



Indian Institute of Technology, Kharagpur
Department of Computer Science and Engineering

CS39202 : DATABASE MANAGEMENT SYSTEMS LAB

ASSIGNMENT 4: WEB APPLICATION DEVELOPMENT

7th March, 2023

Members :

Shivam Raj
20CS10056

Jatin Gupta
20CS10087

Kushaz Sehgal
20CS30030

Rushil Venkateswar
20CS30045

Aniket Kumar
20CS10083

Contents

1. Structure of the Database	2
2. System Languages and Tools	3
2.1 Language:	3
2.2 Tools:	3
3. Functionalities of the Entities	3
3.1 Administrator	3
3.1.1 Administrator's Dashboard	3
3.1.2 Add/Delete Users	4
3.1.3 Add Rooms	4
3.1.4 Implementing Data Security with suitable Access Control	5
3.2 FrontDesk Operator	6
3.2.1 FrontDesk Operator - Dashboard	6
3.2.2 FrontDesk Operator - Register Patient	7
3.2.3 FrontDesk Operator - Admit Patient	7
3.2.4 FrontDesk Operator - Discharge Patient	8
3.2.5 FrontDesk Operator - Schedule Appointment	8
3.3 DataEntry Operator	9
3.3.1 DataEntry Operator - Dashboard	9
3.3.2 DataEntry Operator - Add Treatment	10
3.3.3 DataEntry Operator - Add Test Result	11
3.4 Doctor	12
3.4.1 Doctor - Dashboard	12
3.4.2 Query Patient	13
3.4.3 Assign Prescriptions	15
3.4.4 Email Report Automation	16
4. Appendix	17
4.1 SQL Queries Used	17

1. Structure of the Database

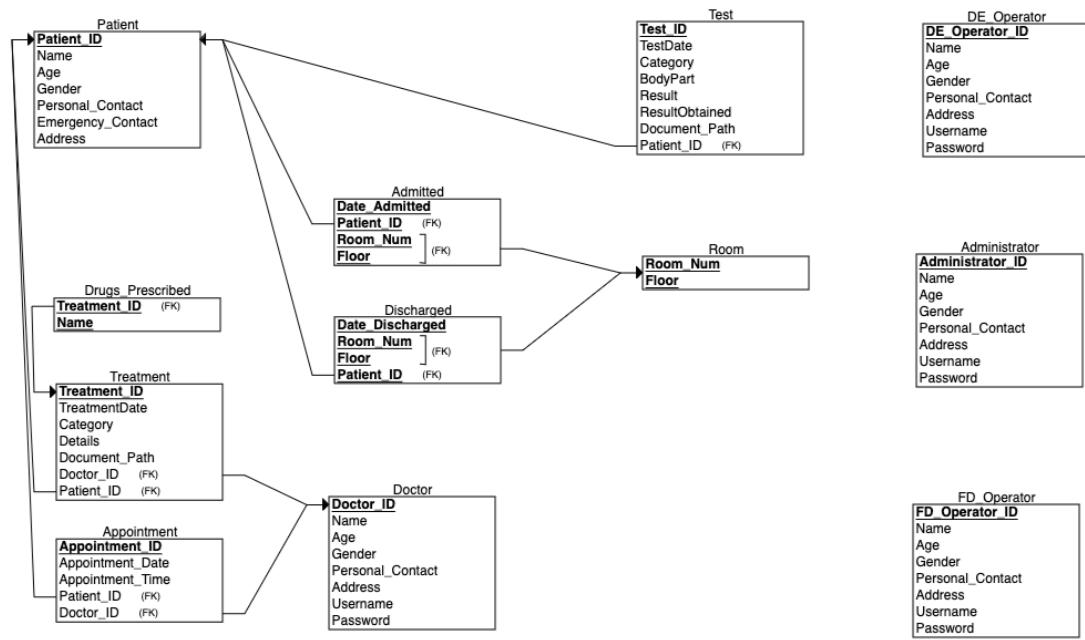


Figure 1: Relational Schema for Hospital Management System

2. System Languages and Tools

We describe below the tech-stack we have used to implement Hospital Management System (HMS)

2.1 Language:

- HTML: structure of web page
- CSS: Add style and formatting to web page
- Bootstrap: CSS framework to provide responsive and mobile-first design
- Python: for back-end of application

2.2 Tools:

- Flask: lightweight and flexible Python-based web framework for back-end of application
- MySQL: Open source relational database management system
- Flask-Login: Handling user authentication and authorization
- Flask-WTF: Creating HTML forms in back-end
- Werkzeug: Password hashing and secure filename
- Flask-Mail: Sending mails using SMTP server
- Flask-APScheduler: Scheduling weekly mails
- PDFKit: generating health reports in PDF format

3. Functionalities of the Entities

3.1 Administrator

The Administrator serves to add/delete other users to/from the database. By design, the Administrator can add all types of users (Doctors, Front Desk Operators, Data Entry Operators and other Admins as well).

Further, an additional feature provided is the ability to 'Register a Room'. This is just a convenience feature which allows us to register rooms in the MySQL Database through the GUI.

3.1.1 Administrator's Dashboard

Soni Group of Hospitals					
Administrator					
Main DashBoard	Total Doctors 3	Total FrontDesk Operators 1	Total DataEntry Operators 1		Logout
Add/Delete User					
Register a Room					
Recently Registered Doctors					
Doctor ID	Name	Address	Age	Gender	Personal_Contact
3	Dr. A	Random	21	Male	1246019240
4	Dr. B	Random	21	Male	1480129401
5	Dr. C	Random	21	Male	1241820401
Recently Registered FrontDesk Operators					
FD_Operator ID	Name	Address	Age	Gender	Personal_Contact
1	Rushil	C-14	21	Male	1249019240
Recently Registered DataEntry Operators					
DE_Operator ID	Name	Address	Age	Gender	Personal_Contact
1	Rushil	C-14	21	Male	1280419204

Figure 2: Admin Dashboard

The Administrator's Dashboard gives information about the number of currently registered users of each type in the database. It displays three tables each with the following fields -

1. User ID
2. Name
3. Address
4. Age
5. Gender
6. Personal Contact Number

Only the top 5 recently registered Users of each type are displayed there.

3.1.2 Add/Delete Users

The screenshot shows a web-based administrative dashboard for 'Sonit Group of Hospitals'. The main header includes the hospital name, a 'Logout' button, and a 'View Doctors' section. On the left, there's a sidebar with links for 'Administrator', 'Main DashBoard', 'Add/Delete User' (which is highlighted in red), and 'Register a Room'. The central content area is titled 'View Doctors' and contains a table with the following data:

Doctor ID	Name	Address	Age	Gender	Contact No.	Delete
3	Dr. A	Random	21	Male	1248018240	<button>Delete</button>
4	Dr. B	Random	21	Male	1480129401	<button>Delete</button>
5	Dr. C	Random	21	Male	1241820401	<button>Delete</button>

Figure 3: Admin Add/Delete User

Provides an option to select the type of user to be added/deleted. A table is displayed once the user type has been selected and each entry in the table has a delete button. There is also an 'Add User' button above the table.

3.1.3 Add Rooms

Provides functionality to add a room with Room number and Floor.

3.1.4 Implementing Data Security with suitable Access Control

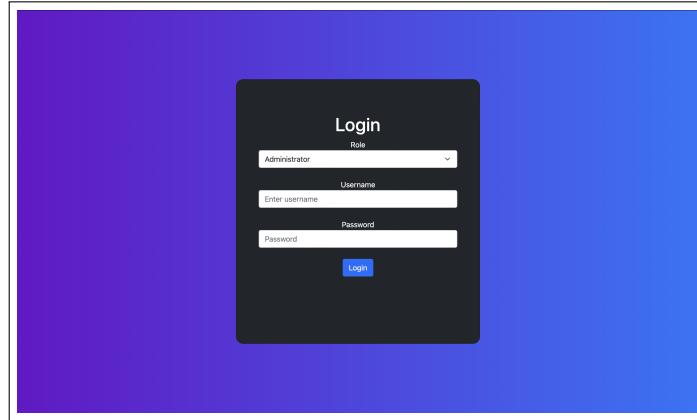


Figure 4: Login Page

We have provided a Login page with the option to select the type of User that is trying to login. This ensures restricted access to functionalities (i.e. a doctor cannot delete another doctor as that power exists only with the Administrator). If some user tries to access the page of another user without logging in by forcefully entering the corresponding URL (eg. a front desk operator trying to access admin page by typing `http://127.0.0.1:5000/admin` in the browser), they will be denied. All users need to be signed in first before accessing functionalities.

This is implemented by using two decorator functions:

1. `@login_required` : specifies that a particular URL is accessible only when the user is logged in and authenticated
2. `@requires_access_level()` : specifies that a particular URL can be accessed only by a user with a given access level
 - (a) Administrator : `access_level= 1`
 - (b) Doctor : `access_level= 2`
 - (c) Front Desk Operator : `access_level= 3`
 - (d) Data Entry Operator : `access_level= 4`

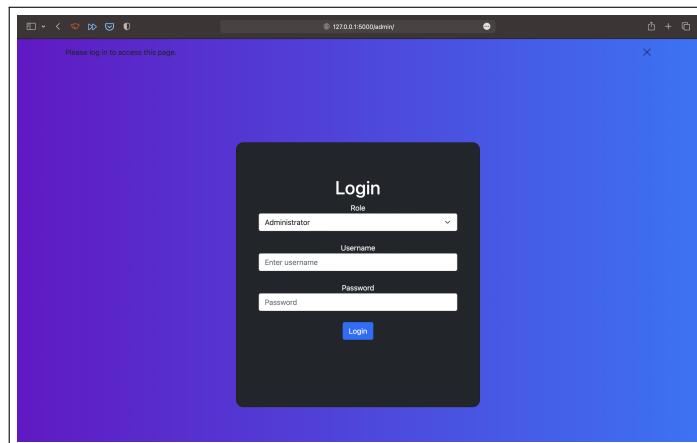


Figure 5: Pop-Up in case of unauthorised access

3.2 FrontDesk Operator

The functionalities of the FrontDesk operator includes registering a new patient, admitting a registered patient, discharging an admitted patient and scheduling appointments with the doctor. All these functionalities are achieved using a user-friendly web interface and a well-designed database schema. The FrontDesk operator has the following pages:

3.2.1 FrontDesk Operator - Dashboard

Frontdesk Operator						
Main Dashboard	Total Patients	Admitted Patients	Available Rooms			
Register Patient	4	2	0			
Admit Patient						
Discharge Patient						
Schedule Appointment	Recent Patients Registered					
Patient ID	Name	Address	Age	Gender	Personal_Contact	Emergency_Contact
4	Mr. Z	ABCDE	29	Male	1204910240	2491294912
3	Mrs. Y	ABCD	25	Female	1204120408	1204921049
2	Mr. X	ABCD	21	Male	1240818208	1204128401
1	hjfdks	jfdksh	231	Male	1234567890	1234567890
Admitted Patients						
Admission Date	Patient ID	Room Number	Floor			
2023-03-08	1	212	2			
2023-03-08	3	1	1			
Discharged Patients						
Discharge Date	Patient ID	Room Number	Floor			
2023-03-07	1	1	1			
2023-03-08	2	1	1			

Figure 6: Frontdesk Operator's Dashboard

The FrontDesk's dashboard displays insights about the database. It tells about the total patients registered, total admitted patients and the total rooms in the hospital. It also displays the list of recently added patients and admitted patients in the hospital using two tables:

1. Recent Patients Registered

This table displays information about the recently registered patients using the following fields:

- PatientID
- Name
- Address
- Age
- Gender
- Personal Contact
- Emergency Contact

2. Admitted Patients

This table describes about the admitted patients using the following fields:

- Admission Date
- PatientID
- Room Number
- Floor

3.2.2 FrontDesk Operator - Register Patient

Patient Name
Gender: Male Patient Age
Patient Contact Number
Patient Emergency Contact
Patient Address
Street, Locality, City, State
Register Patient

Figure 7: Register Patient Page

The FrontDesk Operator can use this option to register a new patient to the hospital database. He/She has to enter the following details about the patient, after which the patient will be added to the hospital database:

- Patient Name
- Gender
- Patient Age
- Patient Contact Number
- Patient Emergency Number
- Patient Address

3.2.3 FrontDesk Operator - Admit Patient

Patient ID	Name	Address	Age	Gender	Contact No.	Emergency No.	Admit
2	Mr. X	ABCD	21	Male	1234567890	1204123456	Admit
4	Mr. Z	ABCDE	29	Male	1204910240	2491234912	Admit

Figure 8: Admit Patient Page

This functionality allows the FrontDesk Operator to admit an already registered patient. On selecting this option, a list of registered non-admitted patients is displayed. The Operator has to click on the Admit button to the corresponding patient which has to be admitted. After selecting the patient, if a room is available, the patient will be admitted.

3.2.4 FrontDesk Operator - Discharge Patient

The screenshot shows a web-based application for a hospital's front desk operator. The main menu on the left includes options like Main Dashboard, Register Patient, Admit Patient, Discharge Patient (which is highlighted in blue), and Schedule Appointment. The central area is titled "Discharge Patient" and displays a table of admitted patients:

Patient ID	Name	Address	Age	Gender	Contact No.	Emergency No.	Discharge
1	hfjksd	jdksjh	23	Male	1234567890	1234567890	<button>Discharge</button>
3	Mrs. Y	ABCD	25	Female	1204120408	1204921049	<button>Discharge</button>

Figure 9: Discharge Patient Page

The FrontDesk Operator can select this option to discharge an admitted patient. This page displays a table of admitted patients in the hospital. On clicking the Discharge button corresponding to the patient to be discharged, the patient will be discharged.

3.2.5 FrontDesk Operator - Schedule Appointment

The screenshot shows a web-based application for a hospital's front desk operator. The main menu on the left includes options like Main Dashboard, Register Patient, Admit Patient, Discharge Patient, and Schedule Appointment (which is highlighted in blue). A modal window titled "Select Priority" is open over the main content area. It contains two radio buttons: "Urgent" and "Normal". Below the buttons are "Close" and "Submit" buttons. The background of the main area is dimmed.

Figure 10: Schedule an Appointment Page

This functionality allows the FrontDesk Operator to schedule appointments of patients with doctors according to the priority. There exist two types of priority:

- **Urgent:** If it's an urgent appointment, then the appointment will be scheduled at the earliest slot available next day with any doctor.
- **Normal:** If it is a normal appointment, then the Operator has to select the doctor and the appointment date. If there exist a free slot of the selected doctor on that date, the appointment will be scheduled.

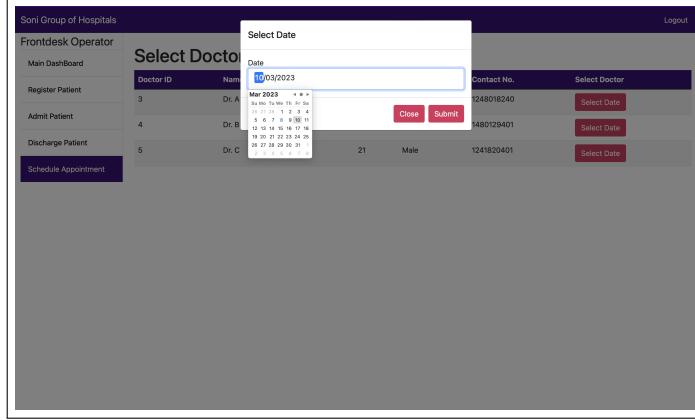


Figure 11: Schedule a 'Normal' Appointment Page (date selector)

3.3 DataEntry Operator

The purpose of the entity - **Dataentry Operator** is to add tests results and treatments to patients by their respective doctors. The implemented functionalities are as follows -

3.3.1 DataEntry Operator - Dashboard

Treatments					
Treatment ID	Treatment Date	Category	Details	Doctor Name	Patient Name
1	2023-03-08	Operation	Liver Transplant	Dr. A	Mr. X
2	2023-03-08	General	Appendectomy	Dr. C	Mrs. Y
3	2023-03-08	Physiotherapy	Dry Needling	Dr. B	Mr. X

Tests - Completed					
Test ID	Test Date	Category	Patient Name	Body Part	Results
4	2023-03-08	Biopsy	Mrs. Y	Lung	Lung Cancer

Tests - InComplete					
Test ID	Test Date	Category	Patient Name	Body Part	Body Part
1	2023-03-03	PET Scan	Mjds	Brain	
2	2023-03-08	CT Scan	Mjds	Lung	
3	2023-03-08	Ultrasound	Mrs. Y	Liver	
5	2023-03-08	PET Scan	Mr. Z	Brain	

Figure 12: DataEntry Operator's Dashboard

The DataEntry Operator's Dashboard describes the entire list of tests and treatments assigned to any patient admitted to the hospital.

It contains three tables -

1. Treatments

The Treatment table describes each treatment, which has the following fields

- Treatment ID
- Treatment Date
- Treatment Category
- Doctor Name
- Patient Name
- Details Regarding Treatment

2. Completed Tests

The Completed Tests table describes each test whose result has been obtained, which has the following fields -

- Test ID
- Test Date
- Test Category
- Doctor Name
- Patient Name
- Associated Body Part
- Details Regarding Test Result

3. Pending Tests

The Pending Tests table describes each test whose result has been NOT been obtained, which has the following fields -

- Test ID
- Test Date
- Test Category
- Doctor Name
- Patient Name
- Associated Body Part

3.3.2 DataEntry Operator - Add Treatment

The screenshot shows a web-based application for a DataEntry Operator. The top navigation bar includes 'Sonni Group of Hospitals', 'Logout', 'DataEntry Operator', 'Main Dashboard', and 'Add Test Result'. The main content area is titled 'Select Patient for Adding Treatment'. A table lists four patients with columns: Patient ID, Name, Address, Age, Gender, Contact No., Emergency No., and Add Treatment. Each row has a red 'Add Treatment' button in the last column.

Patient ID	Name	Address	Age	Gender	Contact No.	Emergency No.	Add Treatment
1	hfkds	jfdksh	231	Male	1234567890	1234567890	Add Treatment
2	Mr. X	ABCD	21	Male	1240818208	1204128401	Add Treatment
3	Mrs. Y	ABCD	25	Female	1204120408	1204921049	Add Treatment
4	Mr. Z	ABCDE	29	Male	1204910240	2491294912	Add Treatment

Figure 13: DataEntry Operator Add Treatment

This functionality allows the data entry operator to choose a patient and his/her's corresponding doctor to assign a treatment to said patient.

We ask the DataEntry operator to enter the following fields -

- Treatment Date
- Treatment Doctor (uneditable field, chosen earlier)
- Treatment Patient (uneditable field, chosen earlier)
- Details Regarding the Treatment
- Any Report / File associated to the treatment (Optional)

Figure 14: DataEntry Operator Treatment Form

3.3.3 DataEntry Operator - Add Test Result

Test ID	Test Date	Category	Patient Name	Associated Body Part	Test Result
1	2023-05-03	PET Scan	hijklkjds	Brain	<button>Add Test Result</button>
2	2023-03-08	CT Scan	hijklkjds	Lung	<button>Add Test Result</button>
3	2023-03-28	Ultrasound	Mrs. Y	Liver	<button>Add Test Result</button>
5	2023-03-08	PET Scan	Mr. Z	Brain	<button>Add Test Result</button>

Figure 15: DataEntry Operator Add Test Result

This functionality allows the dataentry operator to choose a pending test, wherein the dataentry operator will add the result to said test.

We ask the DataEntry operator to enter the following fields to add a test result -

- Test Date (uneditable field, chosen earlier)
- Test Doctor (uneditable field, chosen earlier)
- Test Patient (uneditable field, chosen earlier)
- Test Category (uneditable field, chosen earlier)
- Associated Body Part (uneditable field, chosen earlier)
- Details Regarding the Test Result
- Any Report / File associated to the Test (Optional)

Figure 16: DataEntry Operator Test Form

Bonus Functionality: Upload a File

As seen above we allow the DataEntry Operator to upload a treatment/test report (image, excel sheet, pdf and word document are supported file types) for the corresponding test/treatment.

3.4 Doctor

The purpose of the entity - **Doctor** is to display all the records of the patients treated by him/her as a dashboard, query for any patient information, and to be able to record drugs/treatments prescribed to a patient. The implemented functionalites are as follows -

3.4.1 Doctor - Dashboard

Patient ID	Patient Name	Patient Age	Patient Gender	Patient Address	Patient Personal Contact	Patient Emergency Contact
2	Mr. X	21	Male	ABCD	1240818208	1204128401
1	hfjksd	231	Male	jfdksh	1234567890	1234567890
4	Mr. Z	29	Male	ABCDE	1204910240	2491294912

Appointment ID	Appointment Date	Appointment Time	Patient Name	Patient Age	Patient Gender
1	2023-03-11	10:00:00	Mr. X	21	Male
2	2023-03-12	10:00:00	Mr. Z	29	Male

Figure 17: Doctor Dashboard

The Doctor's Dashboard describes the entire list of patients(along with their details) that have been treated by the Doctor and any upcoming appointments along with some of the patient name that has made the appointment in a sorted manner.

It contains two tables -

1. Patients

The Patient table describes each patient that the doctor has treated and has the following fields

- Patient ID
- Patient Name
- Patient Age
- Patient Gender
- Patient Address
- Patient Personal Contact
- Patient Emergency Contact

2. Your Appointments

The Appointments table describes each upcoming Appointment briefly in a sorted manner and has the following fields -

- Appointment ID
- Appointment Date
- Appointment Time
- Patient Name
- Patient Age
- Patient Gender

3.4.2 Query Patient

The screenshot shows a web-based application interface for a doctor. At the top, there is a purple header bar with the text "Soni Group of Hospitals" on the left and "Logout" on the right. Below the header, the word "Doctor" is displayed. Under "Doctor", there are two main menu items: "Main Dashboard" and "Query Patients", with "Query Patients" being highlighted by a blue background. To the right of these menus is a section titled "Patients". This section contains a table with the following data:

Patient ID	Name	Address	Age	Gender	Personal Contact	Emergency Contact	Medical History
1	Hjkds	Jfdish	23	Male	1234567890	1234567890	<button>Details</button>
2	Mr. X	ABCD	21	Male	1240818208	1204128401	<button>Details</button>
3	Mrs. Y	ABCD	25	Female	1204120408	1204921049	<button>Details</button>
4	Mr. Z	ABCDE	29	Male	1204910240	2491294912	<button>Details</button>

Figure 18: Doctor Query Patients

The Query Patient Page allows the doctor to see a list of all Registered Patients in a tabular form, this allows the Doctor to quickly find the required patient and look up more details about his/her previous tests or treatments.

It contains the following Table-

- Patient ID
- Name
- Address
- Age

- Gender
- Personal Contact
- Emergency Contact
- Medical History

Upon clicking the Medical History button the Doctor will have access to the Medical History of the selected Patient and will be directed to a page showing details about all the tests and treatments, as shown below:-

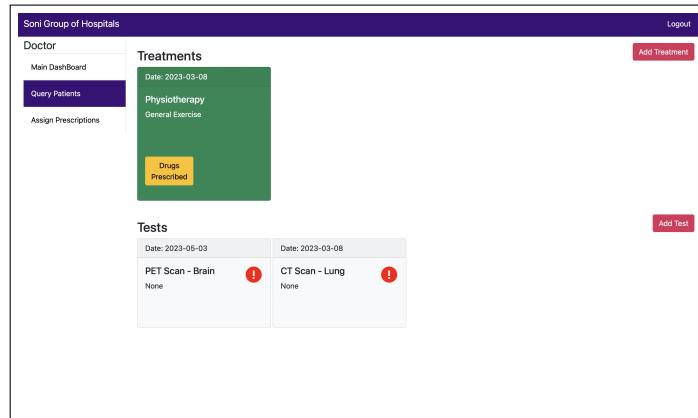


Figure 19: Doctor Query Patient Details

The Doctor also has the functionality to Add Tests and Treatments to any selected Patient, through a modal interface as shown below:-

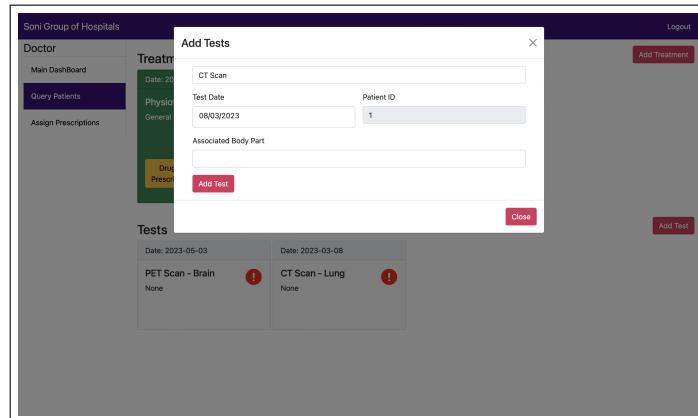


Figure 20: Doctor Add Test

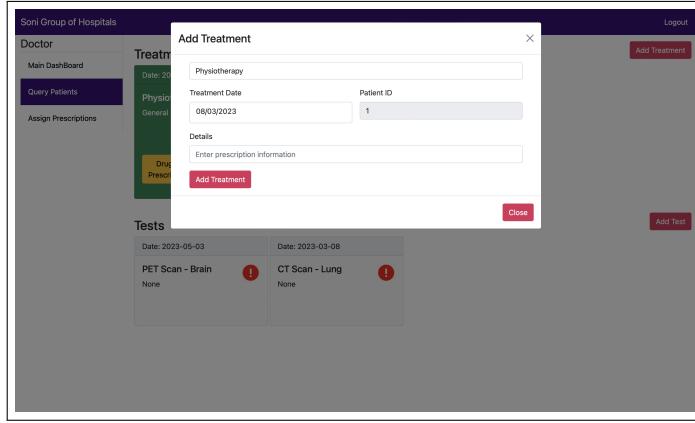


Figure 21: Doctor Add Treatment

3.4.3 Assign Prescriptions

The Doctor also has the functionality to assign Prescriptions to selected patients for this the doctor chooses a patient from the patient Table to add the prescription to.

The Patient Table contains the following fields-

- Treatment ID
- Treatment Date
- Category
- Details
- Patient Name
- Patient Age
- Patient Gender
- Add Prescription

This can be seen below:-

Patients							Add Prescription
Treatment ID	Treatment Date	Category	Details	Patient Name	Patient Age	Patient Gender	
1	2023-03-08	Operation	Liver Transplant	Mr. X	21	Male	<button>Add</button>
2	2023-03-08	General	Appendectomy	Mrs. Y	25	Female	<button>Add</button>
3	2023-03-08	Physiotherapy	Dry Needling	Mr. X	21	Male	<button>Add</button>
4	2023-03-08	Physiotherapy	General Exercise	hfkds	231	Male	<button>Add</button>
5	2023-03-08	Operation	Brain Biopsy	Mr. Z	29	Male	<button>Add</button>

Figure 22: Doctor Add Treatment

The button Add Prescription opens a modal and allows the Doctor to easily and quickly assign the required prescription to the selected Patient the picture of the modal is shown below:-

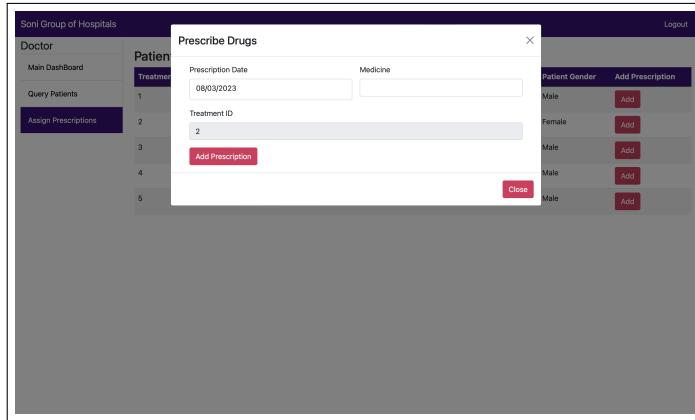


Figure 23: Doctor Add Treatment

3.4.4 Email Report Automation

The web application sends automated email reports to doctors about their treated patients' health information weekly. We utilized **APScheduler** library to schedule emails. The email is sent using Flask-Mail extension which configures the **SMTP** server and provides necessary credentials for authentication. A health report including tests, treatments and other health details about the patient is generated in PDF format using **pdfkit** library and attached to the mail. This feature ensures that doctors are up-to-date with patient's health status.

4. Appendix

4.1 SQL Queries Used

1. "SELECT distinct Patient_ID FROM Treatment WHERE Doctor_ID = %s", (doctor_id[0],) : To fetch patients being treated by a particular doctor to whom mails are to be sent.
2. "SELECT Doctor_ID, Name, Address, Age, Gender, Personal_Contact FROM Doctor ORDER BY Doctor_ID DESC LIMIT 5" : Get 5 most recently added doctors.
3. f"SELECT user_type_ID, Name, Address, Age, Gender, Personal_Contact FROM user_type" : To fetch all users of a particular type.
4. f"INSERT INTO user_type (Username, Password, Name, Address, Age, Gender, Personal_Contact) VALUES ('form_1.username.data', 'generate_password_hash(form_1.password1.data, method='sha256')', 'form_1.name.data', 'form_1.address.data', 'form_1.age.data', 'form_1.gender.data', 'form_1.contact_number.data')" : To insert a row in user table.
5. In order to delete a doctor, we need to delete the entries with same user_id from Treatment/Appointment tables first :
 - f"DELETE FROM Drugs_Prescribed WHERE Treatment_ID IN (SELECT Treatment_ID FROM Treatment WHERE user_type_ID = user_id)"
 - f"DELETE FROM Treatment WHERE user_type_ID = user_id"
 - f"DELETE FROM Appointment WHERE user_type_ID = user_id"
 - f"DELETE FROM user_type WHERE user_type_ID = 'user_id'"
6. Tables displayed on DataEntry Operator's Dashboard
 - "SELECT Test_ID , TestDate, Category , Patient.Name, BodyPart FROM Test JOIN Patient where Test.Patient_ID = Patient.Patient_ID and Test.ResultObtained = 0"
 - "SELECT Test_ID , TestDate, Category , Patient.Name, BodyPart , Result FROM Test JOIN Patient where Test.Patient_ID = Patient.Patient_ID and Test.ResultObtained = 1"
 - "SELECT Treatment_ID, TreatmentDate, Category, Details, Doctor.Name, Patient.Name FROM Treatment JOIN Patient JOIN Doctor where Treatment.Patient_ID = Patient.Patient_ID and Treatment.Doctor_ID = Doctor.Doctor_ID"
7. "SELECT Test_ID , TestDate, Category , Name, BodyPart FROM Test JOIN Patient where Test.Patient_ID = Patient.Patient_ID and Test.ResultObtained = 0" : Display all tests
8. f"UPDATE Test SET ResultObtained = 1 , Result = 'form.result.data', Document_Path = 'patient_data_path + filename'WHERE Test_ID = 'test_id'" : Add result for a test created by Doctor
9. f"INSERT INTO Treatment (Patient_ID, Doctor_ID, TreatmentDate, Category, Details, Document_Path) VALUES ('patient_id', 'form.doctor_id.data', 'form.treatment_date.data', 'form.category.data', 'form.details.data', 'patient_data_path + filename')" : Assign a Doctor to a Patient and add to Treatment table
10. Tables displayed on FrontDesk Operator's Dashboard :
 - "SELECT * FROM Patient ORDER BY Patient_ID DESC LIMIT 5"
 - "SELECT * FROM Admitted"
 - "SELECT * FROM Discharged"
 - "SELECT * FROM Room WHERE (Room_Num, Floor) NOT IN (SELECT Room_Num, Floor FROM Admitted)" : Rooms which are currently unoccupied

11. "INSERT INTO Patient (Name, Address, Age, Gender, Personal_Contact, Emergency_Contact) VALUES (%s, %s, %s, %s, %s, %s)", (form.name.data, form.address.data, form.age.data, form.gender.data, form.contact_number.data, form.emergency_contact.data) : Register a new patient
12. "SELECT * FROM Patient WHERE Patient_ID NOT IN (SELECT Patient_ID FROM Admitted)" : All patients not currently admitted but registered in the DataBase
13. Discharge a patient and insert it into discharged table
 - f"DELETE FROM Admitted WHERE Patient_ID = patient_id"
 - "INSERT INTO Discharged (Patient_ID, Room_Num, Floor, Date_Discharged) VALUES (%s, %s, %s, %s)", (patient_id, room[0], room[1], date)
14. "SELECT Doctor_ID, Name FROM Doctor WHERE Doctor_ID NOT IN (SELECT Doctor_ID FROM Appointment WHERE Appointment_Date = %s)", (date_to_schedule,) : Get the doctors free at the desired date
15. "INSERT INTO Appointment (Patient_ID, Doctor_ID, Appointment_Date, Appointment_Time) VALUES (%s, %s, %s, %s)", (patient_id, doctors_id, date_to_schedule, '10:00:00') : Schedule an appointment
16. "INSERT INTO Drugs_Prescribed (Name, Treatment_ID) VALUES (%s, %s)", (form.medicine.data, form.treatment_id.data) : Add a prescription