MySQL Part:

After creating the hospital_portal schema, I first developed the patients table, with the patient id as the primary key, it has the unique attribute so IDs can't be repeated, and auto_increment to automatically assign ID values, the other columns are the patient name, age, the date of admission, and the date of discharge.

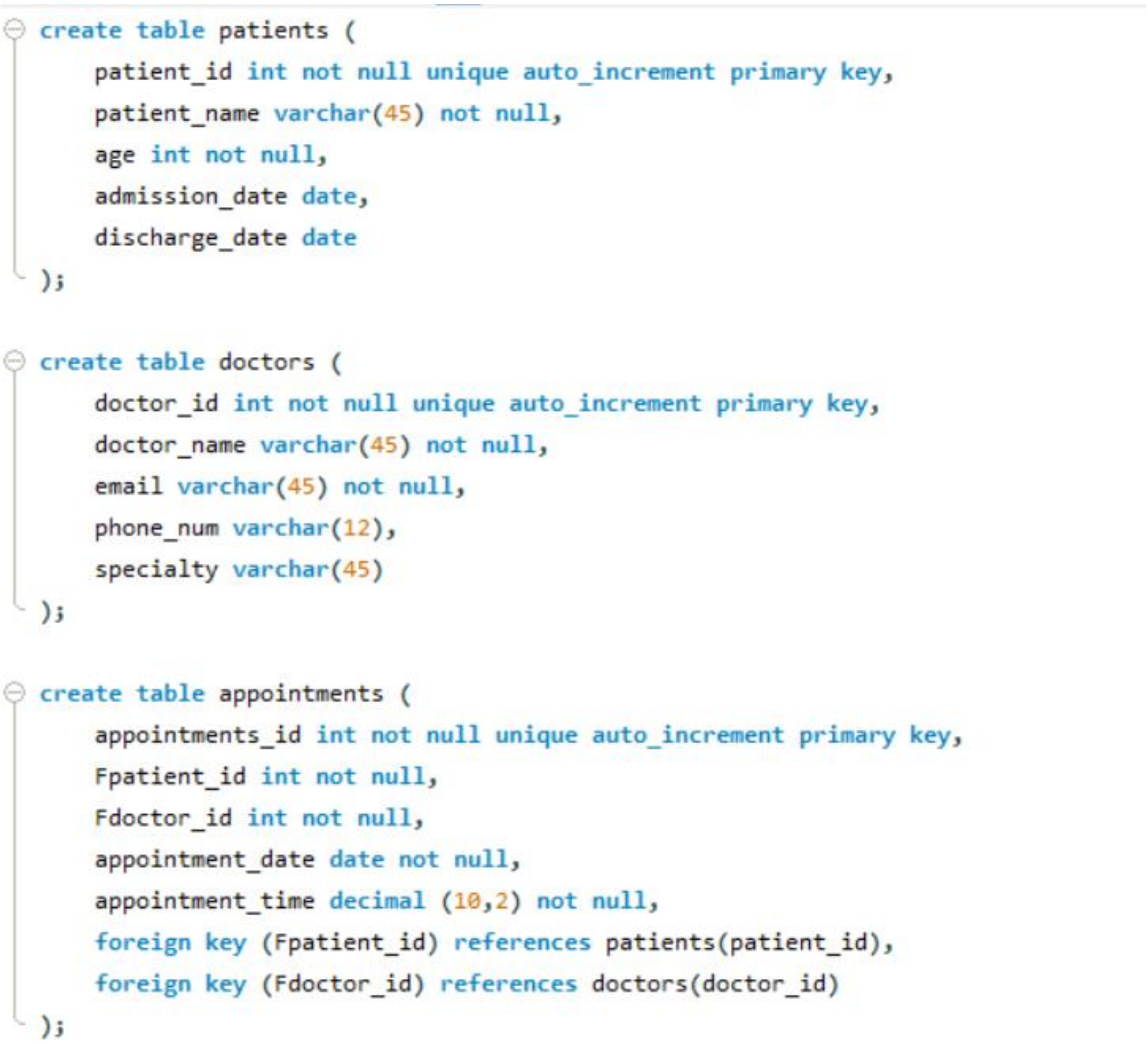
Afterwards, I developed the doctors table (This had to be done first as the appointment tables requires a doctor's id as a foreign key), and after defining the ID and name columns, I added the doctor's email, phone number, and specialty.

Next up was the appointments table, the most notable thing regarding this one is the appointment time being written in decimal, I decided to allow up to 4 digits (last two are after the decimal dot) for 24h clock authenticity, and renamed the foreign key columns (patient_id and doctor_id into Fpatient_id and Fdoctor_id, F for foreign) to circumvent an issue creating the join view later on.

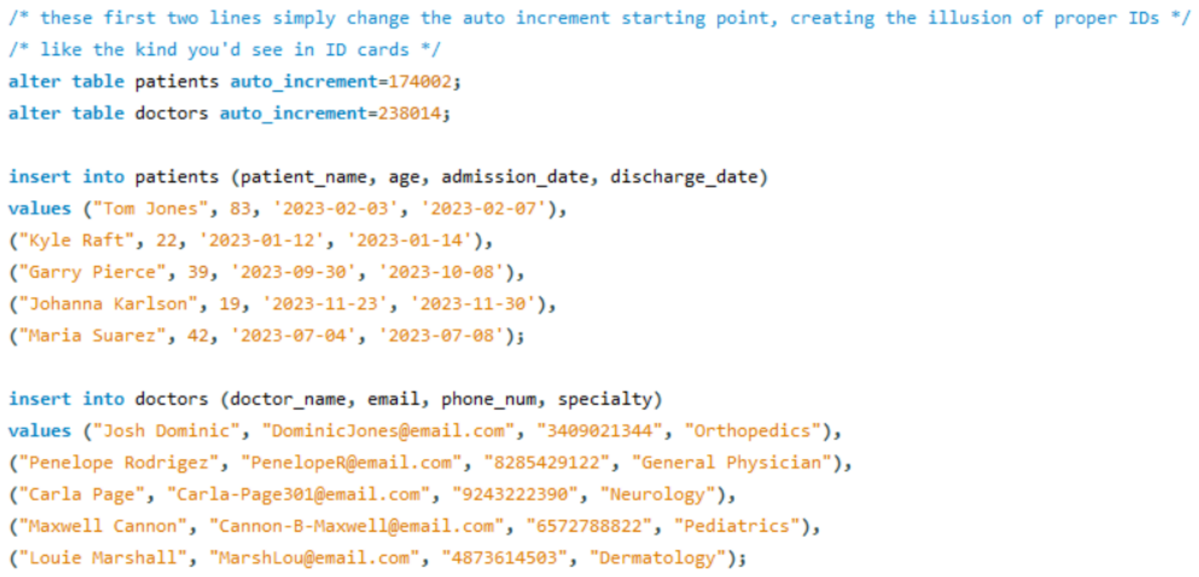
With the tables created, I proceeded to work on the values, five patients and five doctors, each with unique 6-digit IDs, then it was time for the procedures, starting with AppointmentSchedule, which takes the input for both patient and doctor's ID, alongside the appointment date and time, and then updates the admission date of the corresponding patient to the set date.

Next comes dischargePatient, which only needs the input of the ID of the patient we want to discharge, and removes the appointment from the appointments table, finally updating the discharge date to the date the command is run on.

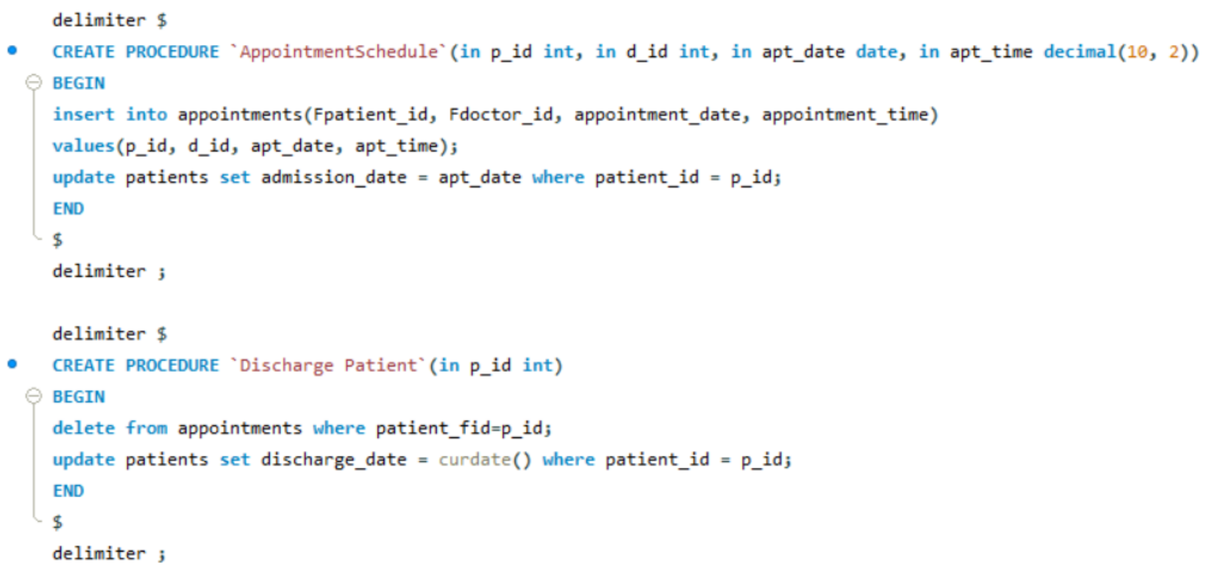
And finishing with the MySQL portion the last piece needed was the view of the join of all three tables, the view selects all the data from appointments and then sequentially uses a left join for the other two tables based on the IDs, this is where renaming the patient and doctor IDs in the appointments table came in handy, as it prevented duplicate IDs from causing an error when running the code.



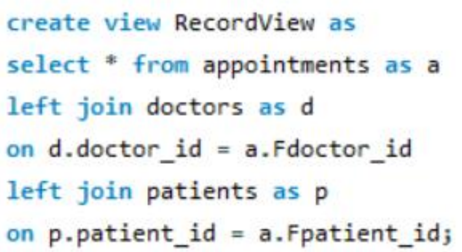
```
create table patients (
    patient_id int not null unique auto_increment primary key,
    patient_name varchar(45) not null,
    age int not null,
    admission_date date,
    discharge_date date
);

create table doctors (
    doctor_id int not null unique auto_increment primary key,
    doctor_name varchar(45) not null,
    email varchar(45) not null,
    phone_num varchar(12),
    specialty varchar(45)
);

create table appointments (
    appointments_id int not null unique auto_increment primary key,
    Fpatient_id int not null,
    Fdoctor_id int not null,
    appointment_date date not null,
    appointment_time decimal (10,2) not null,
    foreign key (Fpatient_id) references patients(patient_id),
    foreign key (Fdoctor_id) references doctors(doctor_id)
);
```



```sql
/* these first two lines simply change the auto increment starting point, creating the illusion of proper IDs */
/* like the kind you'd see in ID cards */
alter table patients auto_increment=174002;
alter table doctors auto_increment=238014;

insert into patients (patient_name, age, admission_date, discharge_date)
values ("Tom Jones", 83, '2023-02-03', '2023-02-07'),
("Kyle Raft", 22, '2023-01-12', '2023-01-14'),
("Garry Pierce", 39, '2023-09-30', '2023-10-08'),
("Johanna Karlson", 19, '2023-11-23', '2023-11-30'),
("Maria Suarez", 42, '2023-07-04', '2023-07-08');

insert into doctors (doctor_name, email, phone_num, specialty)
values ("Josh Dominic", "DominicJones@email.com", "3409021344", "Orthopedics"),
("Penelope Rodrigez", "PenelopeR@email.com", "8285429122", "General Physician"),
("Carla Page", "Carla-Page301@email.com", "9243222390", "Neurology"),
("Maxwell Cannon", "Cannon-B-Maxwell@email.com", "6572788822", "Pediatrics"),
("Louie Marshall", "MarshLou@email.com", "4873614503", "Dermatology");
```



```sql
delimiter $
CREATE PROCEDURE `AppointmentSchedule`(in p_id int, in d_id int, in apt_date date, in apt_time decimal(10, 2))
BEGIN
insert into appointments(Fpatient_id, Fdoctor_id, appointment_date, appointment_time)
values(p_id, d_id, apt_date, apt_time);
update patients set admission_date = apt_date where patient_id = p_id;
END
$
delimiter ;

delimiter $
CREATE PROCEDURE `Discharge Patient`(in p_id int)
BEGIN
delete from appointments where patient_fid=p_id;
update patients set discharge_date = curdate() where patient_id = p_id;
END
$
delimiter ;
```

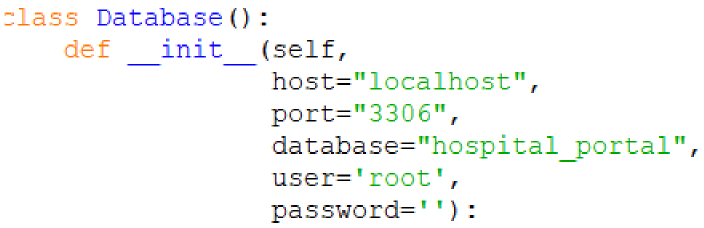


```sql
create view RecordView as
select * from appointments as a
left join doctors as d
on d.doctor_id = a.Fdoctor_id
left join patients as p
on p.patient_id = a.Fpatient_id;
```

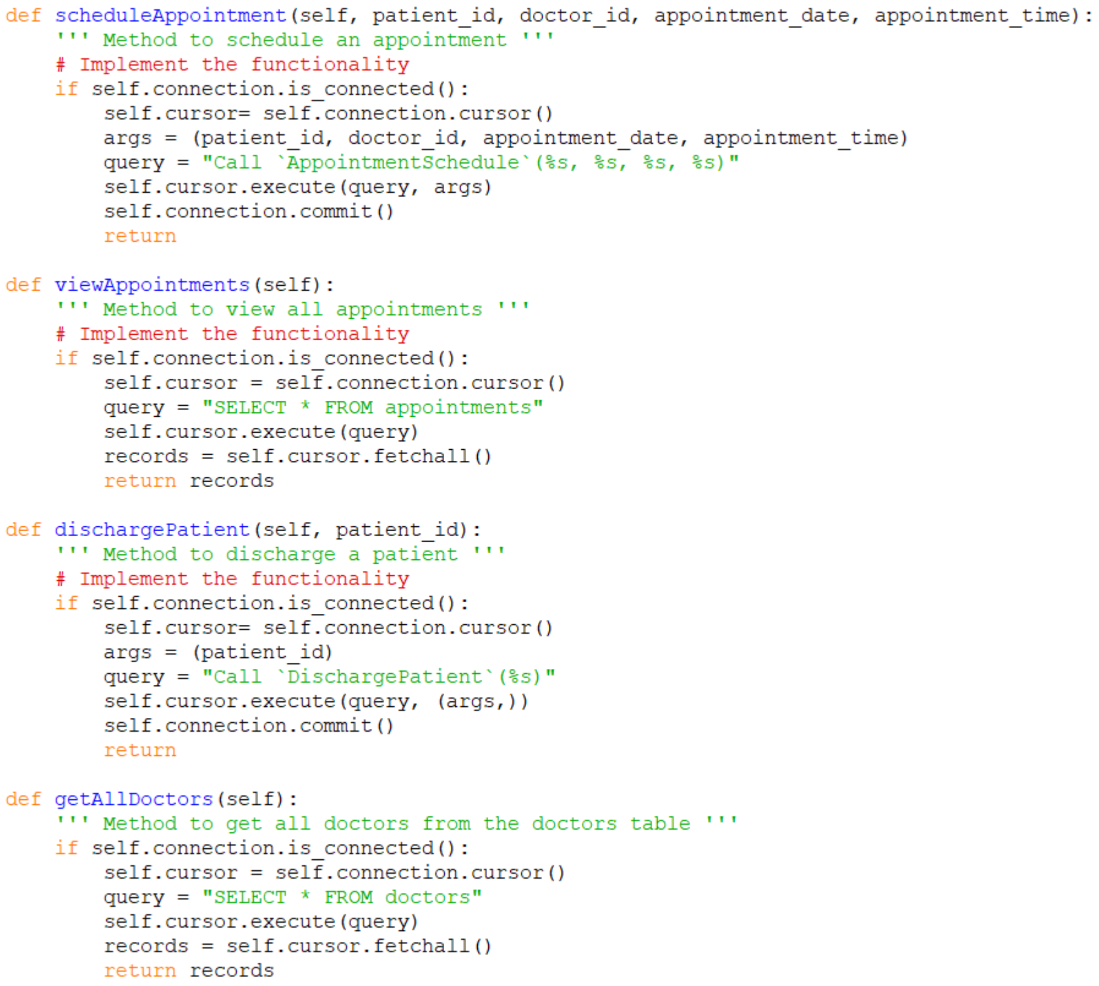
Python Part:

Now that the Database is developed, it was time to modify the starter code to allow it to connect to, interface, and output our database into interactable HTML code, starting with portalDatabase.py, changed the port to the one used by my local database, and added my password, afterwards I added the functionality code to scheduleAppointment, viewAppointments, and dischargePatient, taking queues from addPatient and getAllPatients, but using the proper lines to account for the stored procedures being called (using a variable to store the input data for the procedure, and then executing it alongside the query), I then added a getAllDoctors method (This helps to get the IDs to schedule appointments)

Then it was time for portalServer.py which took a bit more work as it had the code for the HTML page, starting by completing the code to add patients by calling the method to add patients from portalDatabase, then it was free reign to complete the pages for the unimplemented functions (schedule appointments, view appointments, get all doctors, discharge patients) view appointments used the code to view all patients as a basis as it simply was a matter of displaying the data already stored in the database, same thing for get all doctors, for scheduling appointments and discharging patients, I implemented forms, scheduling only required numbers but it was important to ensure that appointment time needed to accept decimal values as that was the format in the database, I also had to add another do_post definition to the code, otherwise the data would go nowhere, once scheduling was functional, discharge was almost identical, only requiring a single input value for the ID of the patient being discharged, finally I added code to recognize the newly created page for the get all doctors function, one last thing of note is that the starter code was missing an except for the do_get definition’s try.

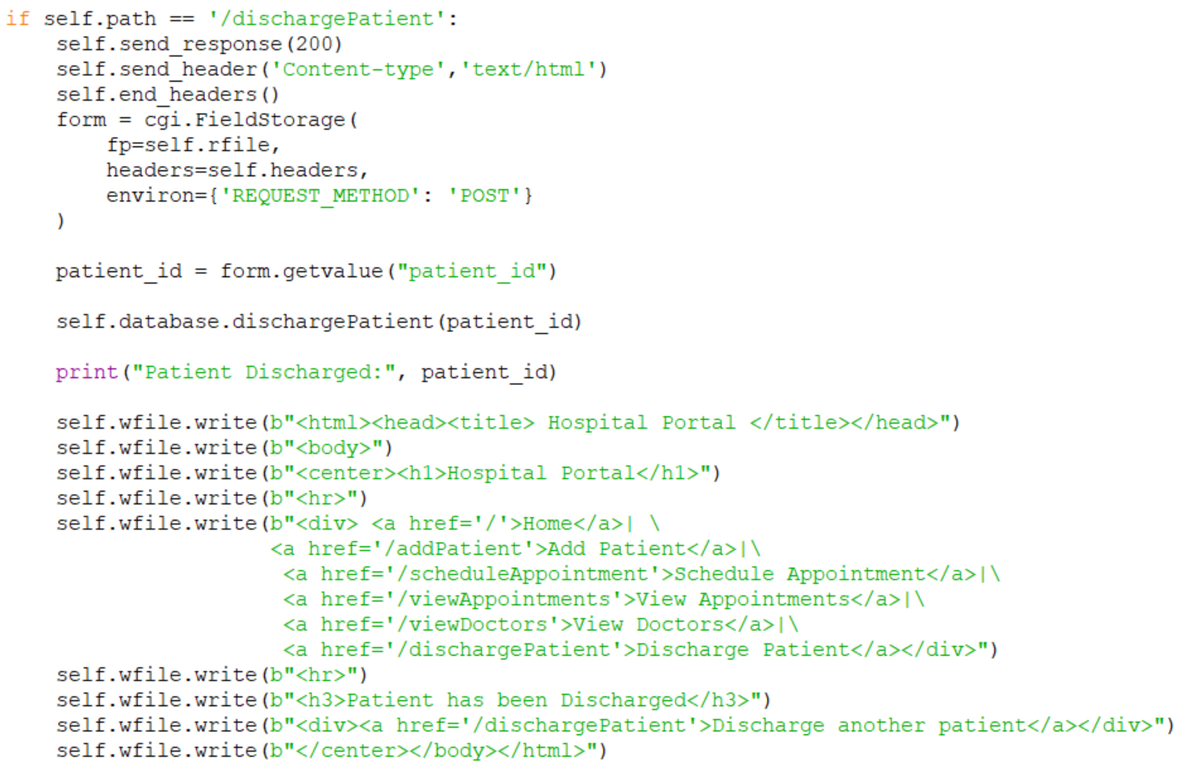


```
class Database():
    def __init__(self,
                 host="localhost",
                 port="3306",
                 database="hospital_portal",
                 user='root',
                 password=''):
```

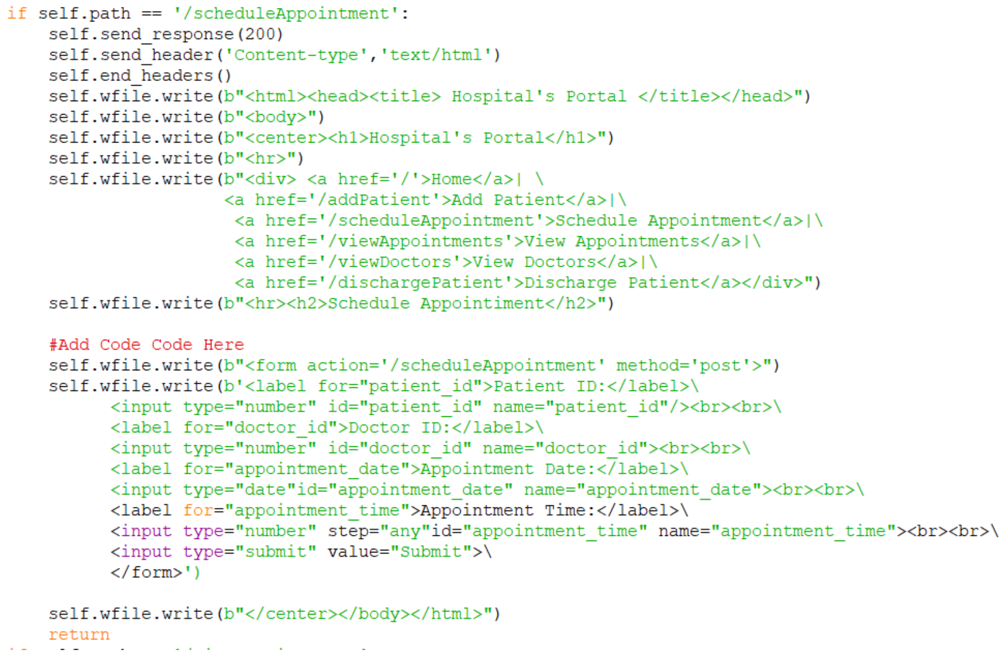


```
    def scheduleAppointment(self, patient_id, doctor_id, appointment_date, appointment_time):
        ''' Method to schedule an appointment '''
        # Implement the functionality
        if self.connection.is_connected():
            self.cursor= self.connection.cursor()
            args = (patient_id, doctor_id, appointment_date, appointment_time)
            query = "Call `AppointmentSchedule`(%s, %s, %s, %s)"
            self.cursor.execute(query, args)
            self.connection.commit()
            return

    def viewAppointments(self):
        ''' Method to view all appointments '''
        # Implement the functionality
        if self.connection.is_connected():
            self.cursor = self.connection.cursor()
            query = "SELECT * FROM appointments"
            self.cursor.execute(query)
            records = self.cursor.fetchall()
            return records

    def dischargePatient(self, patient_id):
        ''' Method to discharge a patient '''
        # Implement the functionality
        if self.connection.is_connected():
            self.cursor= self.connection.cursor()
            args = (patient_id)
            query = "Call `DischargePatient`(%s)"
            self.cursor.execute(query, (args,))
            self.connection.commit()
            return

    def getAllDoctors(self):
        ''' Method to get all doctors from the doctors table '''
        if self.connection.is_connected():
            self.cursor = self.connection.cursor()
            query = "SELECT * FROM doctors"
            self.cursor.execute(query)
            records = self.cursor.fetchall()
            return records
```

```
discharge_date = form.getvalue("discharge_date")
# Call the Database Method to add a new patient. Attention please read this comment!!! See Example call!
'''
    Example call: self.database.addPatient(patient_name, age, admission_date,discharge_date)
'''
self.database.addPatient(patient_name, age, admission_date, discharge_date)

print("Patient added:", patient_name, patient_id, age, admission_date)




   if self.path == '/scheduleAppointment':
       self.send_response(200)
       self.send_header('Content-type','text/html')
       self.end_headers()
       form = cgi.FieldStorage(
           fp=self.rfile,
           headers=self.headers,
           environ={'REQUEST_METHOD': 'POST'}
       )

       patient_id = form.getvalue("patient_id")
       doctor_id = form.getvalue("doctor_id")
       appointment_date = form.getvalue("appointment_date")
       appointment_time = float(form.getvalue("appointment_time"))

       self.database.scheduleAppointment(patient_id, doctor_id, appointment_date, appointment_time)

       print("Appointment Scheduled:", patient_id, doctor_id, appointment_date, appointment_time)

       self.wfile.write(b"<html><head><title> Hospital Portal </title></head>")
       self.wfile.write(b"<body>")
       self.wfile.write(b"<center><h1>Hospital Portal</h1>")
       self.wfile.write(b"<hr>")
       self.wfile.write(b"<div> <a href='/'>Home</a>| \
                        <a href='/addPatient'>Add Patient</a>|\
                         <a href='/scheduleAppointment'>Schedule Appointment</a>|\
                         <a href='/viewAppointments'>View Appointments</a>|\
                         <a href='/viewDoctors'>View Doctors</a>|\
                         <a href='/dischargePatient'>Discharge Patient</a></div>")
       self.wfile.write(b"<hr>")
       self.wfile.write(b"<h3>Appointment has been Scheduled</h3>")
       self.wfile.write(b"<div><a href='/scheduleAppointment'>Schedule Another Appointment</a></div>")
       self.wfile.write(b"</center></body></html>")
```



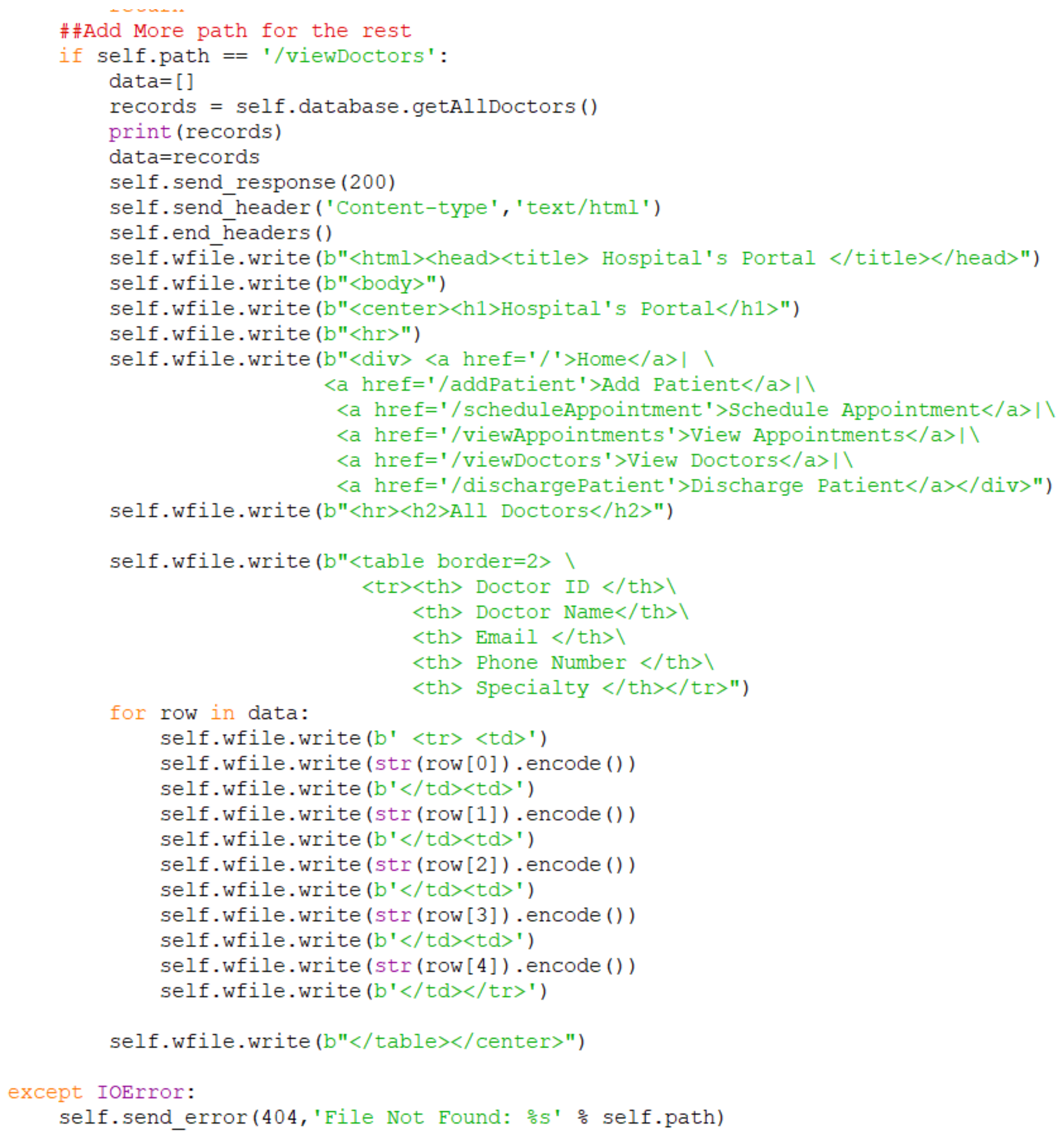
```
if self.path == '/dischargePatient':
    self.send_response(200)
    self.send_header('Content-type','text/html')
    self.end_headers()
    form = cgi.FieldStorage(
        fp=self.rfile,
        headers=self.headers,
        environ={'REQUEST_METHOD': 'POST'}
    )

    patient_id = form.getvalue("patient_id")

    self.database.dischargePatient(patient_id)

    print("Patient Discharged:", patient_id)

    self.wfile.write(b"<html><head><title> Hospital Portal </title></head>")
    self.wfile.write(b"<body>")
    self.wfile.write(b"<center><h1>Hospital Portal</h1>")
    self.wfile.write(b"<hr>")
    self.wfile.write(b"<div> <a href='/'>Home</a>| \
                     <a href='/addPatient'>Add Patient</a>|\
                      <a href='/scheduleAppointment'>Schedule Appointment</a>|\
                      <a href='/viewAppointments'>View Appointments</a>|\
                      <a href='/viewDoctors'>View Doctors</a>|\
                      <a href='/dischargePatient'>Discharge Patient</a></div>")
    self.wfile.write(b"<hr>")
    self.wfile.write(b"<h3>Patient has been Discharged</h3>")
    self.wfile.write(b"<div><a href='/dischargePatient'>Discharge another patient</a></div>")
    self.wfile.write(b"</center></body></html>")
```



```
if self.path == '/scheduleAppointment':
    self.send_response(200)
    self.send_header('Content-type','text/html')
    self.end_headers()
    self.wfile.write(b"<html><head><title> Hospital's Portal </title></head>")
    self.wfile.write(b"<body>")
    self.wfile.write(b"<center><h1>Hospital's Portal</h1>")
    self.wfile.write(b"<hr>")
    self.wfile.write(b"<div> <a href='/'>Home</a>| \
                     <a href='/addPatient'>Add Patient</a>|\
                      <a href='/scheduleAppointment'>Schedule Appointment</a>|\
                      <a href='/viewAppointments'>View Appointments</a>|\
                      <a href='/viewDoctors'>View Doctors</a>|\
                      <a href='/dischargePatient'>Discharge Patient</a></div>")
    self.wfile.write(b"<hr><h2>Schedule Appointiment</h2>")

    #Add Code Code Here
    self.wfile.write(b"<form action='/scheduleAppointment' method='post'>")
    self.wfile.write(b'<label for="patient_id">Patient ID:</label>\
          <input type="number" id="patient_id" name="patient_id"/><br><br>\
          <label for="doctor_id">Doctor ID:</label>\
          <input type="number" id="doctor_id" name="doctor_id"><br><br>\
          <label for="appointment_date">Appointment Date:</label>\
          <input type="date"id="appointment_date" name="appointment_date"><br><br>\
          <label for="appointment_time">Appointment Time:</label>\
          <input type="number" step="any"id="appointment_time" name="appointment_time"><br><br>\
          <input type="submit" value="Submit">\
          </form>')

    self.wfile.write(b"</center></body></html>")
    return
```

```
if self.path == '/viewAppointments':
    data=[]
    records = self.database.viewAppointments()
    print(records)
    data=records
    self.send_response(200)
    self.send_header('Content-type','text/html')
    self.end_headers()
    self.wfile.write(b"<html><head><title> Hospital's Portal </title></head>")
    self.wfile.write(b"<body>")
    self.wfile.write(b"<center><h1>Hospital's Portal</h1>")
    self.wfile.write(b"<hr>")
    self.wfile.write(b"<div> <a href='/'>Home</a>| \
                     <a href='/addPatient'>Add Patient</a>|\
                      <a href='/scheduleAppointment'>Schedule Appointment</a>|\
                      <a href='/viewAppointments'>View Appointments</a>|\
                      <a href='/viewDoctors'>View Doctors</a>|\
                      <a href='/dischargePatient'>Discharge Patient</a></div>")

    #Add Code Code Here
    self.wfile.write(b"<hr><h2>Appointments</h2>")
    self.wfile.write(b"<table border=2> \
                        <tr><th> Appointment ID </th>\
                            <th> Patient ID</th>\
                            <th> Doctor ID</th>\
                            <th> Appointment Date </th>\
                            <th> Appointment Time </th></tr>")
    for row in data:
        self.wfile.write(b' <tr> <td>')
        self.wfile.write(str(row[0]).encode())
        self.wfile.write(b'</td><td>')
        self.wfile.write(str(row[1]).encode())
        self.wfile.write(b'</td><td>')
        self.wfile.write(str(row[2]).encode())
        self.wfile.write(b'</td><td>')
        self.wfile.write(str(row[3]).encode())
        self.wfile.write(b'</td><td>')
        self.wfile.write(str(row[4]).encode())
        self.wfile.write(b'</td></tr>')

    self.wfile.write(b"</table></center>")

    self.wfile.write(b"</center></body></html>")
    return
```

```
if self.path == '/dischargePatient':
    self.send_response(200)
    self.send_header('Content-type','text/html')
    self.end_headers()
    self.wfile.write(b"<html><head><title> Hospital's Portal </title></head>")
    self.wfile.write(b"<body>")
    self.wfile.write(b"<center><h1>Hospital's Portal</h1>")
    self.wfile.write(b"<hr>")
    self.wfile.write(b"<div> <a href='/'>Home</a>| \
                     <a href='/addPatient'>Add Patient</a>|\
                      <a href='/scheduleAppointment'>Schedule Appointment</a>|\
                      <a href='/viewAppointments'>View Appointments</a>|\
                      <a href='/viewDoctors'>View Doctors</a>|\
                      <a href='/dischargePatient'>Discharge Patient</a></div>")
    self.wfile.write(b"<hr><h2>Discharge Patient</h2>")

    #Add Code Code Here
    self.wfile.write(b"<form action='/dischargePatient' method='post'>")
    self.wfile.write(b'<label for="patient_id">Patient ID:</label>\
          <input type="number" id="patient_id" name="patient_id"/><br><br>\
          <input type="submit" value="Submit">\
          </form>')
    self.wfile.write(b"</center></body></html>")
    return
```



```
        return
    ##Add More path for the rest
    if self.path == '/viewDoctors':
        data=[]
        records = self.database.getAllDoctors()
        print(records)
        data=records
        self.send_response(200)
        self.send_header('Content-type','text/html')
        self.end_headers()
        self.wfile.write(b"<html><head><title> Hospital's Portal </title></head>")
        self.wfile.write(b"<body>")
        self.wfile.write(b"<center><h1>Hospital's Portal</h1>")
        self.wfile.write(b"<hr>")
        self.wfile.write(b"<div> <a href='/'>Home</a>| \
                         <a href='/addPatient'>Add Patient</a>|\
                          <a href='/scheduleAppointment'>Schedule Appointment</a>|\
                          <a href='/viewAppointments'>View Appointments</a>|\
                          <a href='/viewDoctors'>View Doctors</a>|\
                          <a href='/dischargePatient'>Discharge Patient</a></div>")
        self.wfile.write(b"<hr><h2>All Doctors</h2>")

        self.wfile.write(b"<table border=2> \
                            <tr><th> Doctor ID </th>\
                                <th> Doctor Name</th>\
                                <th> Email </th>\
                                <th> Phone Number </th>\
                                <th> Specialty </th></tr>")
        for row in data:
            self.wfile.write(b' <tr> <td>')
            self.wfile.write(str(row[0]).encode())
            self.wfile.write(b'</td><td>')
            self.wfile.write(str(row[1]).encode())
            self.wfile.write(b'</td><td>')
            self.wfile.write(str(row[2]).encode())
            self.wfile.write(b'</td><td>')
            self.wfile.write(str(row[3]).encode())
            self.wfile.write(b'</td><td>')
            self.wfile.write(str(row[4]).encode())
            self.wfile.write(b'</td></tr>')

        self.wfile.write(b"</table></center>")

except IOError:
    self.send_error(404,'File Not Found: %s' % self.path)
```