follows:

* The patient’s specialization is mapped as subclasses only since it works perfectly for disjoint and total participation.
* A relation is made for a relationship between inventory and patient determining the number of items a patient has consumed or been provided with.
* The remaining relations with a one-to-one or a one-to-many relationship have been provided with foreign keys respectively.

### Figure

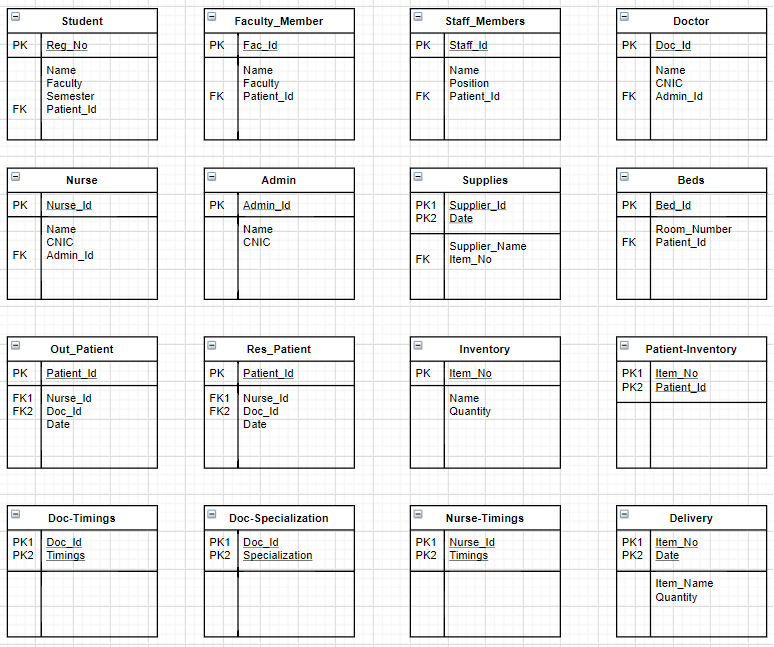


## Normalization

Then the database was normalized to 3NF to reduce data anomalies. The details for the reduction in anomalies are:

* The relation Patient-Inventory was created to resolve the multi-valued attribute as the name and quantity (of the item being used by a patient) will then have to be repeated. (1NF)
* The relation Doc-Timings and Nurse-Timings were created to resolve the anomaly of same doctors and nurses available at different hours. (1NF)
* The relation Doc-Specialization was created to resolve the anomaly of same doctor having more than one specialization. (1NF)
* The relation Delivery was created by decomposing the ‘Supplies’ relation to avoid redundancies of the supplier’s name and id being mentioned again and again with the item bought and its quantity. (3NF)

### Figure

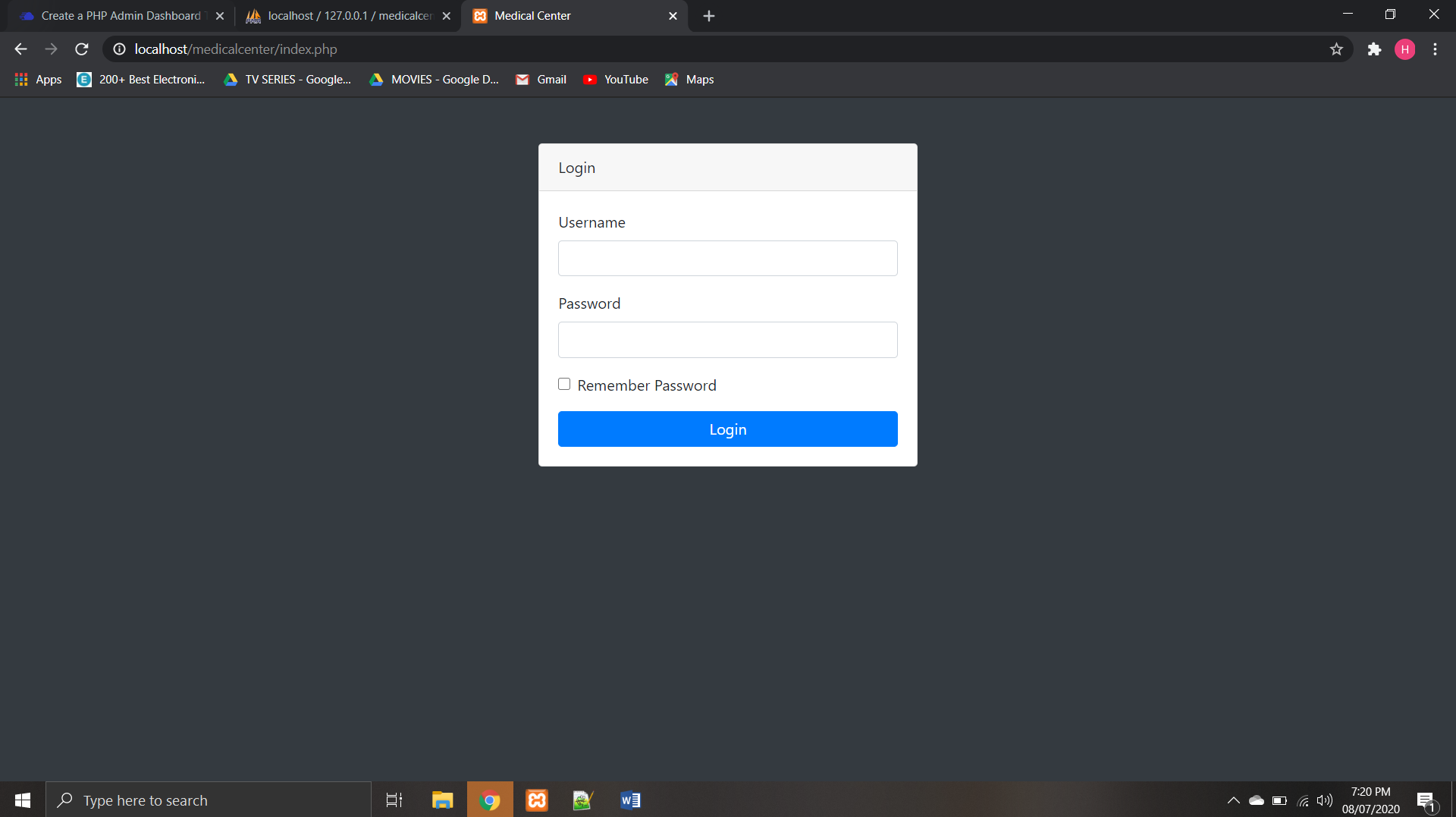


# Database Creation Queries

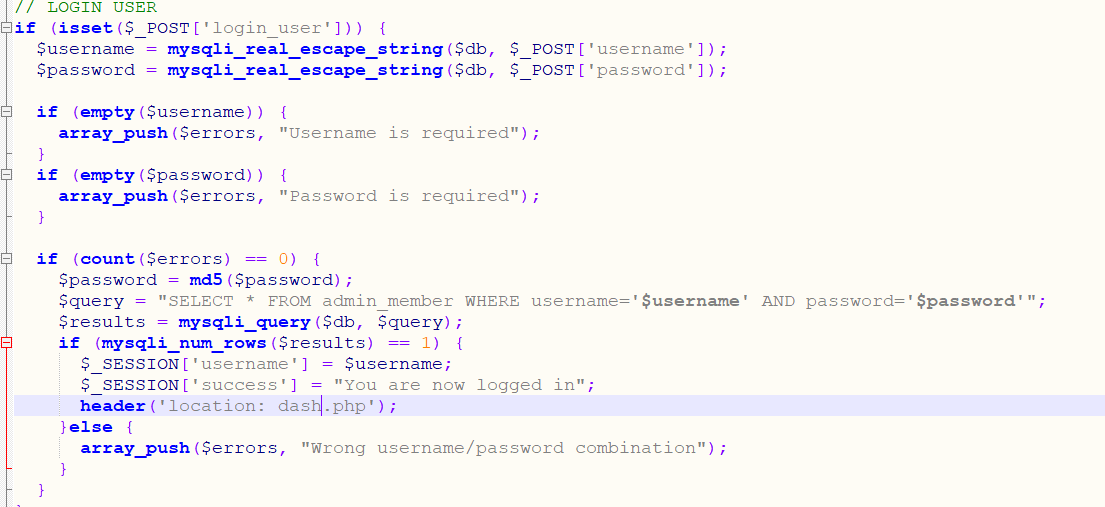
The queries for the creation of schemas in DBMS are as follow:

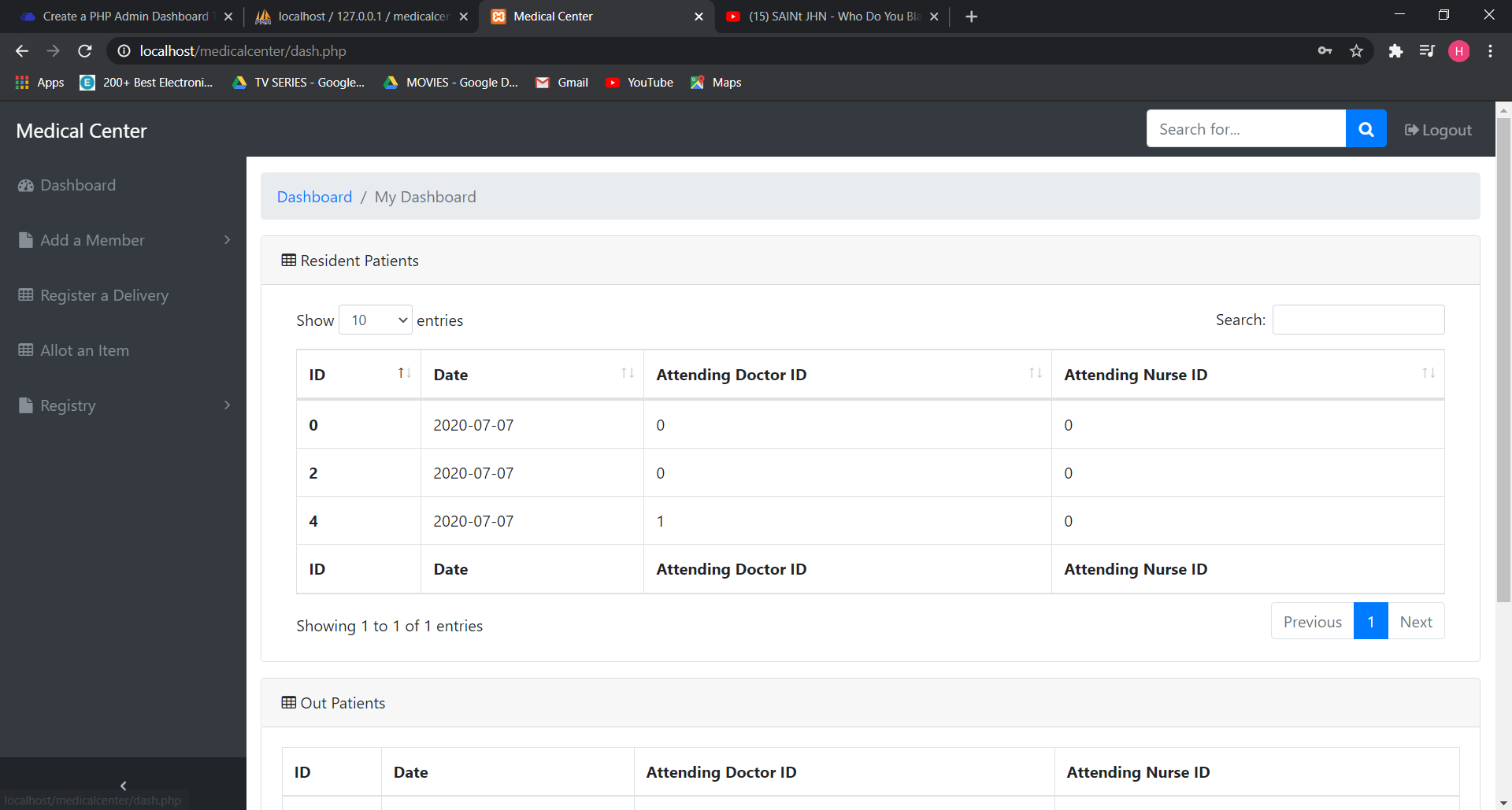
1. CREATE TABLE `admin\_member` ( `admin\_id` int NOT NULL, `username` varchar(40) NOT NULL, `email` varchar(50) NOT NULL, `password` varchar(40) NOT NULL)
2. CREATE TABLE `doctors` ( `doc\_id` int NOT NULL, `name` varchar(40) NOT NULL, `admin\_id` int NOT NULL, primary key (doc\_id), FOREIGN key (admin\_id) references admin (id) )
3. CREATE TABLE `nurses` ( `nurse\_id` int NOT NULL, `name` varchar(40) NOT NULL, `admin\_id` int NOT NULL, primary key (nurse\_id), FOREIGN key (admin\_id) references admin\_member (admin\_id) )
4. CREATE TABLE `student` ( `reg\_no` int NOT NULL, `name` varchar(40) NOT NULL, `faculty` varchar(4) NOT NULL, `patient\_id` int NOT NULL, primary key (reg\_no))
5. CREATE TABLE `faculty\_member` ( `fac\_id` int NOT NULL, `name` varchar(40) NOT NULL, `faculty` varchar(4) NOT NULL, `patient\_id` int NOT NULL, primary key (fac\_id))
6. CREATE TABLE `staff\_member` ( `staff\_id` int NOT NULL, `name` varchar(40) NOT NULL, `position` varchar(40) NOT NULL, `patient\_id` int NOT NULL, primary key (staff\_id))
7. CREATE TABLE `inventory` ( `item\_no` int NOT NULL, `name` varchar(40) NOT NULL, `quantity` int NOT NULL, primary key (item\_no ) )
8. CREATE TABLE ‘supplies’ ( ‘supplier\_id’ int NOT NULL, ‘dod’ date NOT NULL, ‘name’ varchar(40) NOT NULL, ‘item\_no’ int NOT NULL, primary key (supplier\_id, dod), FOREIGN key (item\_no) refrences inventory (item\_no))
9. CREATE TABLE `out\_patient` ( `patient\_id` int NOT NULL, `dod` date NOT NULL ,`doc\_id` int NOT NULL, `nurse\_id` int NOT NULL , primary key (patient\_id), FOREIGN key (doc\_id) references doctors (doc\_id), FOREIGN key (nurse\_id) references nurses (nurse\_id) )
10. CREATE TABLE `res\_patient` ( `patient\_id` int NOT NULL, `dod` date NOT NULL ,`doc\_id` int NOT NULL, `nurse\_id` int NOT NULL , primary key (patient\_id), FOREIGN key (doc\_id) references doctors (doc\_id), FOREIGN key (nurse\_id) references nurses (nurse\_id) )
11. CREATE TABLE `beds` ( `bed\_id` int NOT NULL,`room\_no` int NOT NULL, `patient\_id` int NOT NULL , primary key (bed\_id), FOREIGN key (patient\_id) references res\_patient (patient\_id) )
12. CREATE TABLE `patient\_inventory` ( `item\_no` int NOT NULL, `patient\_id` int NOT NULL , primary key (item\_no,patient\_id), FOREIGN key (item\_no) references inventory (item\_no) )
13. CREATE TABLE `doc\_timings` ( `doc\_id` int NOT NULL, `timings` varchar(20) NOT NULL , primary key (doc\_id,timings), FOREIGN key (doc\_id) references doctors (doc\_id) )
14. CREATE TABLE `nurse\_timings` ( `nurse\_id` int NOT NULL, `timings` varchar(20) NOT NULL , primary key (nurse\_id,timings), FOREIGN key (nurse\_id) references nurses (nurse\_id) )
15. CREATE TABLE `doc\_specialization` ( `doc\_id` int NOT NULL, `specialization` varchar(20) NOT NULL , primary key (doc\_id,specialization), FOREIGN key (doc\_id) references doctors (doc\_id) )
16. CREATE TABLE `delivery` ( `item\_no` int NOT NULL, `dod` date NOT NULL , name varchar(20) not null, quantity int not null, primary key (item\_no,dod), foreign key (item\_no) references inventory (item\_no))

# Website



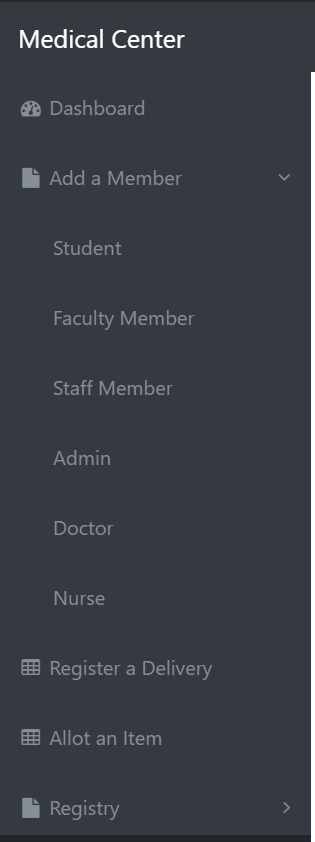
The website opens with a login page which allows the admin to log-in to the panel with his credentials. The query used in this case checks the credentials against the ones saved in the admin table and takes the user to the admin panel if the credentials match.



The admin then gets connected to the dashboard.  
 

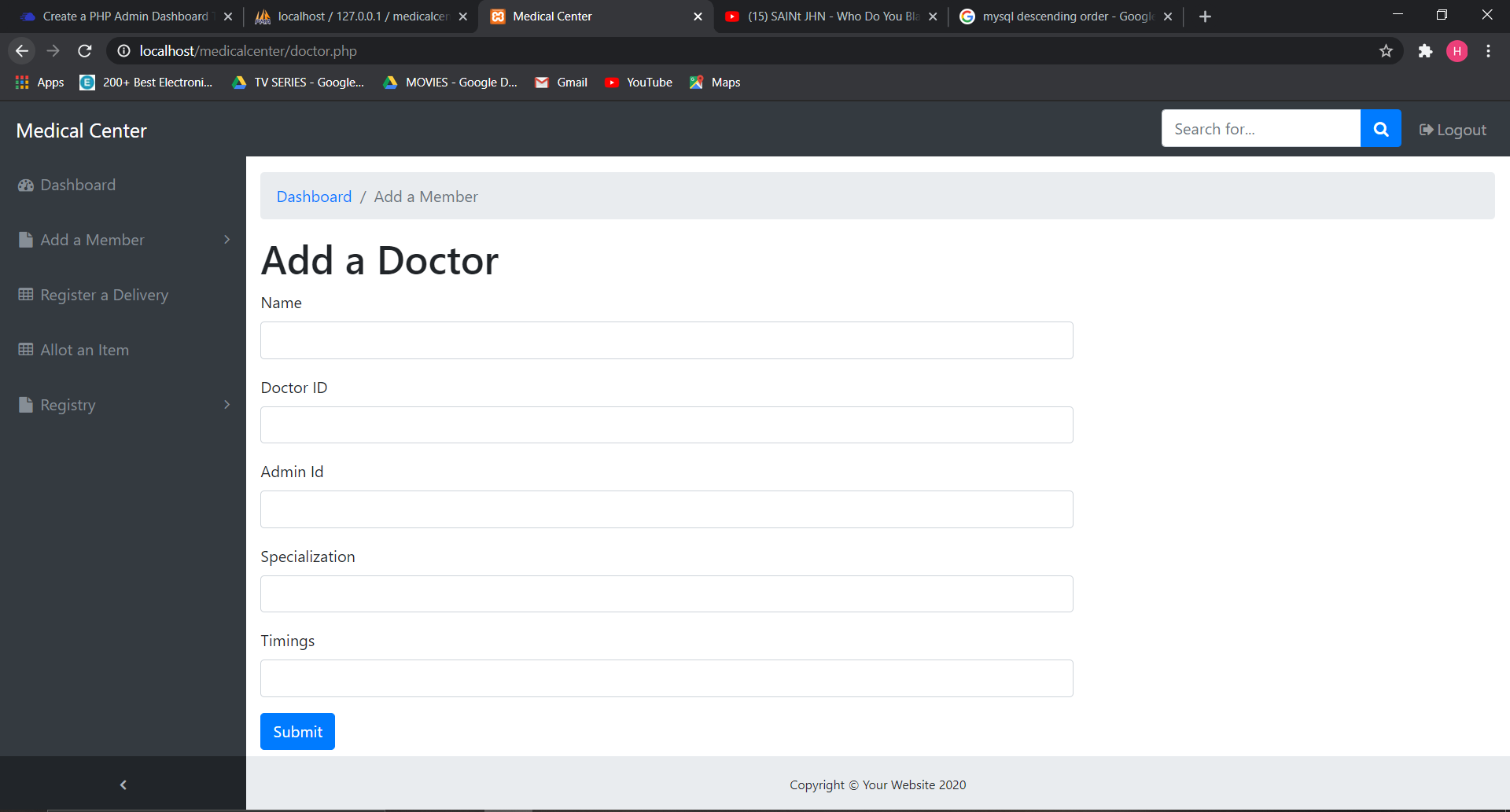
The dashboard shows the all the resident and out patients ordered by the most recent date. The side navbar contains the other accessible components of the hospital management system.

The ‘Add a Member’ item in the side navbar helps you add a patient or a member into the date base.

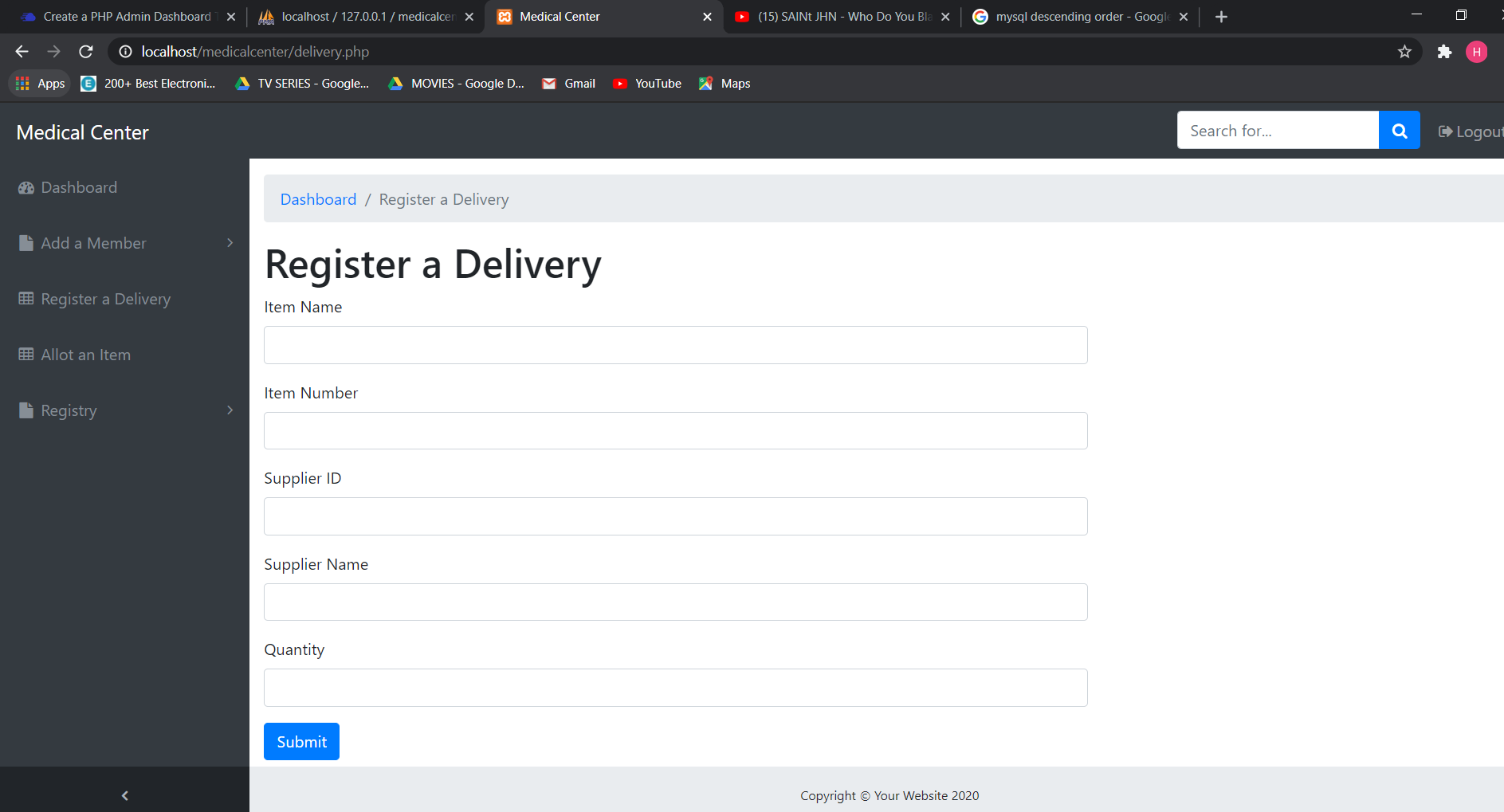


In order to register any member, the forms for each members are uploaded on their respective pages that adds the data to the database.

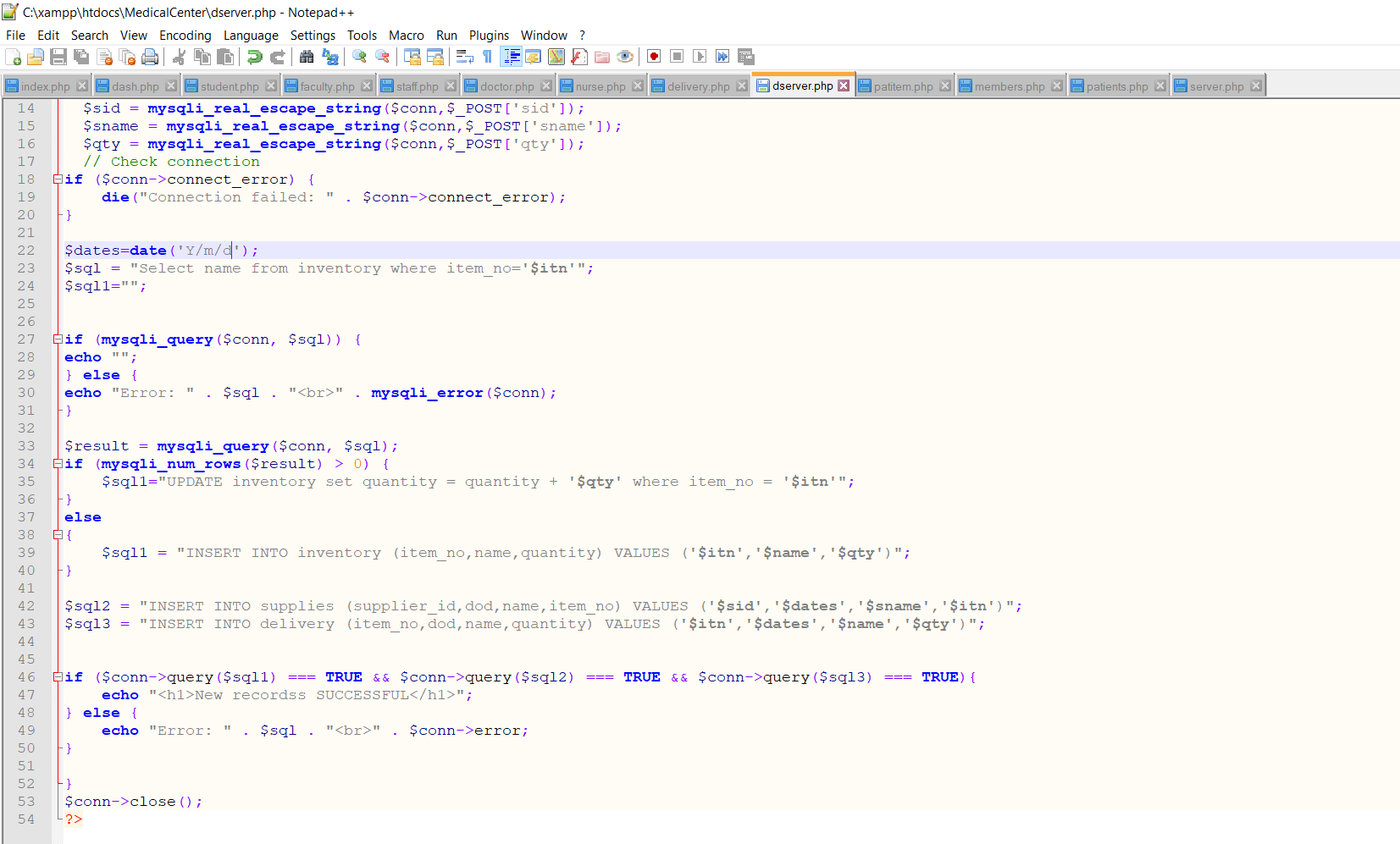
The registry on the other hand displays data of patients and medical staff.



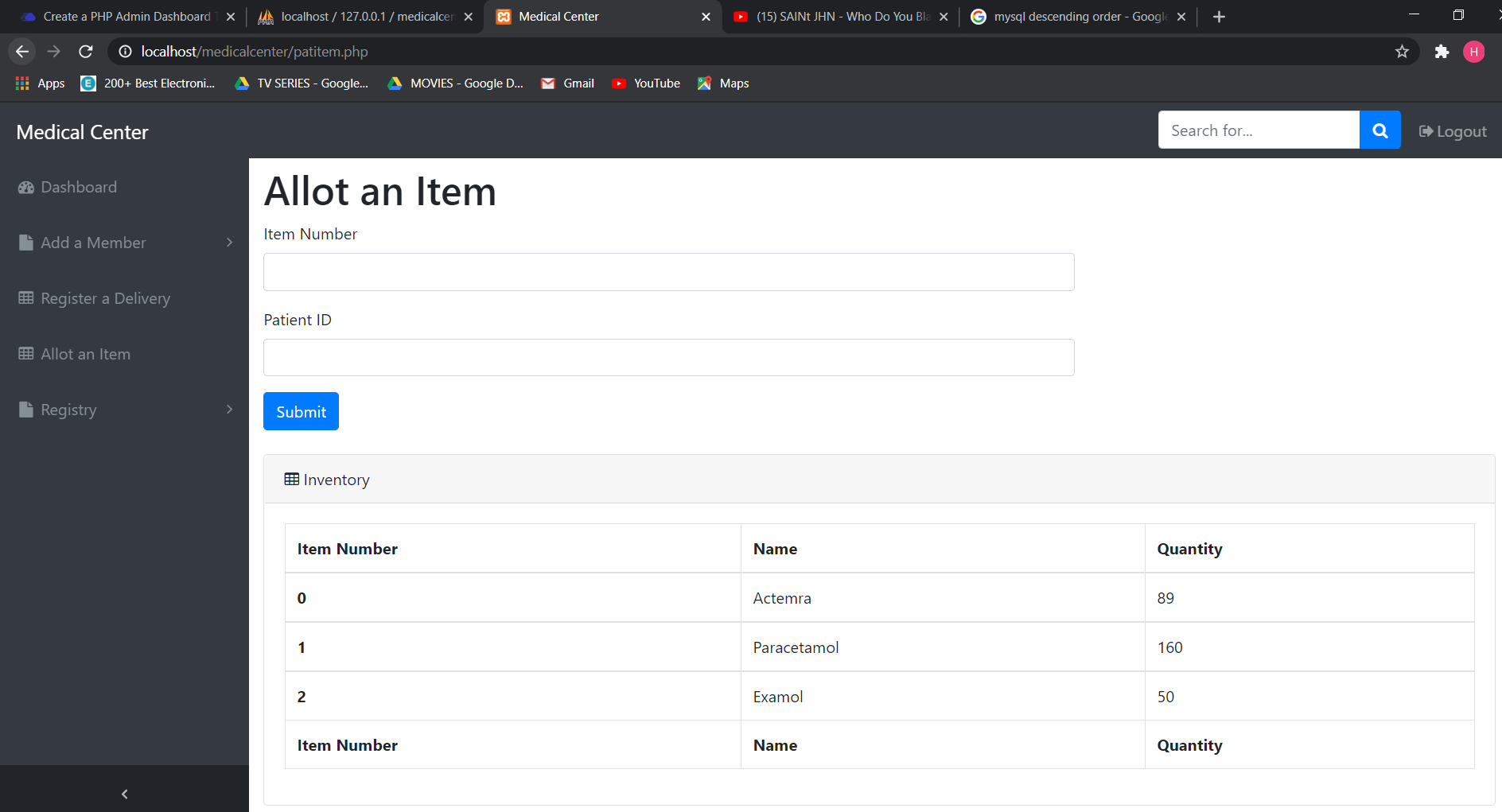
The above snip is a template of how all the members are added to the database. All the tables have foreign key and primary key constraints, so there are very less chances for anomalies to occur.



This component registers the delivery of equipment and items into the table inventory. If a new item gets delivered a new record is made for that entry, but if an old item is brought the former row’s quantity is updated.



The ‘Allot an Item’ component registers an entry in the database of an item being allotted to a patient and reduces the quantity of the item by 1.



The ‘Registry’ tab contains all the data of the patients and doctors.

