CIS 344 FINAL PROJECT REPORT

Name: Md Robiul H Chowdhury

Professor Yanilda Peralta Ramos

Title: Hospital's Portal

Introduction: This project is about a Hospital's portal where the administrator can see the patient's name and their unique ID, age, admission date, discharge date, etc. The administrator is also able to add new patients and assign discharge dates, schedule appointments, and view the appointment date. For this project, I used MYSQL and Python. The file I needed to complete this project is a MySQL file, portalServer.py, and portalDatabase.py file. All the files I uploaded to GitHub. The link is at the end.

Description: The first step for this project is to create a database on MYSQL which I later connected to portalDatabase.py using my MYSQL credentials. First, I created a patient table and an appointments table and assigned the foreign keys to patient_id. I realized that I also need a doctor's table since I cannot assign a foreign key for doctor_id without that table. Both tables needed to have a unique primary key, so I did auto_increment which generates a unique primary key. I inserted values on each table and tested if I was able to see my values.

```
create database hospital_portal;

use hospital_portal;

create table patients (
        patient_id int not null unique primary key auto_increment,
        patient_name varchar(45) not null,
        age int not null,
        admission_date date,
        discharge_date date
);
```

```
select * from Appointments;

insert into patients (patient_name, age, admission_date, discharge_date)
        values ("Maria moon",35, "2023-10-15","2023-10-25"),
               ("Will smith",50, "2023-10-09","2023-10-19"),
               ("John cina",47, "2023-09-15","2023-10-11"),
               ("John wick",34, "2023-10-01","2023-11-15");

select * from patients;//
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap

patient_id	patient_name	age	admission_date	discharge_date
1	Maria moon	35	2023-10-15	2023-10-25
2	Will smith	50	2023-10-09	2023-10-19
3	John cina	47	2023-09-15	2023-10-11
4	John wick	34	2023-10-01	2023-11-15
NULL	NULL	NULL	NULL	NULL

I did the same for the doctor's table. However, I didn't insert any values for the appointment table, since I needed to create a stored procedure for appointment scheduling which I can use again and again. In the appointment table, I assigned the

CIS 344 FINAL PROJECT REPORT

Name: Md Robiul H Chowdhury

Professor Yanilda Peralta Ramos

respective foreign keys. One for the doctor_id and one for the patient_id. For the appointment table, I did the following:

```
create table Appointments (
            appointment_id int not null unique primary key auto_increment,
            patient_id int not null,
            doctor_id int not null,
            appointment_date date not null,
            appointment_time decimal(5,2) not null,
            foreign key (patient_id) references patients(patient_id),
            foreign key (doctor_id) references doctors(doctor_id)
);

select * from Appointments;
```

Now that I have created all the required tables for my database, I can create a stored procedure for appointment scheduling which I will again and again add values.

Note: I did not insert any values on the appointment table yet.

At this point, I need to create a stored procedure called "appointment_scheduling" using the following parameters which I will call to add values. I inserted those parameters into the columns and which will update the appointment table.

```
delimiter //
create procedure appointment_scheduling (in p_patient_id int, in d_doctor_id int, in p_appointment_date date,
                                         in p_appointment_time decimal(5,2))

begin
    insert into Appointments (patient_id, doctor_id, appointment_date, appointment_time)
    values (p_patient_id,d_doctor_id, p_appointment_date, p_appointment_time);

end //
delimiter ;
```

I created this procedure and called it to add values that will help to schedule patient's appointments. Here delimiter is useful because I don't want to see conflicts with semicolons, and it also needed to create procedure.

Result Grid | Filter Rows: | Edit: | Export/Im

appointment_id	patient_id	doctor_id	appointment_date	appointment_time
1	3	2	2023-12-11	12.45
2	1	2	2023-12-12	12.45
3	1	2	2023-12-10	11.45
NULL	NULL	NULL	NULL	NULL

As I can see it is working perfectly. I added a couple more just to confirm. As can be seen on the left. To verify that my procedure is working properly, I used the select command to see my appointment table.

CIS 344 FINAL PROJECT REPORT

Name: Md Robiul H Chowdhury

Professor Yanilda Peralta Ramos

Now I created a stored procedure for discharge patients which will allow them to assign a discharge date or update a patient's discharge date to current. I followed a similar procedure to create this and named it "dischargePatient".

```
delimiter //
create procedure dischargePatient (in p_patient_id int)
begin
    update patients set discharge_date = current_date()
    where patient_id = p_patient_id;
end //
delimiter ;

call dischargePatient(3);
```

To check to see if my procedure working I set the date to current date. It can be seen on the left. Here I also needed a dilimeter to create procedures and parameters, so that I discharge a patient to current date just by using the patient's id.

Result Grid | Filter Rows: | Edit:

patient_id	patient_name	age	admission_date	discharge_date
1	Maria moon	35	2023-10-15	2023-10-25
2	Will smith	50	2023-10-09	2023-10-19
3	John cina	47	2023-09-15	2023-12-18

To verify my change I simply checked my patient's table by using the "select" command and it can be left that for patient id no 3, it changed the discharge date.

Since I created a stored procedure for appointments, I need to see that. In this case, I created a view where I can see my desired appointment date, time, etc. The view is a table that is derived from other tables, and it doesn't change anything in the table. It just displays the information.

```
create view ViewAppointments as
  select a.appointment_id, a.appointment_date, a.appointment_time, p.patient_id, d.doctor_id
  from Appointments a
  join patients p on a.patient_id = p.patient_id
  join doctors d on a.doctor_id = d.doctor_id;

select * from  ViewAppointments;
```

Note: Here p. refers to patient table, d. doctor table and a. appointment table.

As can be seen, a view is created. I also needed to join the three tables. It derives information from the three tables. It displays appointment_id, appointment_date, and appointment_time from the appointment table and patient_id from the patient table, and doctor_id from the doctor's table.

CIS 344 FINAL PROJECT REPORT

Name: Md Robiul H Chowdhury

Professor Yanilda Peralta Ramos

Since I'm done with MYSQL database work, now it is time to work on portalDatabase.py and portalServer.py files. For all the work I have done on the MYSQL database, to display I will use the Python server and database to connect it. To do that first I put my credentials port number, host name, user, password, etc., and saved it. There were some errors I was getting and able to fix them. Before I went to portalServer.py and ran the module, I added some functionality to the portalDatabase.py. Here, after successfully connecting MYSQL server and running the module:

```
Python 3.11.7 (tags/v3.11.7:fa7a6f2, Dec  4 2023, 19:24:49) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

===== RESTART: C:\Users\Robi\Desktop\CIS 344\Final_project\portalServer.py =====

Warning (from warnings module):
  File "C:\Users\Robi\Desktop\CIS 344\Final_project\portalServer.py", line 4
    import cgi
DeprecationWarning: 'cgi' is deprecated and slated for removal in Python 3.13
Starting httpd on port 8000
```

Note: The warning! is not an error but notifying that the "cgi" module will likely be removed on python version 3.13.

Hospital's Portal

Home| Add Patient| Schedule Appointment| View Appointments| Discharge Patient| View All Doctors| View All Records

All Patients

Patient ID	Patient Name	Age	Admission Date	Discharge Date
1	Maria moon	35	2023-10-15	2023-10-25
2	Will smith	50	2023-10-09	2023-10-19
3	John cina	47	2023-09-15	2023-10-11
4	John wick	34	2023-10-01	2023-11-15

To see this table, I went to my web browser and typed "localhost:8000" since the assigned port number was 8000.

Here, I can see my patient's table with all the columns I added.

Hospital Portal

Home| Add Patient| Schedule Appointment| View Appointments| Discharge Patient| View All Doctors| View All Records

Patient has been added

Add Another Patient

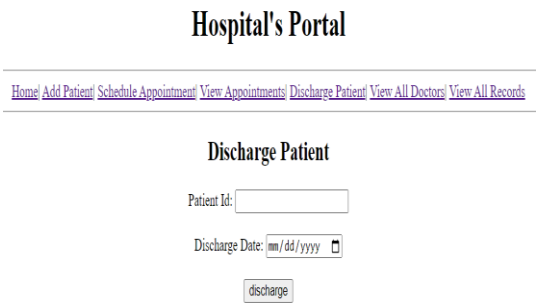
Now if I click on "add patient, I can add a new patient in my table. It can be seen that I can add new patients to my patient's table.

CIS 344 FINAL PROJECT REPORT

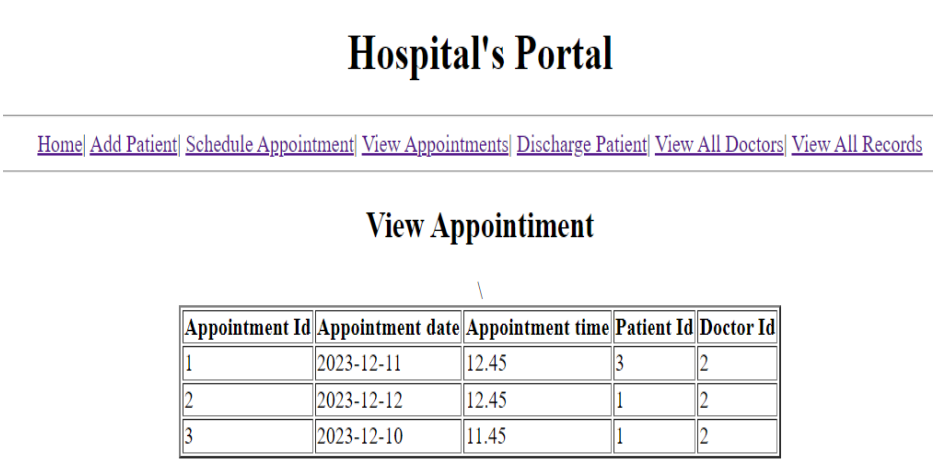
Name: Md Robiul H Chowdhury

Professor Yanilda Peralta Ramos

There are other functionalities that I also added to my hospital's portal.



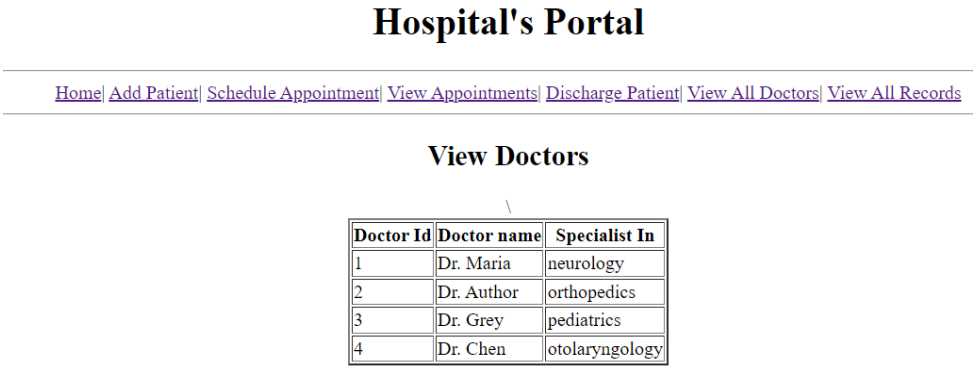
I also created a view to see the patient's appointment in MYSQL which I connected to my Python database. When I click view appointment, I can see the following:



Appointment Id	Appointment date	Appointment time	Patient Id	Doctor Id
1	2023-12-11	12.45	3	2
2	2023-12-12	12.45	1	2
3	2023-12-10	11.45	1	2

It can be seen that I added a few of appointment schedules for my patients. I kept testing and testing just to my sure that my view appointment was working.

I also added another functionality in my Python database to see my doctor's table.



Doctor Id	Doctor name	Specialist In
1	Dr. Maria	neurology
2	Dr. Author	orthopedics
3	Dr. Grey	pediatrics
4	Dr. Chen	otolaryngology

As can be seen, I can see my doctor's table with their name, their unique id, and their specialty.

CIS 344 FINAL PROJECT REPORT

Name: Md Robiul H Chowdhury

Professor Yanilda Peralta Ramos

Lastly, I also added another view where I can see all records for all the tables. I created the view named "ViewAllrecords" and added functionality in the Python database and server, so it displays. I added the result in the following:

```
create view ViewAllrecords as
  select a.appointment_id, a.appointment_date, a.appointment_time,p.patient_id,p.patient_name,p.age,
  p.admission_date, p.discharge_date,d.doctor_id, d.doctor_name,d.specialist_in
  from Appointments a
  join patients p on a.patient_id = p.patient_id
  join doctors d on a.doctor_id = d.doctor_id;
```

```
select * from ViewAllrecords;
```

Here I created a view for all three tables in MYSQL.

Hospital's Portal

Home| Add Patient| Schedule Appointment| View Appointments| Discharge Patient| View All Doctors| View All Records

View All Records

Doctor Appoinment Id	Doctor Appoinment Date	Doctor Appointment Time	Patient Id	Patient Name	Patient Age	Admisssion Date	Discharge Date	Doctor Id	Doctor Name	Doctor Specialist In
1	2023-12-11	12.45	3	John cina	47	2023-09-15	2023-10-11	2	Dr. Author	orthopedics
2	2023-12-12	12.45	1	Maria moon	35	2023-10-15	2023-10-25	2	Dr. Author	orthopedics
3	2023-12-10	11.45	1	Maria moon	35	2023-10-15	2023-10-25	2	Dr. Author	orthopedics

Here it displays the whole record.

Pros and cons: There are many problems I faced during this project. Since there were many errors in the starter code, first I had to fix those errors. However, when I was able to fix those errors, I started enjoying the project and was able to complete this project to the best of my ability.

Program: MYSQL workbench 8.0 CE, IDLE (Python 3.11).

Conclusion: The hospital's portal project wasn't an easy task. However, I put my maximum ability into getting this project done and completing it, which gave me some new knowledge. After completing this project, I uploaded my project to GitHub.

GitHub link:
https://github.com/robichy98/Md_Robiul_Chowdhury_CIS_344