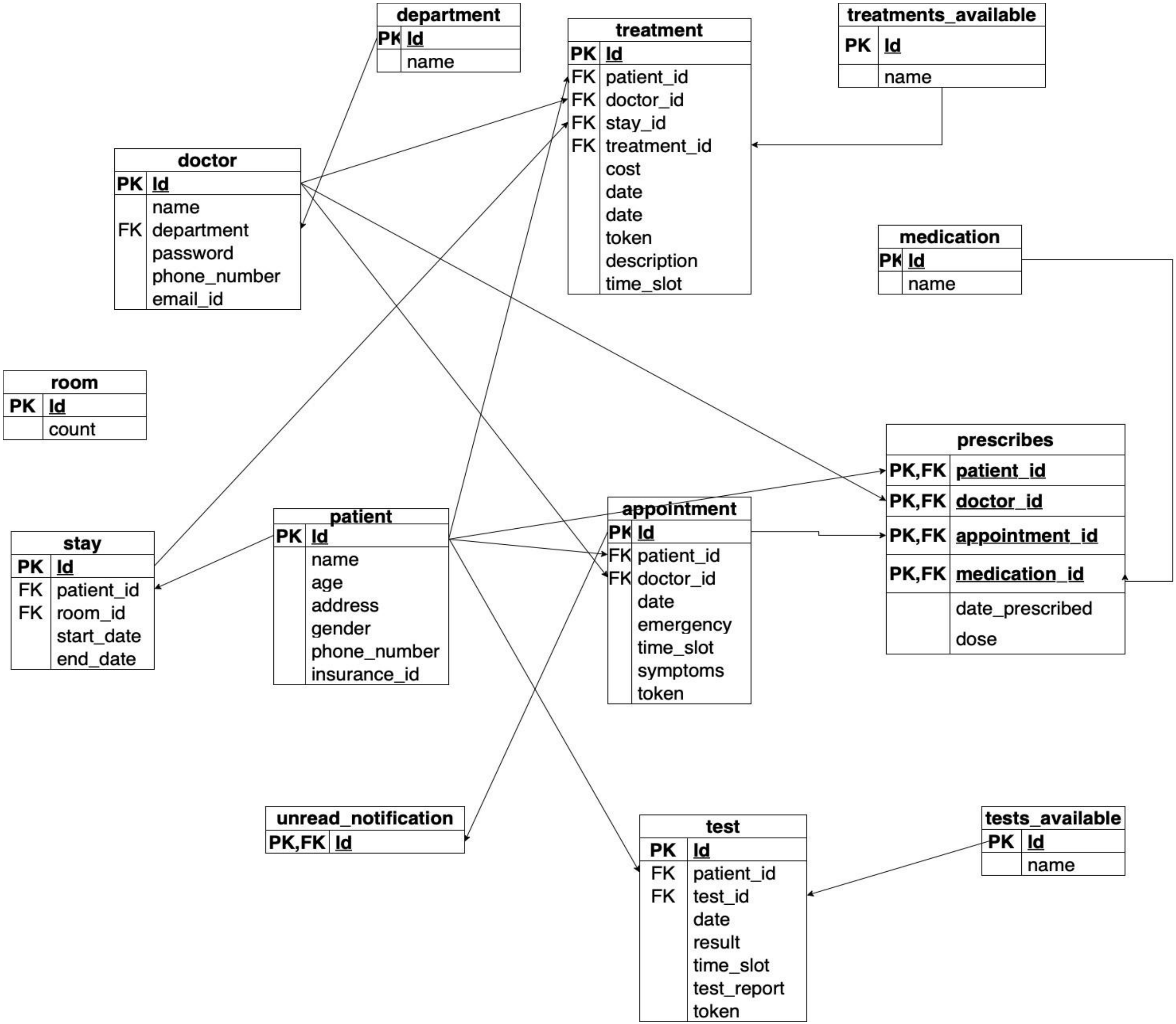
Hospital Management System

Languages and tools used: Python,HTML,CSS,flask framework

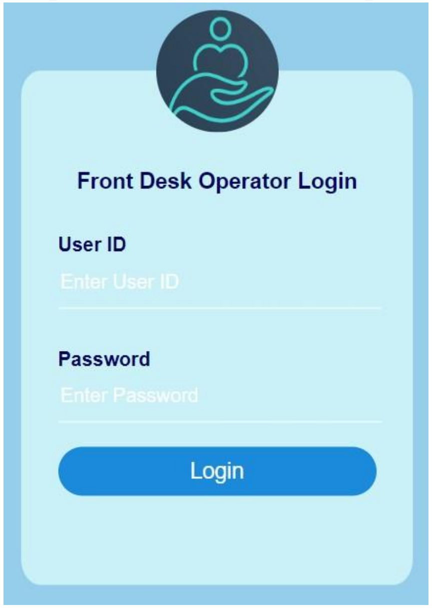
ER Diagram:



Front Desk Operator login page:

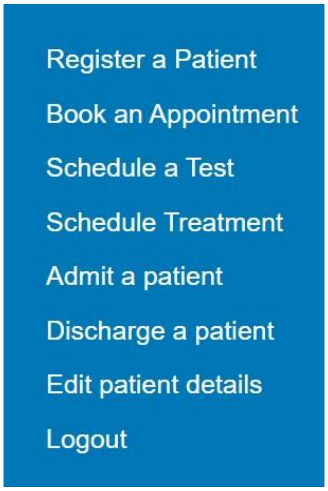
If the front desk operator enters an incorrect userid or password and clicks on login then it shows an error message that “you have entered an incorrect userid or password. Please try again!” and the front desk Operator stays on the same page.

"SELECT * FROM frontdeskop WHERE id = %s AND password = %s", (id, password)



Front Desk Operator page:

Side Nav

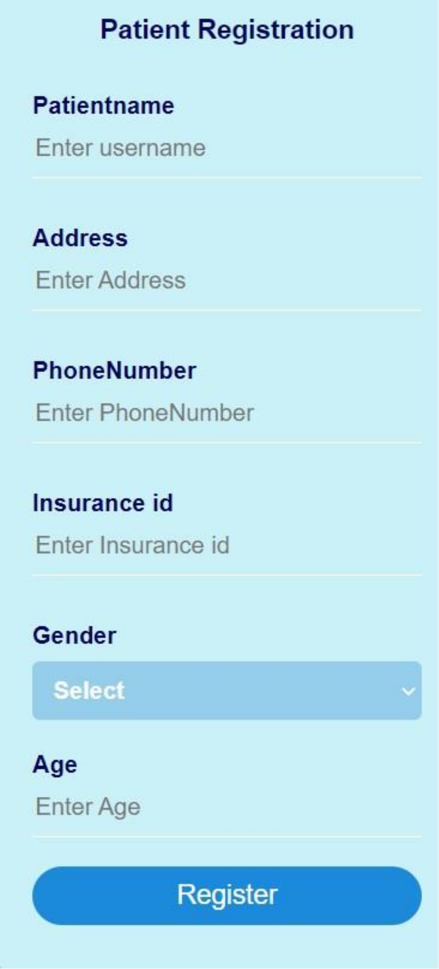


Register a Patient:

Phone Number : Accepts only if it has 10 characters and should lie in between 0 and 9. Gender Dropdown having options : F, M, others.

All the fields mentioned below are required. If the front desk operator clicks on the register, then Patient will be given an id and details will be added to the patient table, the message “successfully registered!” will be displayed and stay on the same page.

“INSERT INTO patient(name,age,address,gender,phone_number,insurance_id) VALUES(%s,%s,%s,%s,%s,%s)”, (name,age,address,gender,phone_number,insurance_id)



Book an appointment page:

Emergency Dropdown having options : 0 or 1
Specialization Drop down having specializations available in the hospital
Time Slot dropdown have 3 options namely “9-12”, “2-5”, “6-10”
Doctor drop down shows available doctors according to the chosen specialization ,chosen date, chosen timeslot.

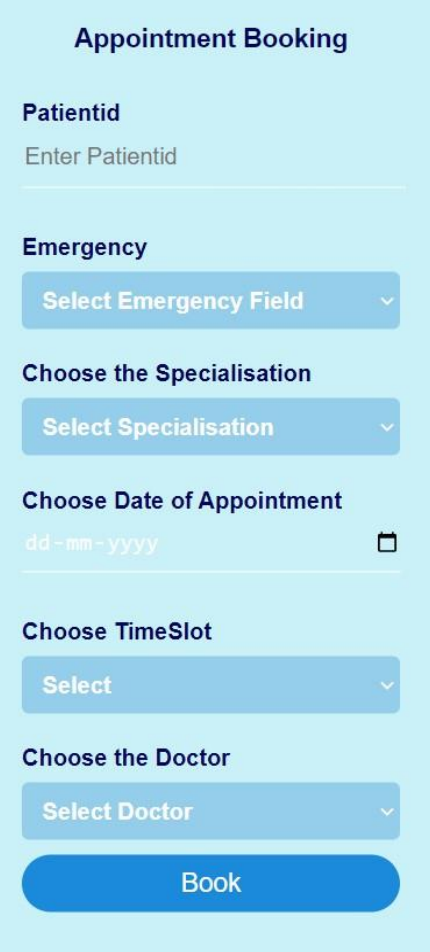
If emergency is 0 then the front desk operator enters all the fields and books an appointment by adding fields to the appointment table. If emergency is 1 then the front desk operator takes only patient id and books an appointment by adding fields to the appointment table.

If emergency == 1:

“INSERT INTO appointment(patient_id,doctor_id,date,emergency) VALUES(%s,%s,NOW(),%s,%s)”, (patient_id,doctor_id,emergency)

else:

“INSERT INTO appointment(patient_id,doctor_id,date,time_slot,emergency,token) VALUES(%s,%s,%s,%s,%s,%s)”, (patient_id,doctor_id,date,time_slot,emergency,token)



Appointment Booking

Patientid

Enter Patientid

Emergency

Select Emergency Field

Choose the Specialisation

Select Specialisation

Choose Date of Appointment

dd-mm-yyyy

Choose TimeSlot

Select

Choose the Doctor

Select Doctor

Book

Test-Scheduling Page:

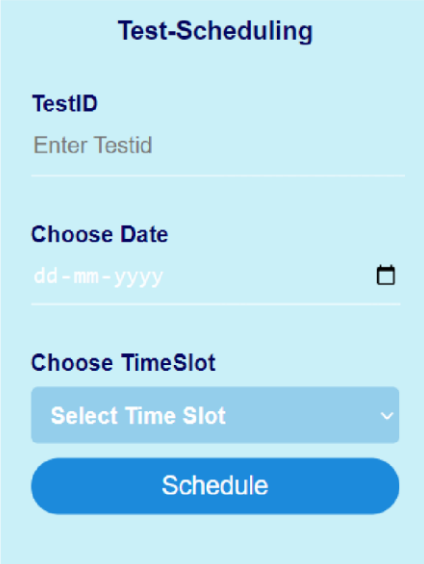
Time Slot dropdown have 3 options namely “9-12”, “2-5”, “6-10”. Adds the fields to the test table on clicking Schedule.

“UPDATE test SET date = %s, time_slot = %s, token = %s WHERE id = %s” ,(Date,timeSlot,token,Test_id)

Treatment-Scheduling Page:

Doctor drop down shows available doctors according to the chosen specialization ,chosen date, chosen timeslot. Adds the fields to the treatment table on clicking Schedule.

“UPDATE treatment SET doctor_id = %s, date = %s, time_slot = %s, token = %s WHERE id = %s”,(doctor_id,Date,timeSlot,token,Treatment_id)



Test-Scheduling

TestID

Enter Testid

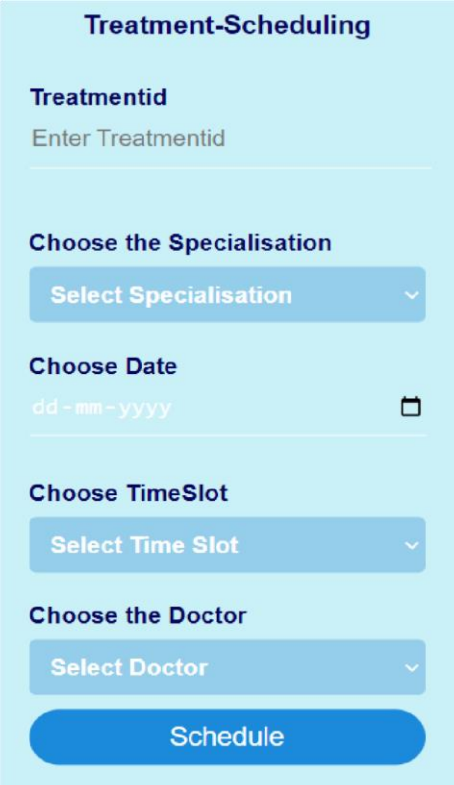
Choose Date

dd-mm-yyyy

Choose TimeSlot

Select Time Slot

Schedule



Treatment-Scheduling

Treatmentid

Enter Treatmentid

Choose the Specialisation

Select Specialisation

Choose Date

dd-mm-yyyy

Choose TimeSlot

Select Time Slot

Choose the Doctor

Select Doctor

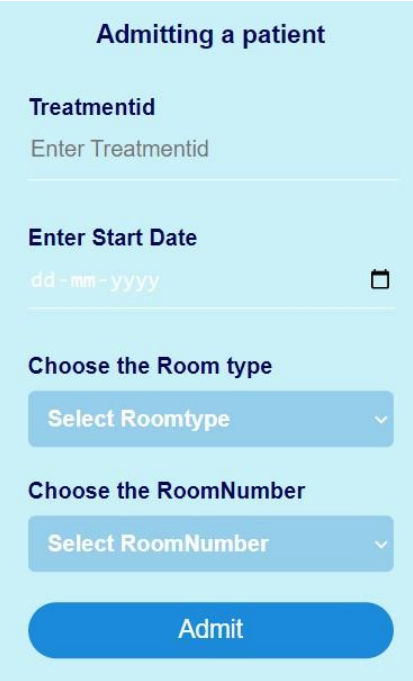
Schedule

Admitting a patient page:

If the front desk operator enters the correct Treatment id and clicks on the Admit then it shows a message that the patient got Admitted successfully, Decrease the count of vacancy in room number by 1 for the room number chosen, adds details to stay table, updates stay_id in treatment table and stay on the same page.

“INSERT INTO stay(patient_id,room_id,start_date) VALUES(%s,%s,%s)”,(patient_id,room_id,start_date)

“UPDATE treatment SET stay_id = (SELECT MAX(id) FROM stay) WHERE id=%s”, (Treatment_id,)

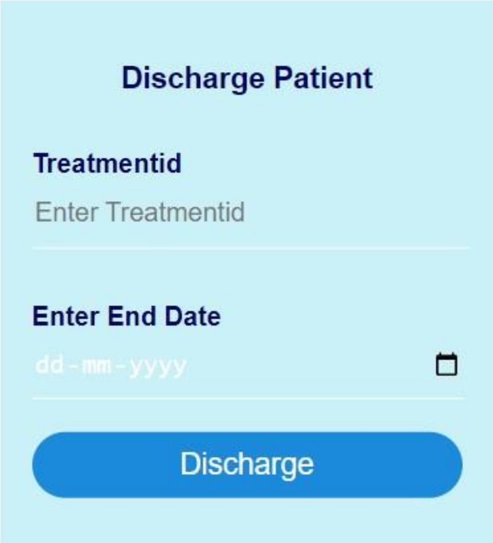


Discharge Patient page:

Increases the count of vacancy by 1 for the room number chosen, updates end_date in the stay table and stays on the same page.

“UPDATE stay SET end_date=%s WHERE id=(SELECT stay_id FROM treatment WHERE id = %s)”, (end_date,Treatment_id)

```
“UPDATE room SET count=count+1 WHERE id = (SELECT stay.room_id FROM
stay,treatment WHERE treatment.stay_id=stay.id AND treatment.id=%s)”,(Treatment_id,)
```



Discharge Patient

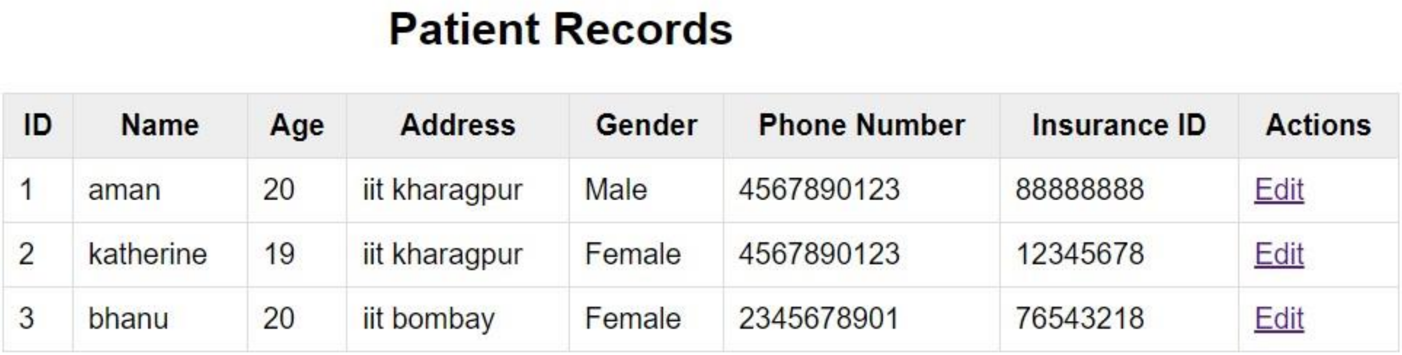
Treatmentid

Enter Treatmentid

Enter End Date

dd-mm-yyyy

Edit-Patient details page: It has a table consisting of patient details and url edit beside every row in a table. If I click the url present in a row, then it goes to the edit page.



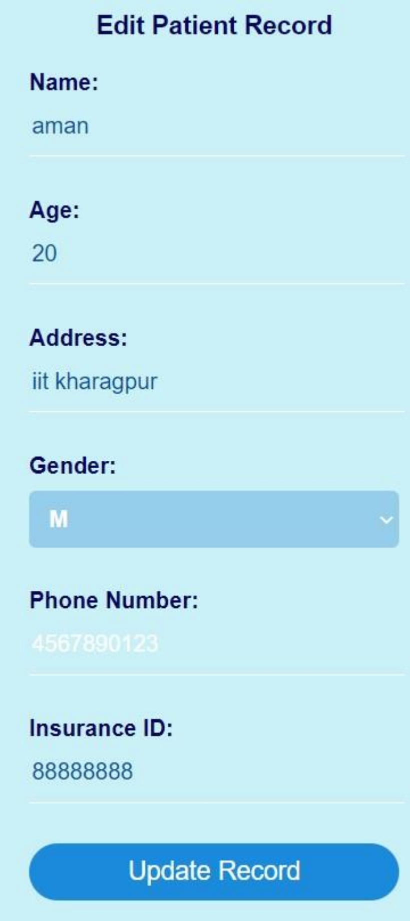
Patient Records

ID	Name	Age	Address	Gender	Phone Number	Insurance ID	Actions
1	aman	20	iit kharagpur	Male	4567890123	88888888	Edit
2	katherine	19	iit kharagpur	Female	4567890123	12345678	Edit
3	bhanu	20	iit bombay	Female	2345678901	76543218	Edit

Edit page: If the front desk operator clicks on the Update Record the data gets updated in the patient table and goes to the edit-patient details page.

"UPDATE patient SET name=%s, age=%s, address=%s, gender=%s, phone_number=%s, insurance_id=%s WHERE id=%s", (name, age, address, gender, phone_number, insurance_id, id)

If the Front Desk Operator clicks on the logout which is present in the side navigation bar then the Front Desk Operator goes to the Home page.



Edit Patient Record

Name:
aman

Age:
20

Address:
iit kharagpur

Gender:
M

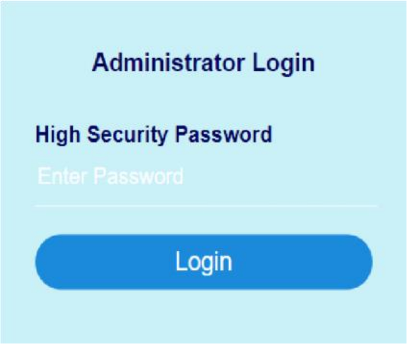
Phone Number:
4567890123

Insurance ID:
88888888

****

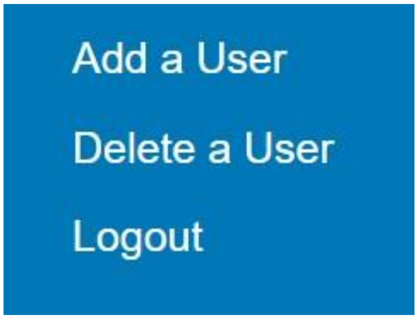
If the user clicks on the Database administration it goes to the Database administration login page.

Database Administrator login page: It has a form containing a field High security password. If the Database Administrator enters the correct password(text) and clicks on login he goes to the Database Administrator page. If the Database Administrator enters an incorrect password and clicks on login then it shows an error message that “you have entered an incorrect password. Please try again!” and the Database Administrator stays on the same page.



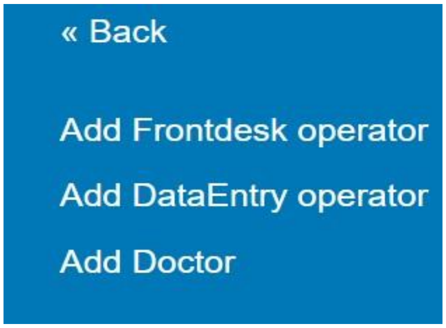
Database Administrator page:

Side Nav



Add a User:

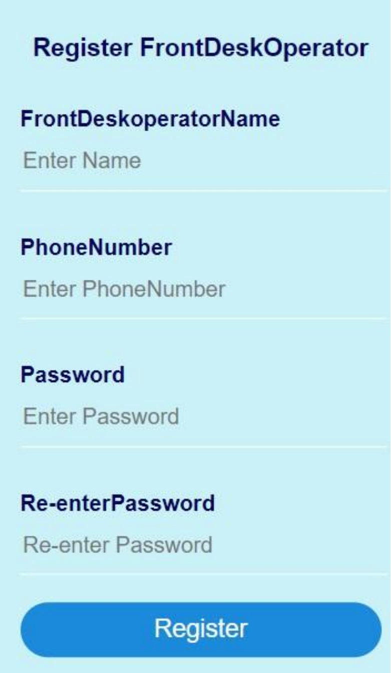
Side Nav



Add a front desk operator page:

If the database administrator enters every field correctly and then clicks on the register then the front desk operator gets registered successfully, will be added to frontdeskop table and the same status message will be shown.

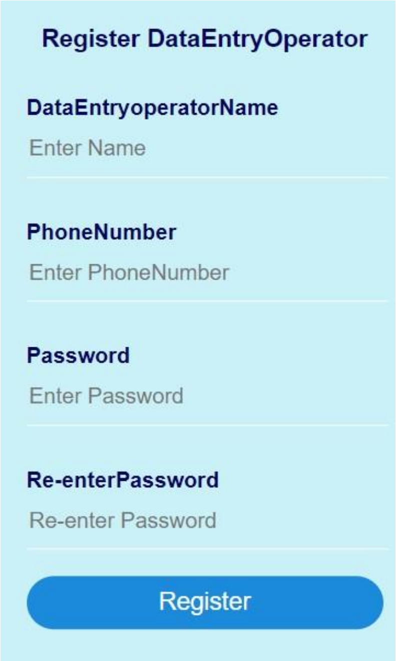
“INSERT INTO frontdeskop(username,phone_number,password) VALUES(%s,%s,%s)”, (name,phone_number,password)



Add a data entry operator page:

If the database administrator enters every field correctly and then clicks on the register then the Data entry operator gets registered successfully, will be added to dataentryop table and the same status message will be shown.

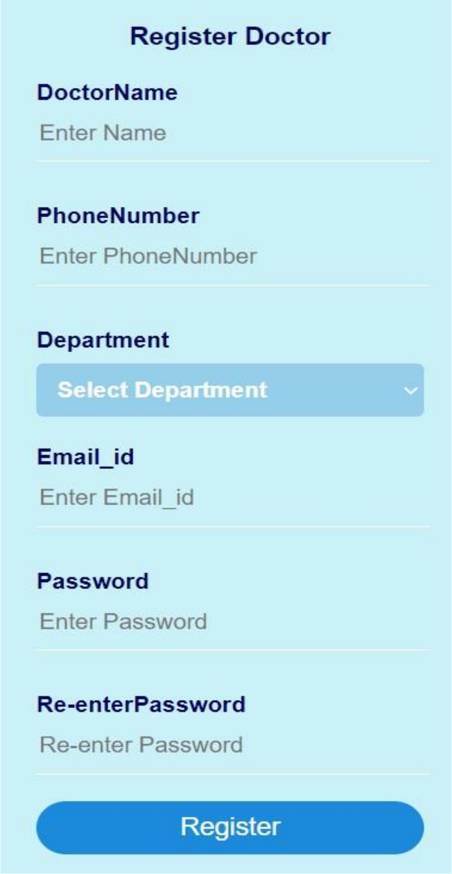
“INSERT INTO dataentryop(username,phone_number,password) VALUES(%s,%s,%s)”, (name,phone_number,password)



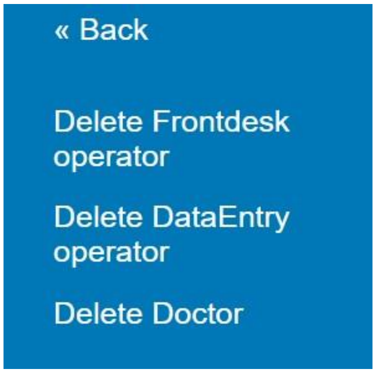
Add a doctor page:

If the database administrator enters every field correctly and then clicks on the register then the Data entry operator gets registered successfully, will be updated in the doctor table and the same status message will be shown.

```
“INSERT INTO doctor(name,department,password,phone_number,email_id)
VALUES(%s,%s,%s,%s,%s)”,(name,department_id,password,phone_number,email_id)
```



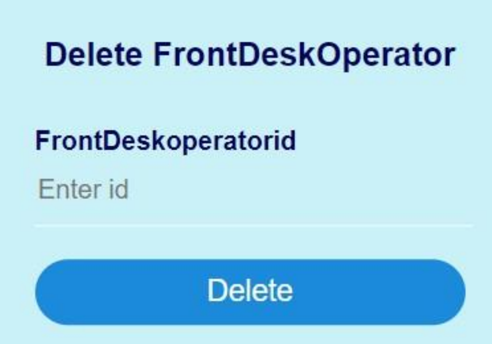
Delete a User:



Delete front desk operator page:

It removes the front desk operator with entered id from frontdeskop table

“DELETE FROM frontdeskop WHERE id = %s”,(id,)

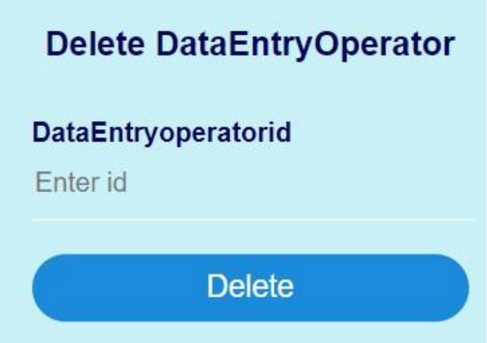

Delete FrontDeskOperator

FrontDeskoperatorid

Delete Data entry operator page:

It removes the data entry operator with entered id from dataentryop table

“DELETE FROM dataentryop WHERE id = %s”,params = (id,)

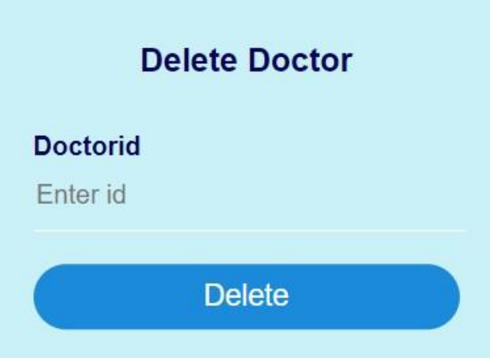

Delete DataEntryOperator

DataEntryoperatorid

Delete Doctor operator page:

It sets password of doctor as NULL with entered id from doctor table (indicating no access to doctor)

“UPDATE doctor SET password=NULL WHERE id = %s”, (id,)

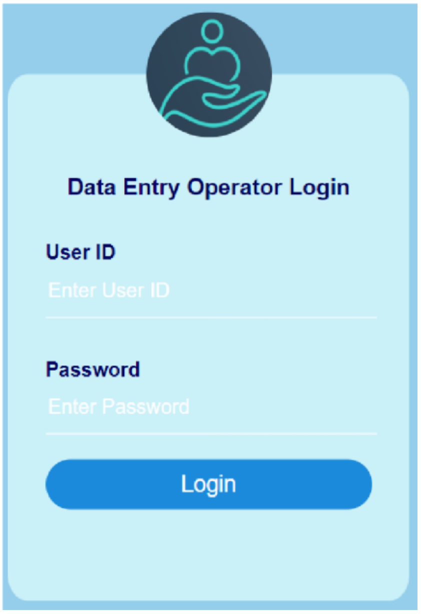

Delete Doctor

Doctorid

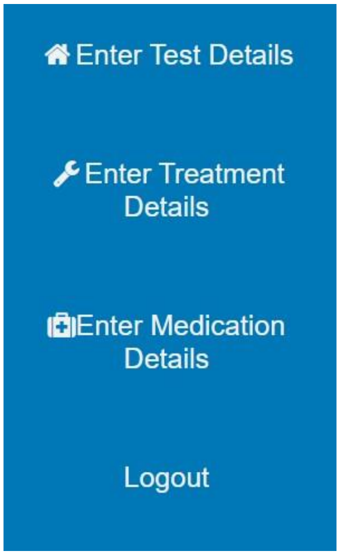
****

Data Entry Operator login page:

If the data entry operator enters an incorrect username or password and clicks on login then it shows an error message that “you have entered an incorrect username or password. Please try again!” and the Data entry operator stays on the same page.



Data Entry Operator page:



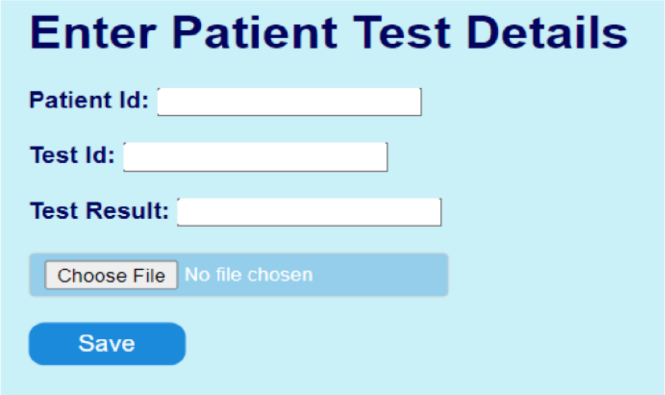
If the Data Entry Operator clicks on Enter Test Details,then theData Entry Operator redirects to the Enter Test Details Page.

Enter Test Details:

Long BLOB data type is used for storing the images of type jpg,png in mysql database.All the fields mentioned are required. Entered details will be updated in the test table. Here, Test ID is taken as a field because the doctor already adds the test name into the table to which ID will be given.

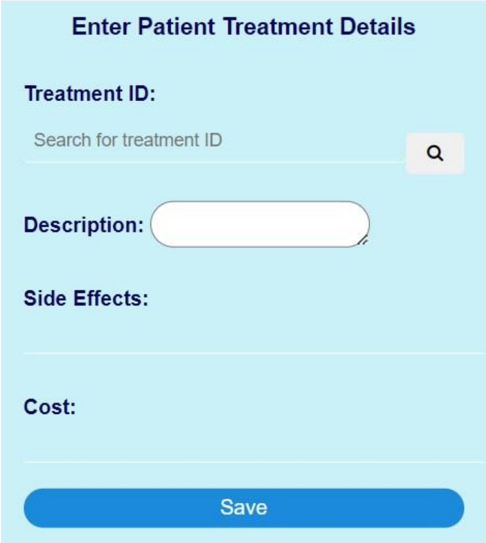
"UPDATE test SET result=%s WHERE patient_id=%s AND id=%s",(test_result, patient_id, test_id)

"UPDATE test SET test_report=%s WHERE patient_id=%s AND id=%s", (file_data, patient_id, test_id)



Enter Treatment Details: All the fields mentioned are required. Here first we enter the Treatment ID and on clicking the search button it displays the already existing description prescribed by doctor in Description field corresponding to the entered treatment id and we can edit the Description field and we also give entries for Side effects ,Cost and on clicking save button, the details get updated in treatment table if the id is valid.

“UPDATE treatment SET description=%s, side_effects=%s, cost=%s WHERE id=%s”, (treatment_description, side_effects, cost, treatment_id)

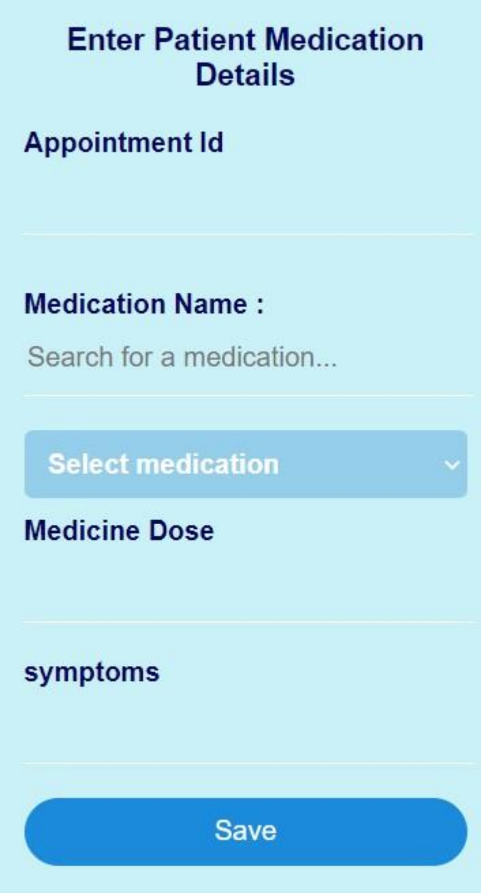


Enter Medication Details:

All the fields mentioned are required. Here first we enter Appointment ID and as Patient ID,Doctor ID are the foreign keys of prescribes table ,we retrieve the patient ID,Doctor ID for a entered Appointment ID on running a query on appointment table, select Medication name from drop down menu consisting of the medication names prescribed by doctor,from this we run a query on medication table to get medication id and we enter medication description ,medicine dose,symptoms and finally on clicking save button, the details get updated in prescribes table if the id is valid.

"INSERT INTO (patient_id,doctor_id,appointment_id,medication_id,date_prescribed,dose) VALUES (%s,%s, %s,%s, NOW(),%s)" ,(patient_id,doctor_id,appointment_id,medication_id,medicine_dose)

"UPDATE appointment SET symptoms=%s WHERE patient_id=%s AND id=%s", (symptoms, patient_id, appointment_id)



****
Doctor:

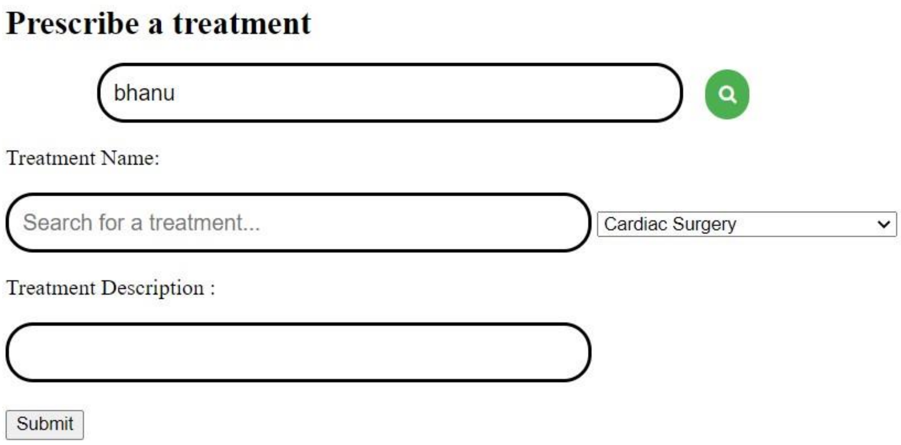
Functionalities accessible to a doctor are:

Prescribe a treatment to a patient :

Search for the patient using the patient id or the name of the patient

Prescribe the test: drop-down displays the treatments available in the hospital **SELECT id, name FROM treatments_available**

INSERT INTO treatment (patient_id, doctor_id, treatment_id, description) VALUES (%s, %s, %s , %s);



Prescribe a treatment

bhanu

Treatment Name: Cardiac Surgery

Treatment Description :

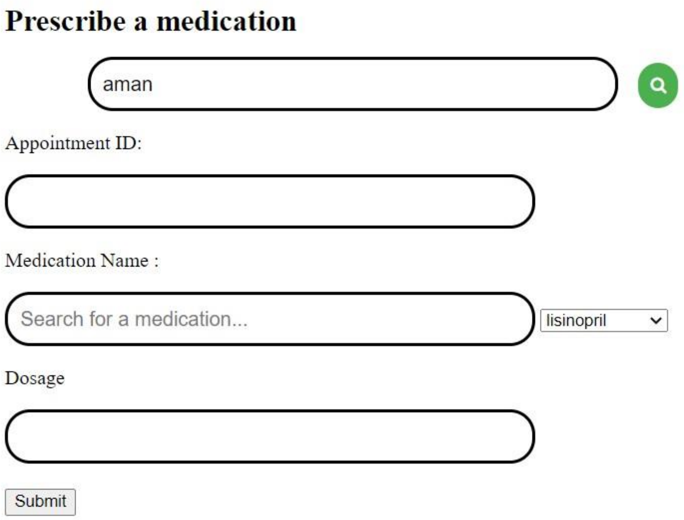
Prescribe medication to a patient :

Search for the patient using the patient id or the name of the patient

Prescribe the medicine and its dosage: drop-down displays the medications and enter the dosage manually

SELECT id , name FROM medication

INSERT INTO prescribes (patient_id, doctor_id, medication_id,date_prescribed, dose,appointment_id) VALUES (%s, %s, %s , NOW(), %s,%s);



Prescribe a medication

aman

Appointment ID:

Medication Name : lisinopril

Dosage

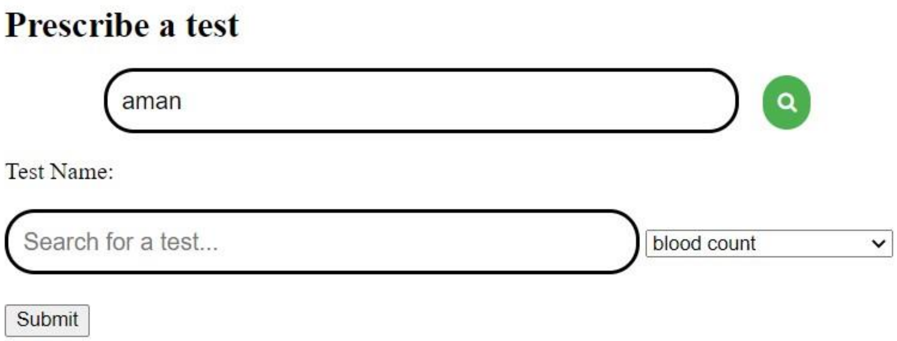
Prescribe a test to a patient :

Search for the patient using the patient id or the name of the patient

Prescribe the test: drop-down displays the tests

SELECT id, name FROM tests_available

INSERT INTO test (patient_id, test_id) VALUES (%s, %s);

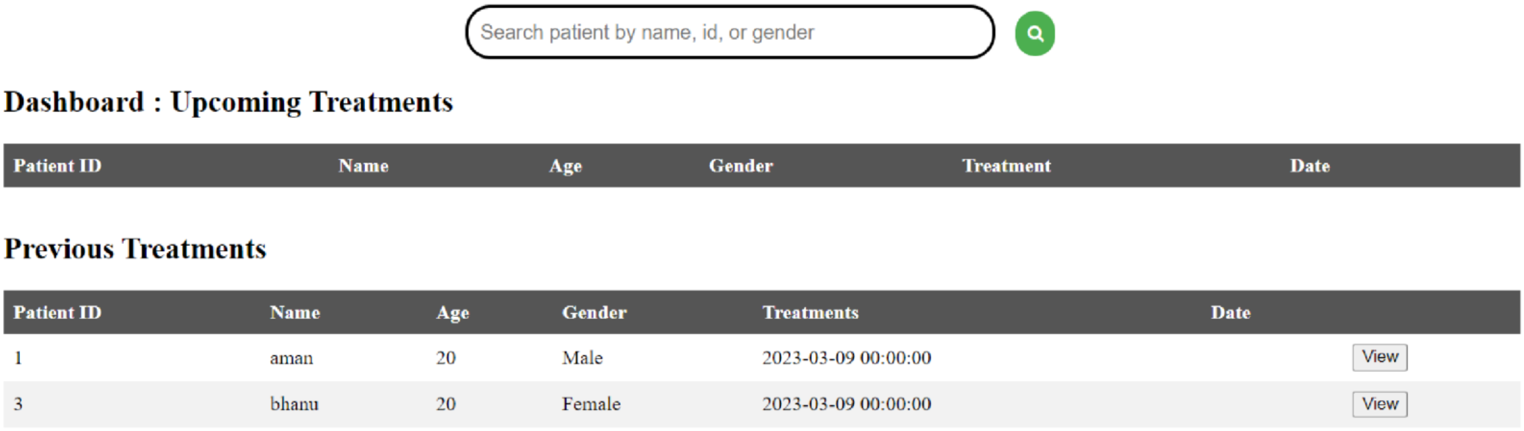


Prescribe a test

aman

Test Name: blood count

Dashboard: contains a list of patients treated by that doctor and also a view button for each patient to view his information - information includes all the treatments undergone by that patient (to see the previous medical history of the patient) and medication prescribed to him/her by this doctor



Dashboard : Upcoming Treatments

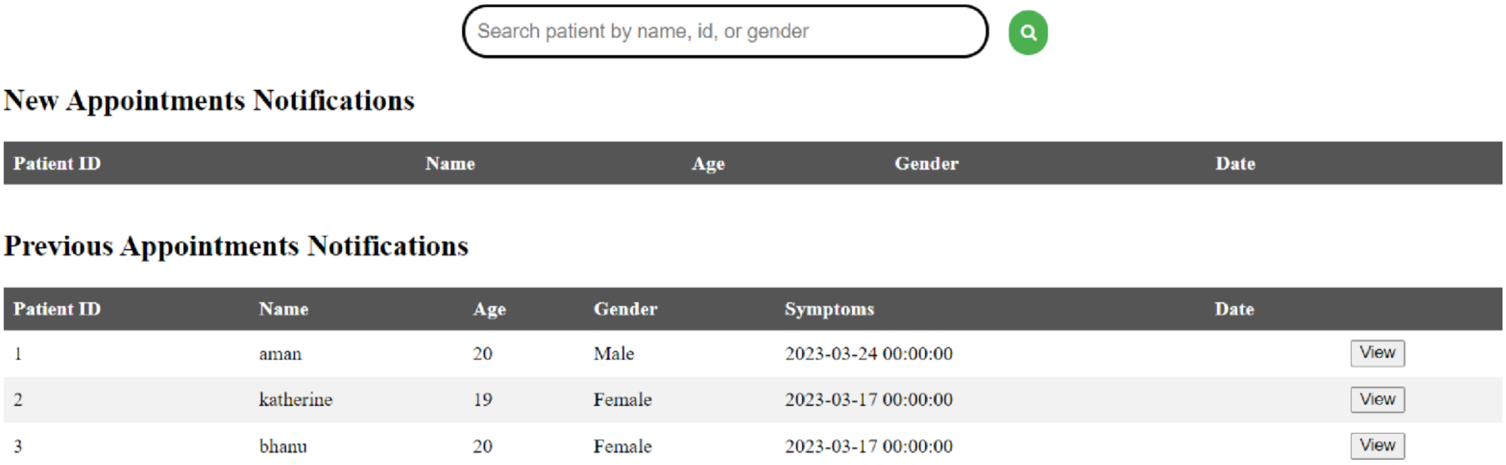
Patient ID	Name	Age	Gender	Treatment	Date

Previous Treatments

Patient ID	Name	Age	Gender	Treatments	Date	
1	aman	20	Male	2023-03-09 00:00:00		View
3	bhanu	20	Female	2023-03-09 00:00:00		View

Appointments: The doctor will be notified if a new appointment is made with him (count is displayed)

Contains the list of new appointment notifications and previous appointment notifications. Each notification includes the patient Id, name, age, gender



New Appointments Notifications

Patient ID	Name	Age	Gender	Date

Previous Appointments Notifications

Patient ID	Name	Age	Gender	Symptoms	Date	
1	aman	20	Male	2023-03-24 00:00:00		View
2	katherine	19	Female	2023-03-17 00:00:00		View
3	bhanu	20	Female	2023-03-17 00:00:00		View

6. In addition to the above features, doctor can search for a patient using his/her name or patient id, and can see the patient's medical history which includes Medications prescribed to the patient by this doctor - medication name, dosage, date, symptoms related

Treatments undergone by the patient -information includes treatment name, description of the treatment, the date on which treatment was taken place

Tests taken by the patients - information includes test name, the result of the test, and the test report (eg-x rays in image formats, etc)

Search for the patient using the patient id or the name of the patient:

SELECT id FROM patient WHERE name LIKE %s OR age LIKE %s OR gender LIKE %s

7. Weekly health report of patients treated by him is emailed to him

Query to get patient information: **SELECT patient.id, patient.name, age, address, gender, patient.phone_number, description FROM (patient JOIN treatment ON treatment.patient_id = patient.id AND treatment.doctor_id=%s) JOIN doctor ON treatment.doctor_id = doctor.id AND doctor.email_id IS NOT NULL**

Thankyou

Amulya Reddy Pala