Documentation for Database Project

Hospital Management System Database

Moldovan Christine Matilda

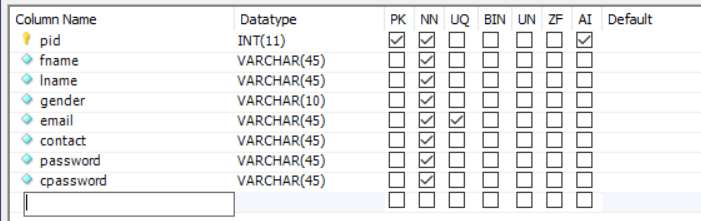
Group 30323

1. Short presentation of the database

The database I created should serve a Hospital Management because it keeps track of the pacients, who can create appointments at the doctors from the hospital. The doctors can than write prescriptions based on the pacients appointments. The admin is the one responsible to add new doctors, with possibilities to add also pacients and appointments.

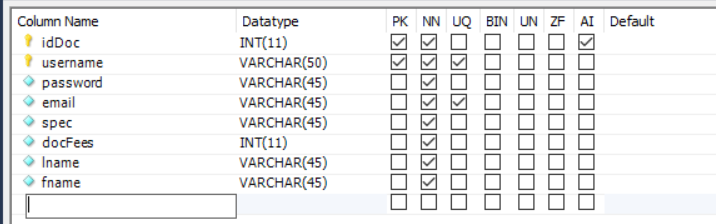
* 1. The Tables

Pacient table:



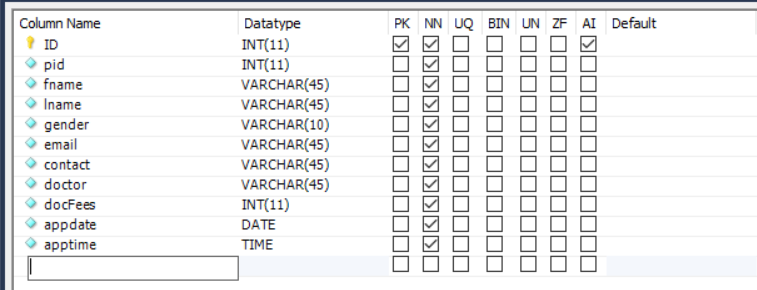
Each pacient has a unique id “pid”, name, surname, gender, unique email, contact, password, and a check password. The pacient must enter all of them to be able to access the system.

Doctor table:



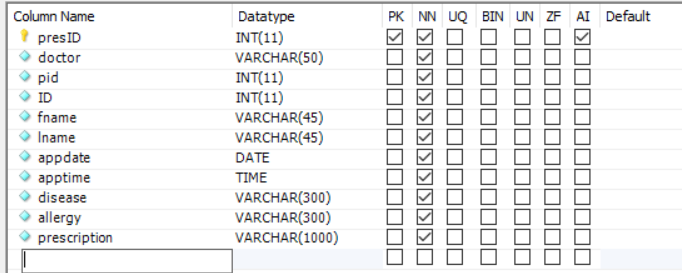
Each doctor has a unique id “idDoc”, a unique username to be able to login, password, unique mail, spec for specialization, doctor fees, name and surname.

Appointments table:



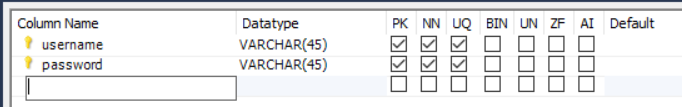
The appointments table needs the id, name, surname, gender, email and contact of the pacient and the username, doctor fees of the doctor. Besides that, it is mandatory to enter an appointment data and time. It has also a unique id for each appointment created.

Prescription table:



The prescription table requires the ID of the appointment, the id of the pacient and the username of the doctor. It takes the name and surname of the pacient, the appointment date and time from the appointment table. Here only the doctor is able to add descriptions of the disease, allergy and finally the prescription for each pacient.

Admin table:



The admin table consists of his username and password, both being unique.

* 1. The Diagram

The relationship between the tables and the columns is the following:

Starting with the Pacient table:

* Can have one and only unique email, password, gender, name, surname and contact

Doctor table:

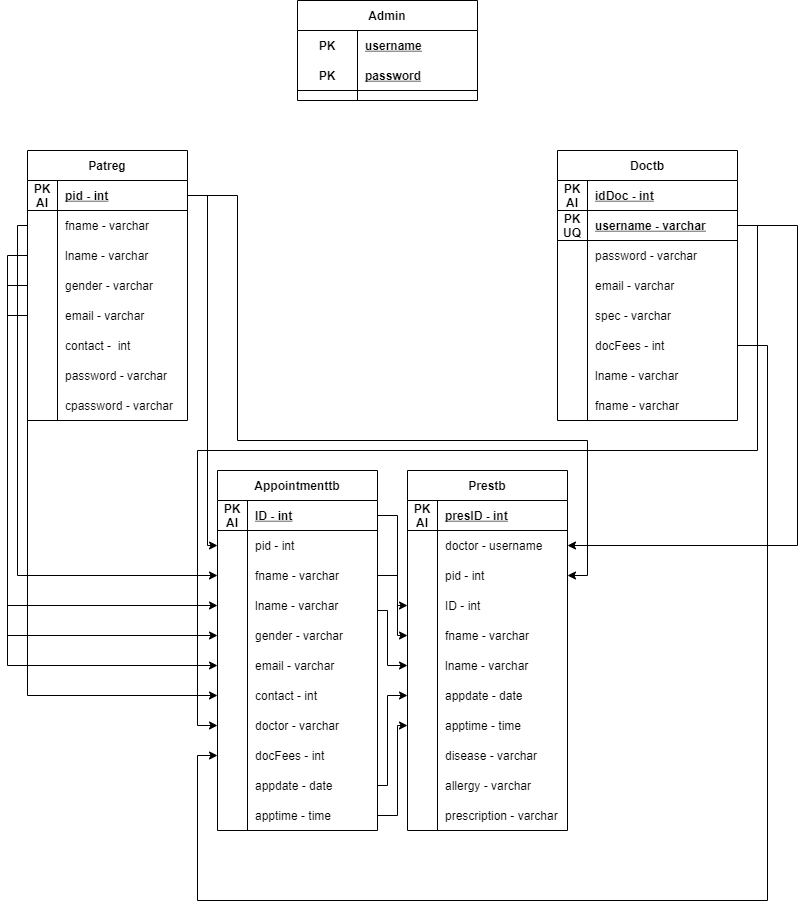
* Can have one and only username, name, surname, password, email (unique), specialization, fee

Appointment table:

* The appointment table can have 0 to many pacient registrations
* A pacient can register 0 to many appointments, to one or many doctors
* A doctor can resolve 0 to many appointments based his appointments, to one and many pacients
* An appointment is associated to one and only doctor
* An appointment is associated to one and only pacient
* A pacient can make an appointment to different doctors, and appointments can be made by various pacients (many-to-many)
* Each pacient can have many appointments (one-to-many)

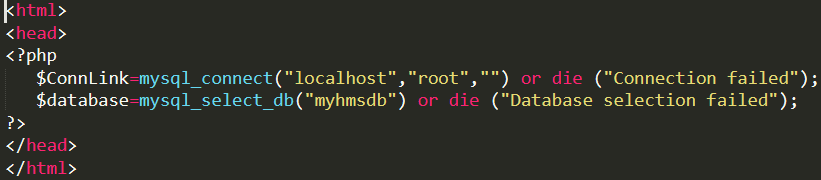
Prescription table:

* A doctor can create 0 to many prescriptions based on his appointments, to one or many pacients
* A pacient can receive 0 to many prescriptions completed by the doctor
* A prescription is associated to one and only appointment (one-to-one)



* 1. Connection to the DataBase

For the localhost server I used Mowes. The connection of the PHP Code to the Database is accomplished by the following code:



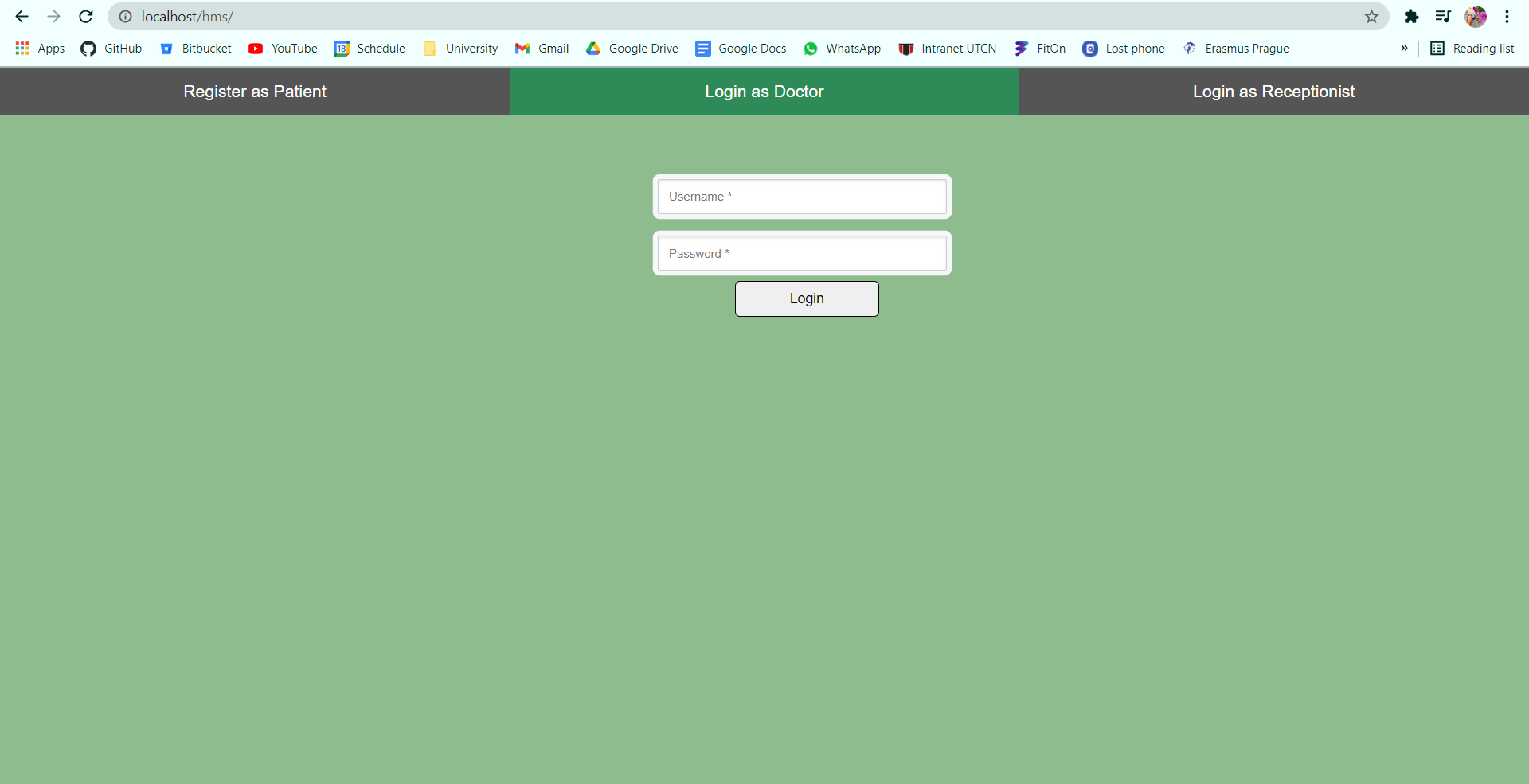
The database I created is called “myhmsdb”, and it is the one that is selected in the above code.

2. The Login Form

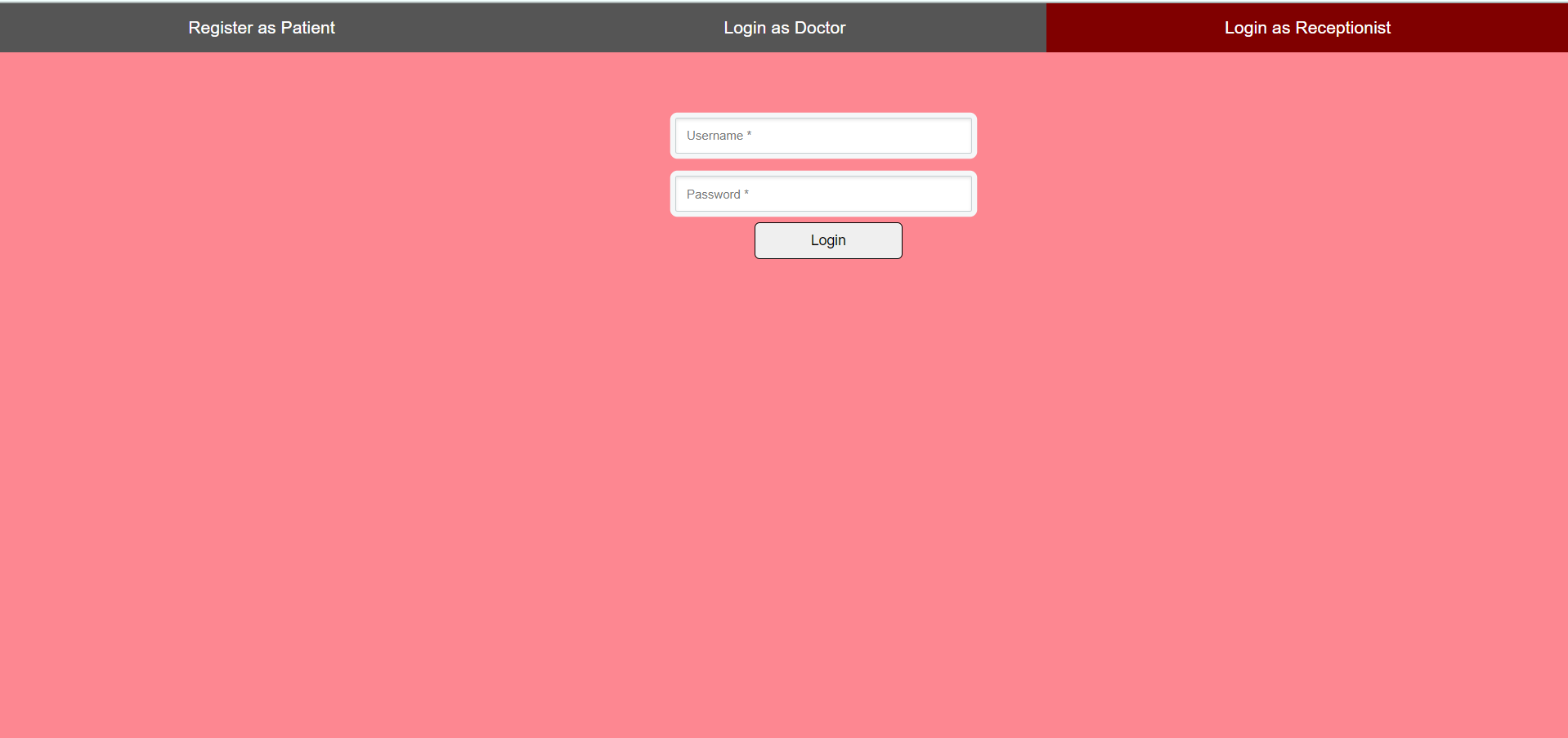
The login form is divided in three main parts in one index, with another part in another index.

Each part is assigned to a specific user.

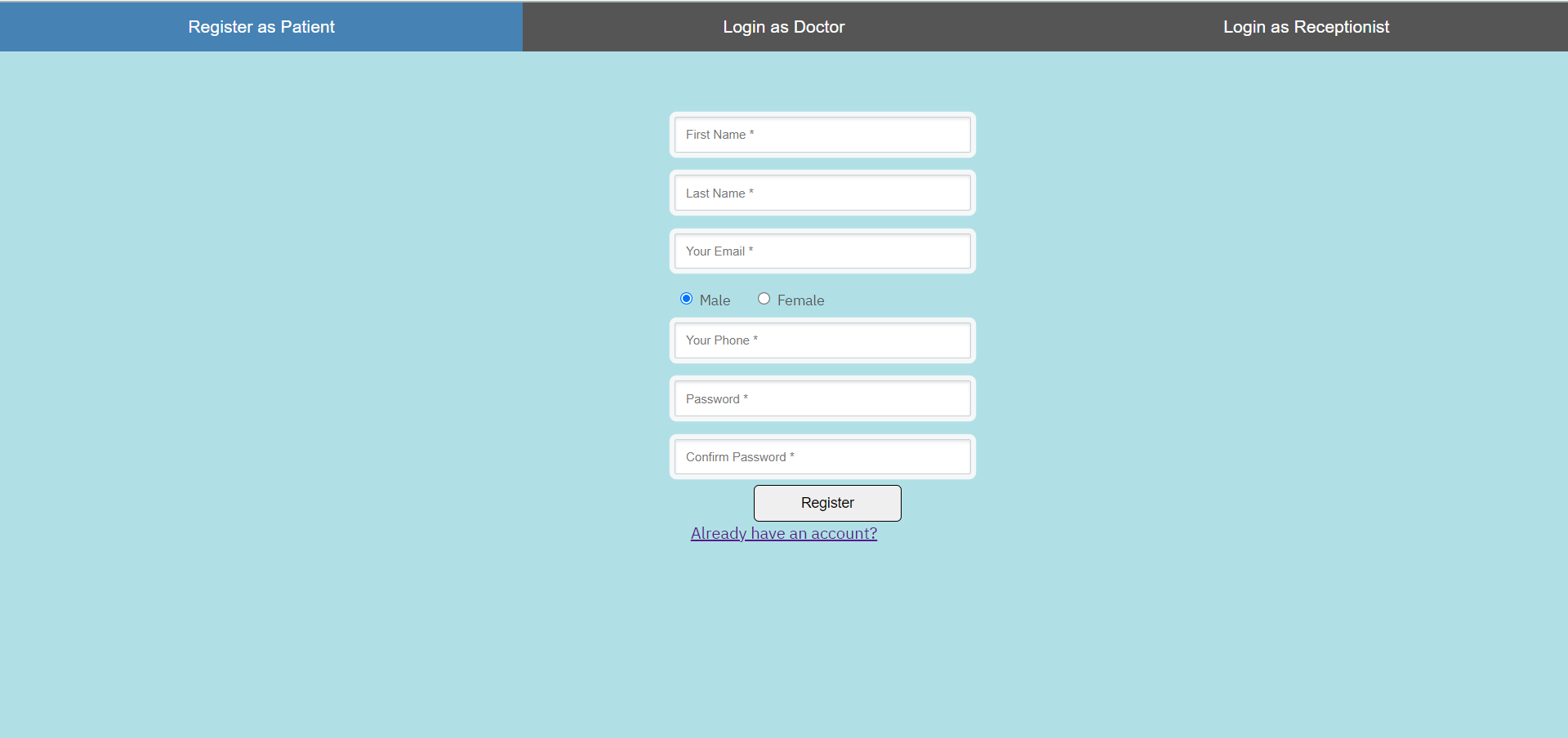
For the **doctor** to login, the tab is the middle one:



For the **admin** login is the right tab:



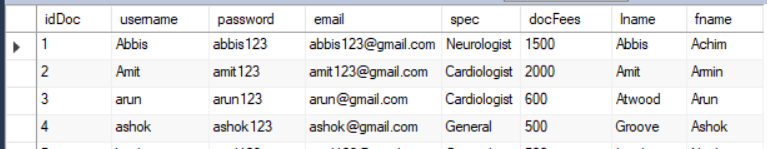
And finally for the **pacient** the tab is the left one:



Here, the pacient can register to enter in the system or press the text below the button Register, „Already have an account?” to login if he already has an account. This way the user is redirectioned to the following index page:

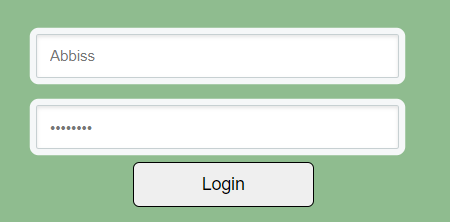


For example, if a doctor wants to login, he must enter his username and password that is stored in the database:

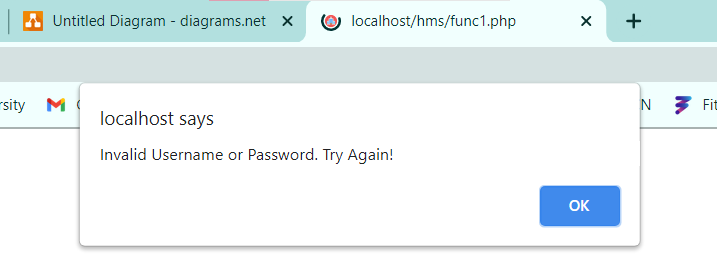


Let’s consider that dr. Abbis Achim wants to login, he will need to enter the username:*Abbis* and password:*abbis123*

If he enters his username or password wrong, like this, he double the „s”:



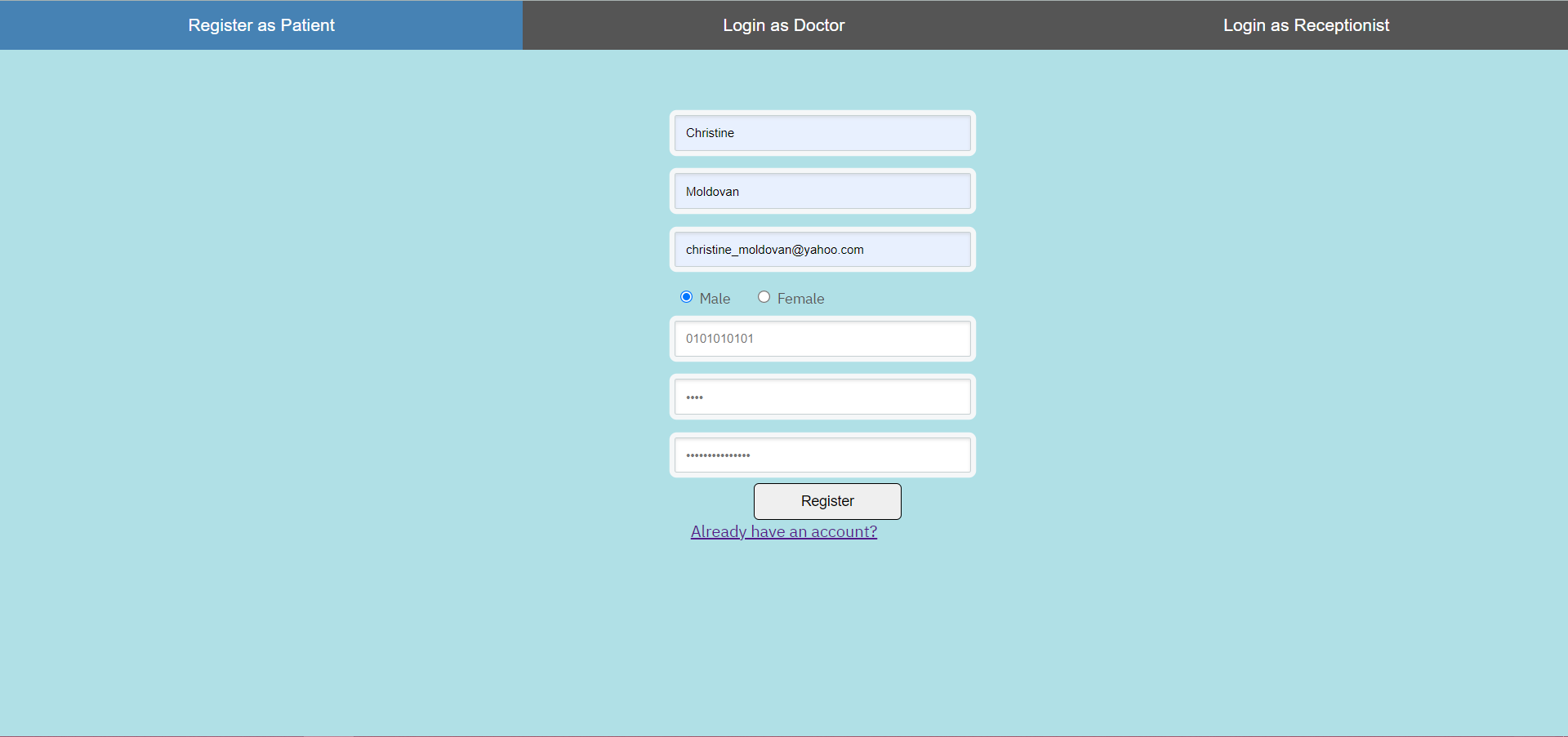
It will show us a pop-up saying to login again:



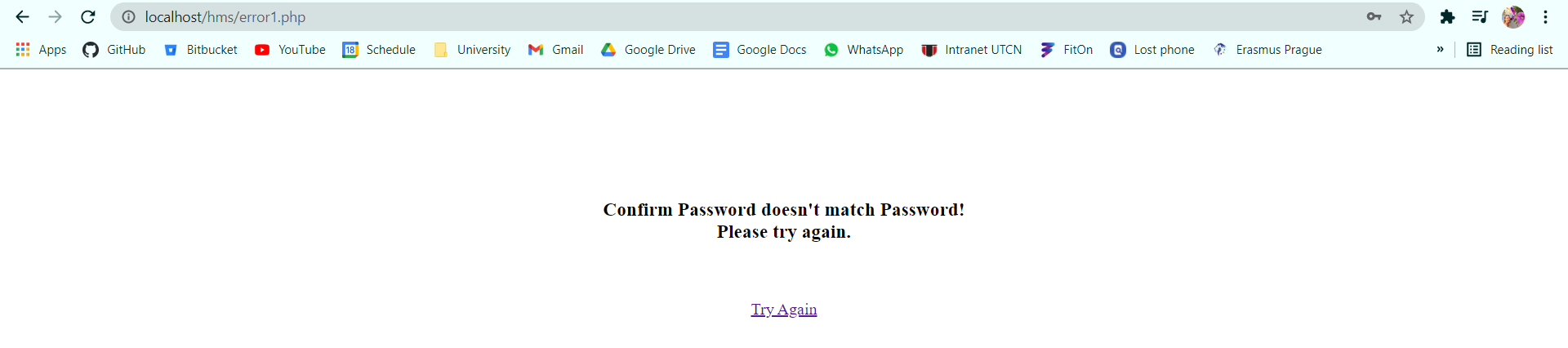
But if he enters in the correct credentials, he will be redirected to his main page:



If a pacient wants to login by registering, he is required to complete the following text areas:



Let’s say he typed the first password:*aaa* and the second he mistyped it and wrote *aaaa*. As the two passwords don’t match, a message will show up after pressing register:



But if he typed in the same passwords in both fields, he will be added to the database and redirected to the pacient page:



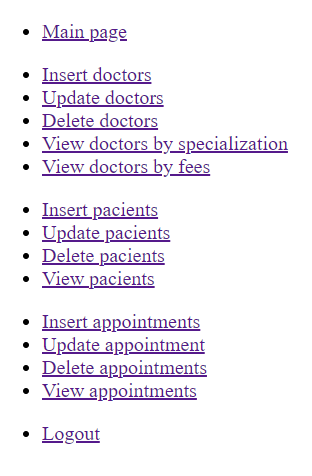
The admin page looks like this:



3. The main pages

3.1 Admin main page

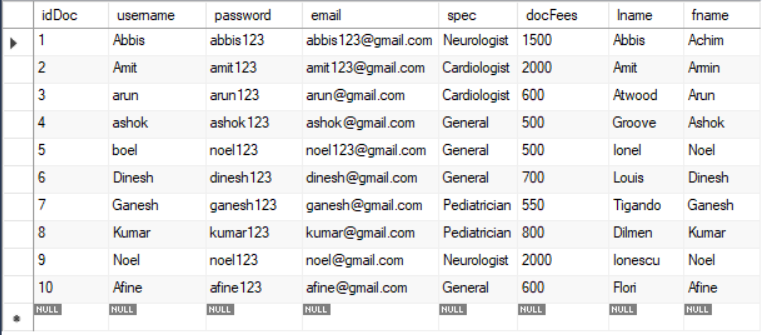
The admin can do the following tasks in the Hospital Management System:



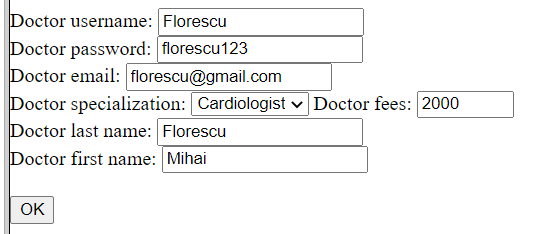
* Can add new doctors, update existing doctors, delete them, or view them by their specialization or by their fees.
* He can also insert new pacients, update their data, delete them or view all of them
* The admin can insert, update, delete or view appointments, as all of this takes place inside his hospital
* The admin CAN NOT work with the prescription table as he is not certified to analyse the health of the pacients

For example, let us consider that the admin wants to add a new doctor.

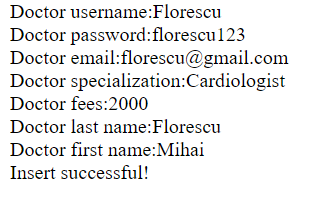
The table looks initially like this:



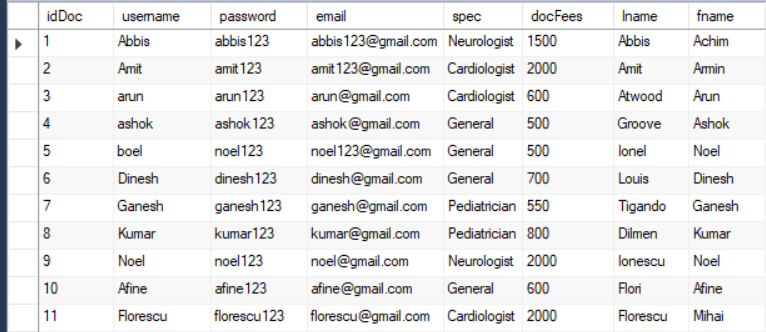
As it can be seen, the last doctor has the id 10. But by adding a new doctor from the admin page:



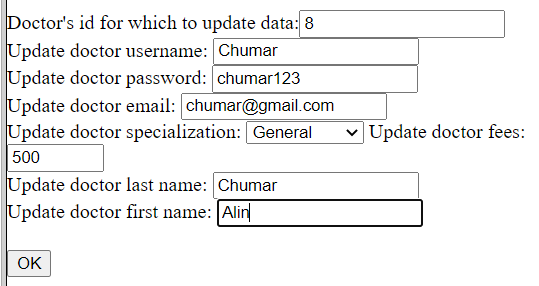
The new doctor has been inserted in the database:



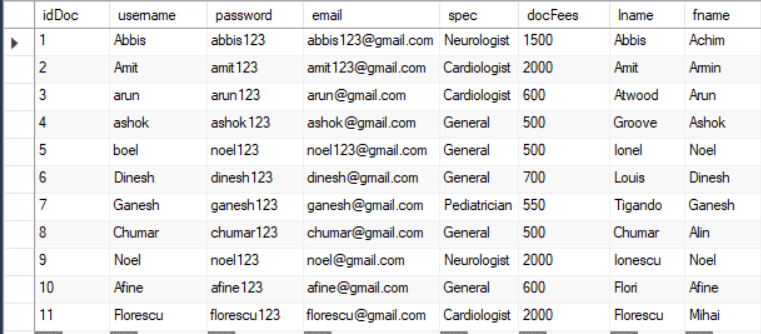
We can observe in the MySQL table the new doctor with the id 11:



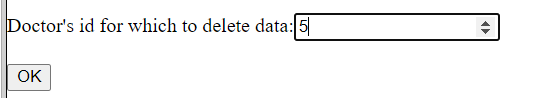
Now the admin wants to update a doctor, let us say he modifies the doctor with id 8.



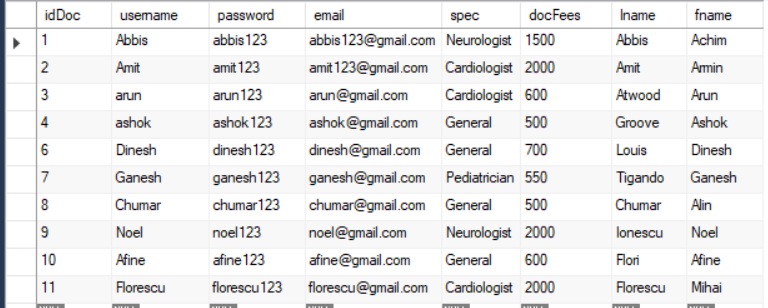
If we look at the table, we can see the changes for the doctor with id 8:



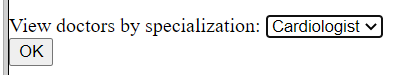
Deleting a doctor, the admin has to enter the id of the specific doctor to delete, let us say the doctor with id 5:



In MySQL the doctor with id 5 has been deleted:



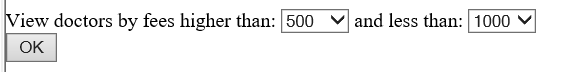
Admin can visualize doctors by their specialization, let’s say he want to see all doctors working as Cardiologists:



And the result will be displayed in the middle:



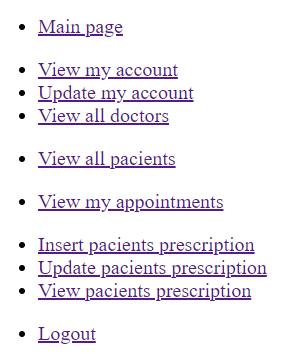
Now if we want to see them by fees, between 500 and 1000:



We can see them displayed and ordered by their fees:



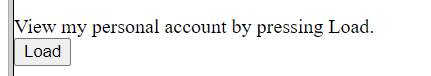
3.2 Doctor main page



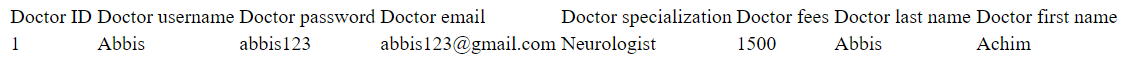
The doctor can do the following:

* He can view only his account data, such as email, password, name etc. and update only his account.
* He can view the other doctors in the hospital by specialization like the admin can.
* He can view all the pacients in the hospital
* He can view the appoinments that were assigned to only him
* He can write prescriptions according to his appointments, update them and even see what he wrote to each pacient.

For example, I logged into dr. Abbis account, by pressing “Load”:



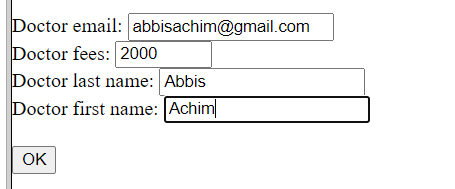
All of his data will be displayed:



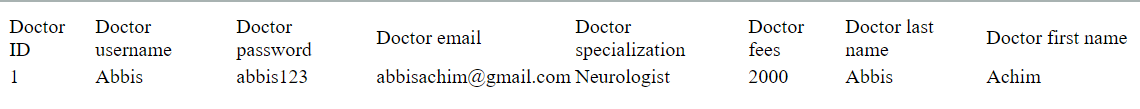
He can see this way his current password, email, ID, the fees that his charging etc.

If he wants to update his account, he can press the „Update account”, there he can only update his email, fees, last name and first name, as he must remain with the same username in the database to be recognized by the pacients. He can not change his specialization as he is certified to do only one domain.

Let’s say he updated his email and fees to:

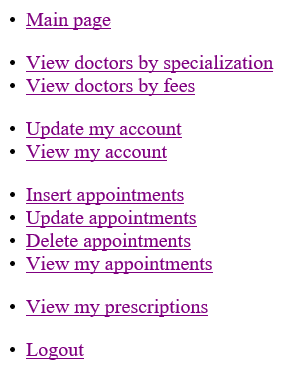


Now, if he looks back to his account, the specified fields have been modified:



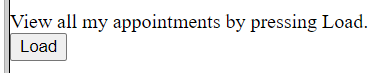
3.3 Pacient main page

The pacient can do the following:



* A pacient can view all the doctors in the hospital by their specialization or by their fees
* Can update or view only his account (the one that he is logged into)
* Can create new appointments or delete his own appointments, he cannot update or delete others appointments as his email that he logged in with is checked and it must correspond with the email that the appointment was made with.

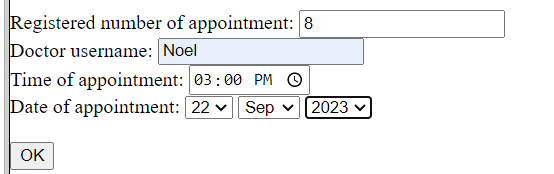
I logged into one of the pacients account, by pressing view appointments:



There will be displayed only her own appointments:



If the user wants to update her appointments, she can modify only the doctor, appointment date and time by introducing the registered number seen in the first column:

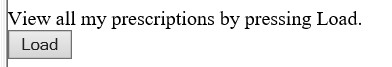


It can be observed that the doctor fees updates based on the doctor that is chosen:



The previous doctor was dr. Ganesh, and the current one for the appointement with registered number 8 is dr. Noel.

If she wants to see her prescriptions:



The prescriptions remain based on the doctor the appointment was made initially, even if the user updated the appointment, if the doctor already prescribed it, it will remain there.

