THE UNIVERSITY OF TEXAS AT DALLAS.

CS 6360 DATABASE DESIGN.

TELEMEDICINE DATABASE MANAGEMENT SYSTEM.

PROFESSOR: - Dr. JALAL OMER.

12/05/2022

NAME	ID
PRIJESH BHINGRADIYA	PHB210001
DARSH MAHESHKUMAR VAGHASIYA	DXV210035
HET DIPAKKUMAR PATEL	HDP210002
PRAJAY SANDIPKUMAR MEHTA	PXM210064

TABLE OF CONTENTS: -

INTRODUCTION: -

Telemedicine Database Management System is an online platform that will provide consultation to patients from certified doctors via telecommunications. The system will have an integrated database consisting of doctors, patients, chemists and medicines based on their location.

Our platform will provide different user interfaces based on stake holders use case. These details are stored in an organized and structured manner in a single database system.

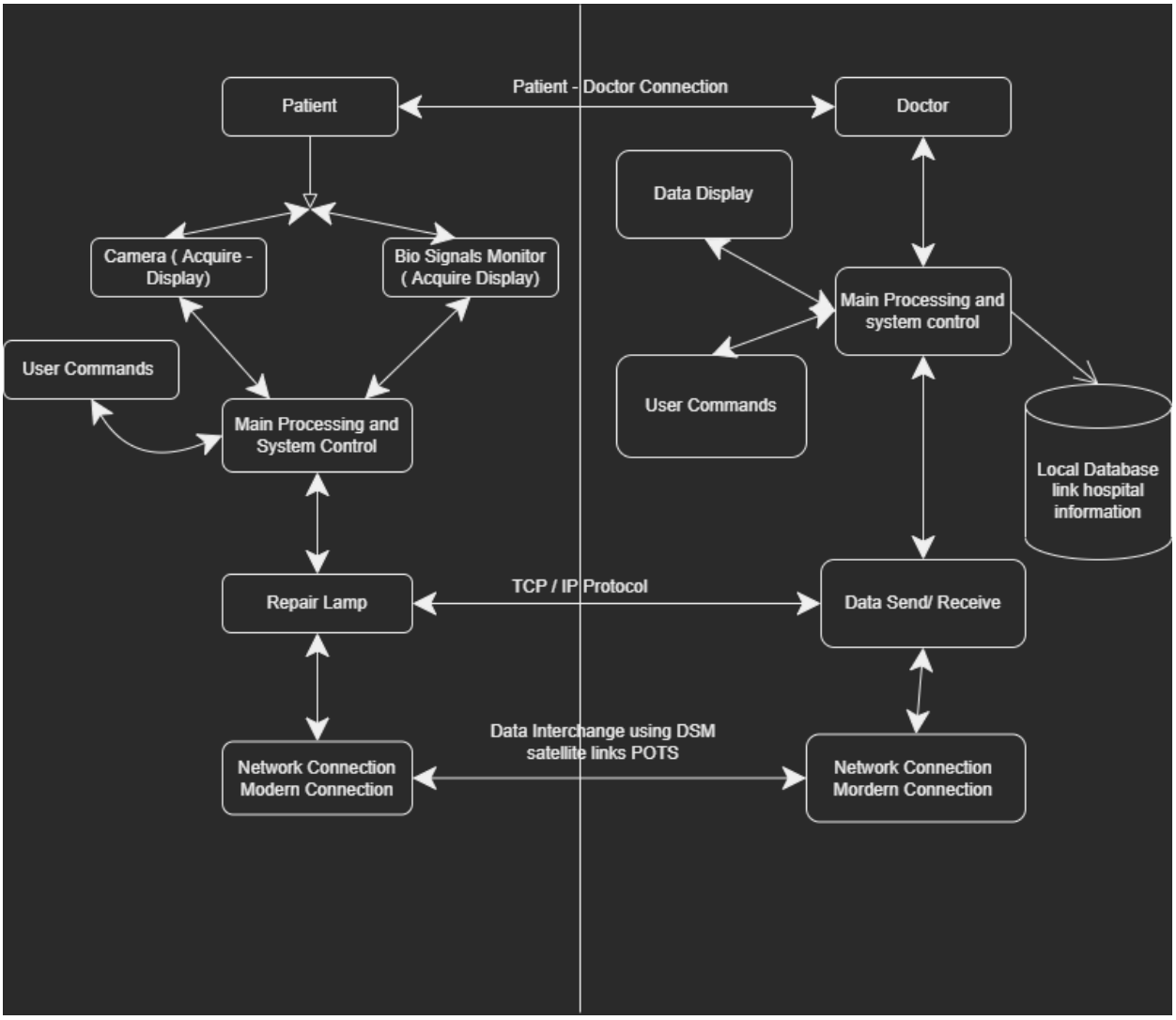
Our system will maintain the details of customers. This will include their Name, SSN, Date of Birth, address, email id, phone number. Separate table for their information regarding the options they opt in for appointment confirmation and brief history of ailments. Customers can buy membership to get an extra time slot for appointments, hidden time slots from normal customers and other benefits.

Doctors need to sign up to the doctor's side portal and access a dashboard where they can manage appointments, availability, payment receipts, profile, etc. Doctors will select the time slots when they are free and available. They can just switch on the online button during that time slot so they will be visible to customers when patients are booking the appointments online.

Once the patient provides information of ailment and selects the doctor they want to consult with, they will be prompted to pay a fee for consultation. After the patient finishes their transaction, they will receive mail or text message based on their preferences. The patient can call back again in the next half hour if there is a call drop due to any reason.

Once the doctor is done with the consultation, he will prescribe medicines to the patient. The list of medicines will be texted and emailed to the patient.

SYSTEM REQUIREMENTS: -



FUNCTIONAL REQUIREMENTS: -

Patients: -

- All patients should be able to register themselves as a new patient and create their account.
- The system will allow the patients to log in and will verify the used id and password.
- The system will have separate modules for different types of users i.e., patients, doctors.

- All patients should be able to view their profile.
- All patients should be allowed to modify their Telemedicine service.
- All patients should be able to view/book/cancel their appointment.
- All patients should be able to see their previous appointments.
- The system will allow users to update and reset their password.
- The system will allow the patients to download their billing statements.

Doctors: -

- The system will allow the doctors to log in and will verify the used id and password.
- The system will allow the doctors to read the patients' records.
- They should be able to view the list of appointments and their details and reschedule them if needed.
- They should be able to update their availability.
- The system will allow doctors to add prescriptions for the patient.

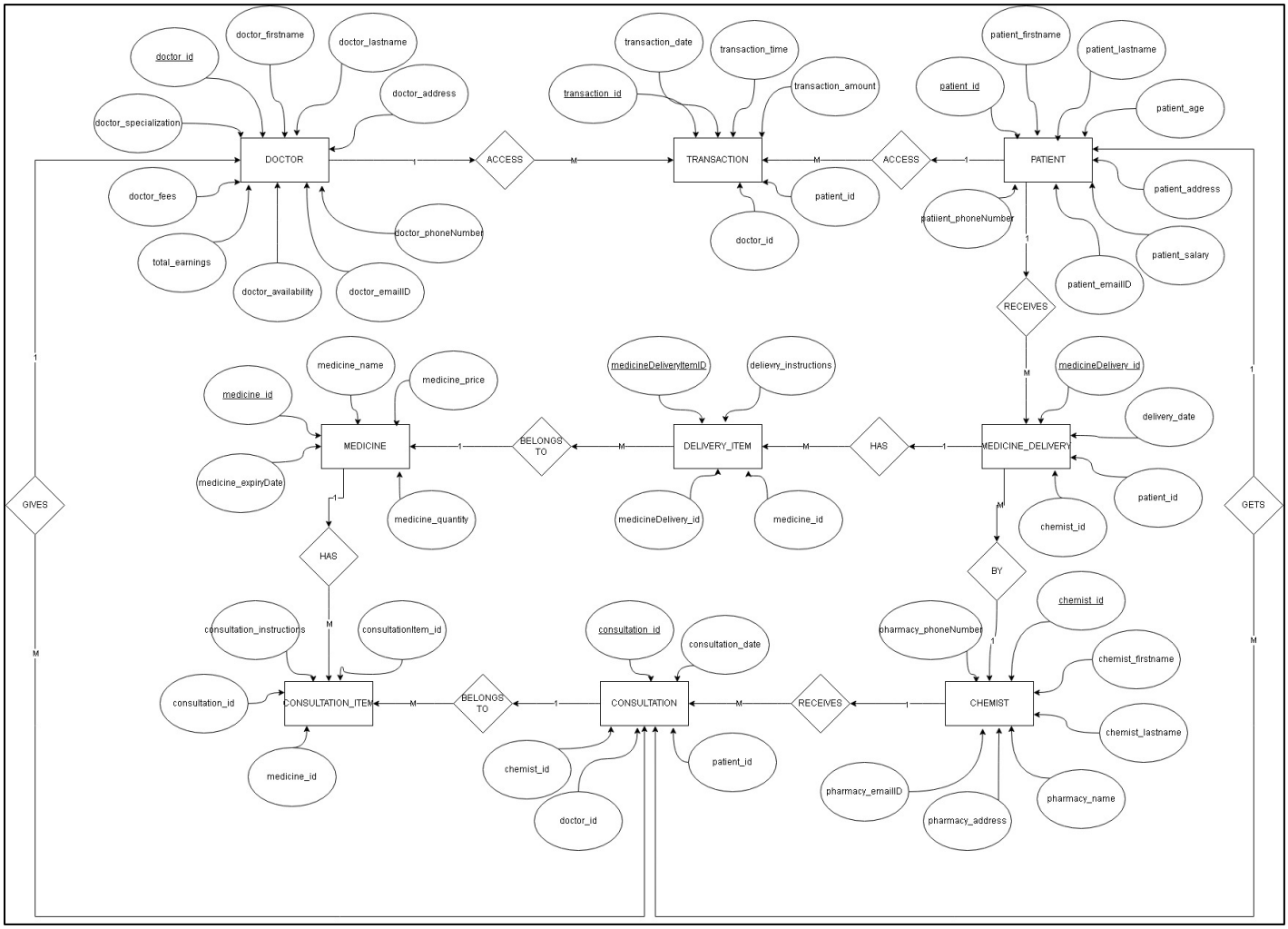
Administration: -

- They should be able to view payment receipts of appointments.
- They should be able to generate payment-related reports.

Non-Functional Requirements: -

- The system is portable, so it can be used across various operating systems.
- The system requires authorization for accessing the personal records of the patient. Only a patient and his/her doctor can access them.
- The application is easy to use and understand and gives the user the flexibility to reserve/ cancel or reschedule an appointment.
- The system requires authentication before accessing the program by providing the user id and password.

ER DIAGRAM: -


DOCTOR
doctor_id
doctor_firstname
doctor_lastname
doctor_address
doctor_specialization
doctor_fees
total_earnings
doctor_availability
doctor_emailID
doctor_phoneNumber
ACCESS
TRANSACTION
transaction_id
transaction_date
transaction_time
transaction_amount
patient_id
doctor_id
ACCESS
PATIENT
patient_id
patient_firstname
patient_lastname
patient_age
patient_address
patient_salary
patient_emailID
patient_phoneNumber
RECEIVES
GIVES
GETS
MEDICINE
medicine_id
medicine_name
medicine_price
medicine_expiryDate
medicine_quantity
BELONGS TO
DELIVERY_ITEM
medicineDeliveryItemID
delievry_instructions
medicineDelivery_id
medicine_id
HAS
MEDICINE_DELIVERY
medicineDelivery_id
delivery_date
patient_id
chemist_id
HAS
BY
CONSULTATION_ITEM
consultation_id
consultation_instructions
consultationItem_id
medicine_id
BELONGS TO
CONSULTATION
consultation_id
consultation_date
chemist_id
doctor_id
patient_id
RECEIVES
CHEMIST
chemist_id
pharmacy_phoneNumber
chemist_firstname
chemist_lastname
pharmacy_emailID
pharmacy_address
pharmacy_name

LIST OF BUSINESS RULES AND INTEGRITY CONSTRAINT: -

Following will be the users of Telemedicine database management system: -

- Doctors
- Patients
- Chemists

Why doctors have concerns about important data

- Why Doctors who have registered with our system want to know at the end of the day how many consultations they provided, along with consultation information. As a result, our system will produce a report for the doctors that contains the data from the "consultation" table.
- At the end of the day, doctors would also like to know how many transactions they have completed. From "transaction," a report containing this data will be produced.
- Doctors are interested in learning their overall weekly earnings. Therefore, we will also be producing a report from "transaction" that contains this data.

Additionally, doctors want to search our database:

- To create, edit, or remove a doctor profile.
- The 'doctor' table is updated whenever a doctor adds, modifies, or deletes their profile.
- The doctor is only able to change their own profile because they must use their password to enter into their profiles in order to do so.
- To obtain patient information - Doctors can obtain patient information by reviewing the records in the "patient" table.
- Examining records from the "chemist" table will allow doctors to obtain information about chemists.

- Doctors can look at records from the "medicine" table to obtain information about medications by doing so.

Why Patients have concerns about crucial data

- Patients who use the Telemedicine database management system would like to know how many consultations they have had overall in a month as well as the specifics of those consultations. Our system will provide a monthly report for the patients that contains this data from the "consultation" table.
- Additionally, patients want to know how many transactions they've completed in the previous month. Therefore, our system will produce a monthly report for patients using data from the "transaction" table.

Patients would also like to search our database:

- To create, edit, or remove a patient profile.
- The 'patient' table is updated whenever a patient adds, modifies, or deletes their profile. One constraint over here is that the patient can only modify their own profile as they will need to login into their profiles with their password in order to update.
- To obtain information about doctors, patients can search through the data in the "doctor" table.
- Looking at records from the "chemist" table will allow patients to obtain information about chemists.
- Patients can look at data from the "medicine" table to find out more information about the drugs they are taking.

Why Chemists have major data questions

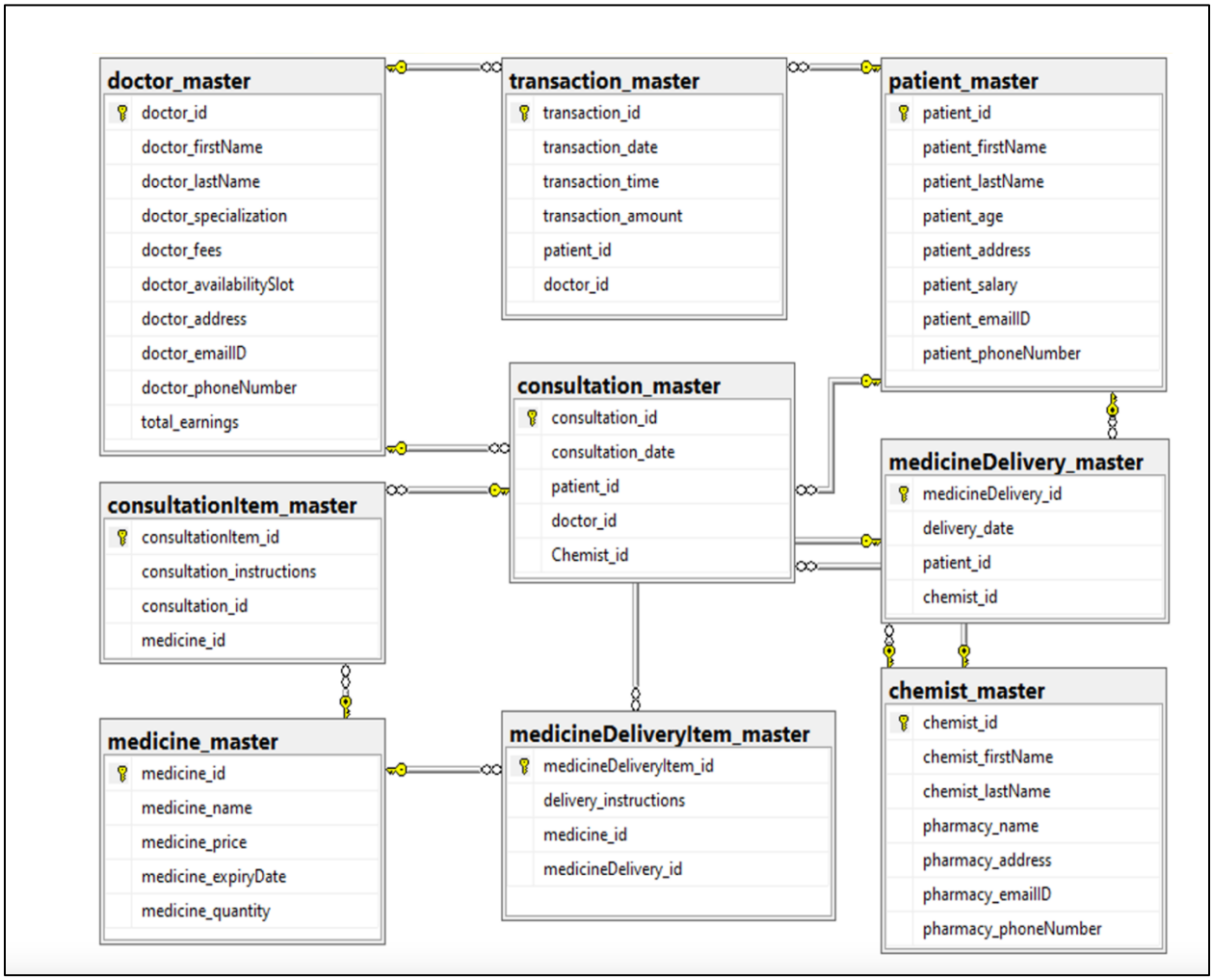
Why Chemists that have signed up for our system want to know how many medication deliveries they have made in a week. Therefore, our system will use the "medicineDelivery" table to generate a weekly report.

Additionally, chemists want to search our database:

- To create, edit, or remove a chemist profile.
- The 'chemist' table is updated whenever a chemist adds, modifies, or deletes their profile.
- One restriction on this site is that chemists can only change their own profiles because they must use their passwords to log into their accounts.

- To learn more about a specific doctor - A chemist can learn more about doctors by perusing the data in the "doctor" table. To get information related to a particular patient – Chemist can get information related to patients by looking at records from the ‘patient’ table.
- To obtain data on medicine stock - Chemists can examine records from the "medicine" database to obtain data on medications and their stock.
- To obtain information regarding consultations, chemists can do so by reviewing the records of the "consultation" table.

DATABASE SCHEMA: -



LOGICAL DATABASE SCHEMA: -

CREATE TABLE doctor (
doctor_id INT PRIMARY KEY,
doctor_firstName VARCHAR(30) NOT NULL,
doctor_lastName VARCHAR(30) NOT NULL,
doctor_specialization VARCHAR(30),
doctor_fees INT,
total_earnings INT,
doctor_availability TIME,
doctor_address VARCHAR(50),
doctor_emailID VARCHAR(30),
doctor_phoneNumber INT);

```
CREATE TABLE patient (
patient_id INT PRIMARY KEY,
patient_firstName VARCHAR(30) NOT NULL,
patient_lastName VARCHAR(30) NOT NULL,
patient_age INT,
patient_address VARCHAR(50),
patient_salary INT,
patient_emailID VARCHAR(30),
patient_phoneNumber INT
);

CREATE TABLE chemist(
chemist_id INT PRIMARY KEY,
chemist_firstName VARCHAR(30) NOT NULL,
chemist_lastName VARCHAR(30) NOT NULL,
pharmacy_name VARCHAR(30) NOT NULL,
pharmacy_address VARCHAR(50) NOT NULL,
pharmacy_emailID VARCHAR(30) NOT NULL,
pharmacy_phoneNumber INT
);

CREATE TABLE medicine (
medicine_id INT PRIMARY KEY,
medicine_name VARCHAR(30) NOT NULL,
medicine_price INT,
medicine_expiryDate DATE,
medicine_quantity INT
);

CREATE TABLE transaction1 (
transaction_id INT PRIMARY KEY,
transaction_date DATE,
transaction_time TIME,
transaction_amount INT,
patient_id INT,
doctor_id INT,
FOREIGN KEY (patient_id) REFERENCES patient(patient_id),
FOREIGN KEY (doctor_id) REFERENCES doctor(doctor_id)
);
```

```
CREATE TABLE consultation (
consultation_id INT PRIMARY KEY,
consultation_date DATE,
patient_id INT,
doctor_id INT,
chemist_id INT,
FOREIGN KEY (patient_id) REFERENCES patient(patient_id),
FOREIGN KEY (doctor_id) REFERENCES doctor(doctor_id),
FOREIGN KEY (chemist_id) REFERENCES chemist(chemist_id)
);

CREATE TABLE consultationItem (
consultationItem_id INT PRIMARY KEY,
consultation_instructions VARCHAR(50),
consultation_id INT,
medicine_id INT,
FOREIGN KEY (consultation_id) REFERENCES consultation(consultation_id),
FOREIGN KEY (medicine_id) REFERENCES medicine(medicine_id)
);

CREATE TABLE medicineDelivery (
medicineDelivery_id INT PRIMARY KEY,
delivery_date DATE,
patient_id INT,
chemist_id INT,
FOREIGN KEY (patient_id) REFERENCES patient(patient_id),
FOREIGN KEY (chemist_id) REFERENCES chemist(chemist_id)
);

CREATE TABLE deliveryItem (
medicineDeliveryItem_id INT PRIMARY KEY,
delivery_instructions VARCHAR(50),
medicine_id INT,
medicineDelivery_id INT,
FOREIGN KEY (medicine_id) REFERENCES medicine(medicine_id),
FOREIGN KEY (medicineDelivery_id) REFERENCES
medicineDelivery(medicineDelivery_id)
);
```

FUNCTIONAL DEPENDENCIES AND DATABASE NORMALIZATION: -

1. Doctor Table-

Doctor_ID	Doctor_FirstName	Doctor_LastName	Doctor_Address	Doctor_PhoneNumber	Doctor_EmailID	Doctor_Availability	Total_Earnings	Doctor_Fees	Doctor_Specialization

FDs:

Doctor_ID -> Doctor_FirstName, Doctor_LastName, Doctor_Address, Doctor_EmailID.

1NF: Doctor_PhoneNumber is a multi-valued attribute.

Normalizing Doctor into:

Doctor(Doctor_ID, Doctor_FirstName, Doctor_LastName, Doctor_Address, Doctor_EmailID, Doctor_Availability, Total_Earnings, Doctor_Fees, Doctor_Specialization) and

Doctor_Contact(Doctor_ID, Doctor_PhoneNumber)

There are no partial dependencies, so the table is in 2NF.

There are no transitive dependencies, so the table is in 3NF.

2. Patient Table-

Patient_ID	Patient_FirstName	Patient_LastName	Age	Address	Patient_Salary	Patient_PhoneNmber

FDs:

Patient_ID -> Patient_FirstName, Patient_LastName, Address.

1NF: Patient_PhoneNumber is a multi-valued attribute.

Normalizing Patient into:

Patient(Patient_ID, Patient_FirstName, Patient_LastName, Age, Address, Patient_salary) and

Patient_Contact(Patient_ID, Patient_PhoneNumber)

There are no partial dependencies, so the table is in 2NF.

There are no transitive dependencies, so the table is in 3NF.

3. Chemist Table-

Chemist_ID	Chemist_FirstName	Chemist_LastName	Pharmacy_Name	Pharmacy_Address	Pharmacy_EmailId	Pharmacy_PhoneNumber

FDs:

Chemist_ID -> Chemist_FirstName, Chemist_LastName, Pharmacy_Name, Pharmacy_Address.

1NF: Pharmacy_PhoneNumber is a multi-valued attribute.

Normalizing Chemist into:

Chemist(Chemist_ID, Chemist_FirstName, Chemist_LastName, Pharmacy_Address, Pharmacy_EmailID) and

Chemist_Contact(Chemist_ID, Pharmacy_PhoneNumber)

There are no partial dependencies, so the table is in 2NF.

There are no transitive dependencies, so the table is in 3NF.

4. Medicine Table-

Medicine_ID	Medicine_Name	Medicine_Price	Medicine_Quantity	Medicine_ExpiryDate

FDs:

Medicine_ID -> Medicine_Name, Medicine_Price, Medicine_Quantity, Medicine_ExpiryDate.

1NF: No multivalued attributes, so table is in 1NF.

2NF: No partial dependencies, so table is in 2NF.

3NF: No transitive dependencies, so table is in 3NF.

5. Transaction Table-

Transaction_ID	Transaction_Date	Transaction_Time	Transaction_Amount	Patient_ID	Doctor_ID

FDs:

Transaction_ID -> Transaction_Date, Transaction_Time, Transaction_Amount, Patient_ID, Doctor_ID.

1NF: No multivalued attributes, so table is in 1NF.

2NF: No partial dependencies, so table is in 2NF.

3NF: No transitive dependencies, so table is in 3NF.

6. Consultation Table-

Consultation_ID	Consultation_Date	Patient_ID	Doctor_ID	Chemist_ID

FDs:

Consultation_ID -> Consultation_Date, Patient_ID, Doctor_ID, Chemist_ID.

1NF: No multivalued attributes, so table is in 1NF.

2NF: No partial dependencies, so table is in 2NF.

3NF: No transitive dependencies, so table is in 3NF.

7. ConsultationItem Table-

ConsultationItem_ID	ConsultationItem_Instructions	Consultation_ID	Medicine_ID

FDs:

ConsultationItem_ID -> ConsultationItem_Instructions, Consultation_ID, Medicine_ID.

1NF: No multivalued attributes, so table is in 1NF.

2NF: No partial dependencies, so table is in 2NF.

3NF: No transitive dependencies, so table is in 3NF.

8. MedicineDelivery Table-

MedicineDelivery_ID	MedicineDelivery_Date	Patient_ID	Chemist_ID

FDs:

MedicineDelivery_ID -> MedicineDelivery_Date, Patient_ID, Chemist_ID.

1NF: No multivalued attributes, so table is in 1NF.

2NF: No partial dependencies, so table is in 2NF.

3NF: No transitive dependencies, so table is in 3NF.

9. MedicineDeliveryItem Table-

MedicineDeliveryItem_ID	Delivery_Instruction	Medicine_ID	MedicineDelivery_ID

FDs:

MedicineDeliveryItem_ID -> Delivery_Instruction, Medicine_ID, MedicineDelivery_ID.

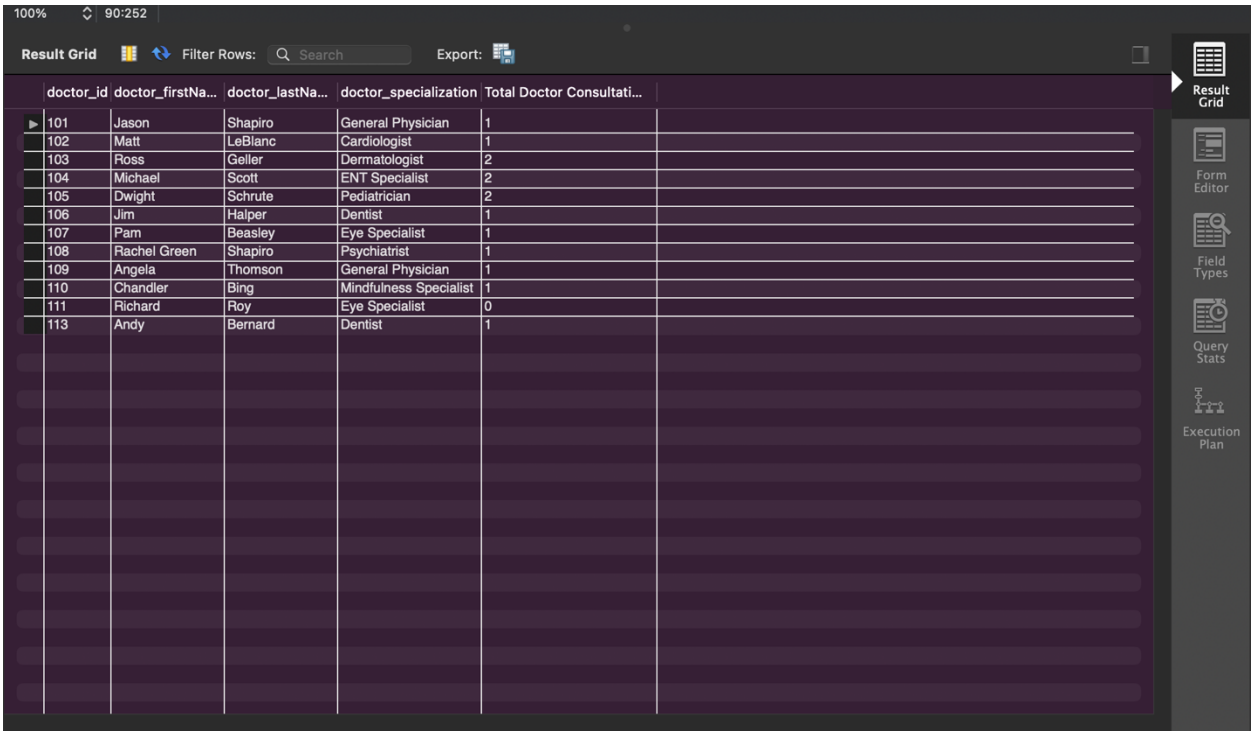
1NF: No multivalued attributes, so table is in 1NF.

2NF: No partial dependencies, so table is in 2NF.

3NF: No transitive dependencies, so table is in 3NF.

ADDITIONAL QUERIES AND VIEWS

```
/*Query to find out the total number of consultations provided by a doctor*/


SELECT dm.doctor_id, dm.doctor_firstName,
dm.doctor_lastName, dm.doctor_specialization,
COUNT(consultation_id) AS 'Total Doctor Consultations'

FROM doctor dm

LEFT OUTER JOIN consultation cm

ON dm.doctor_id=cm.doctor_id

GROUP BY dm.doctor_id, dm.doctor_firstName,
dm.doctor_lastName, dm.doctor_specialization

UNION

SELECT dm.doctor_id, dm.doctor_firstName,
dm.doctor_lastName, dm.doctor_specialization,
COUNT(consultation_id) AS 'Total Doctor Consultations'

FROM doctor dm

RIGHT OUTER JOIN consultation cm

ON dm.doctor_id=cm.doctor_id

GROUP BY dm.doctor_id, dm.doctor_firstName,
dm.doctor_lastName, dm.doctor_specialization;
```



doctor_id	doctor_firstNa...	doctor_lastNa...	doctor_specialization	Total Doctor Consultati...
101	Jason	Shapiro	General Physician	1
102	Matt	LeBlanc	Cardiologist	1
103	Ross	Geller	Dermatologist	2
104	Michael	Scott	ENT Specialist	2
105	Dwight	Schrute	Pediatrician	2
106	Jim	Halper	Dentist	1
107	Pam	Beasley	Eye Specialist	1
108	Rachel Green	Shapiro	Psychiatrist	1
109	Angela	Thomson	General Physician	1
110	Chandler	Bing	Mindfulness Specialist	1
111	Richard	Roy	Eye Specialist	0
113	Andy	Bernard	Dentist	1

```
/*Query to find out the total number of consultations
taken up by a patient*/
```

SELECT pm.patient_id, pm.patient_firstName, pm.patient_lastName, pm.patient_age, COUNT(consultation_id) AS 'Total Patient Consultations'

FROM patient pm

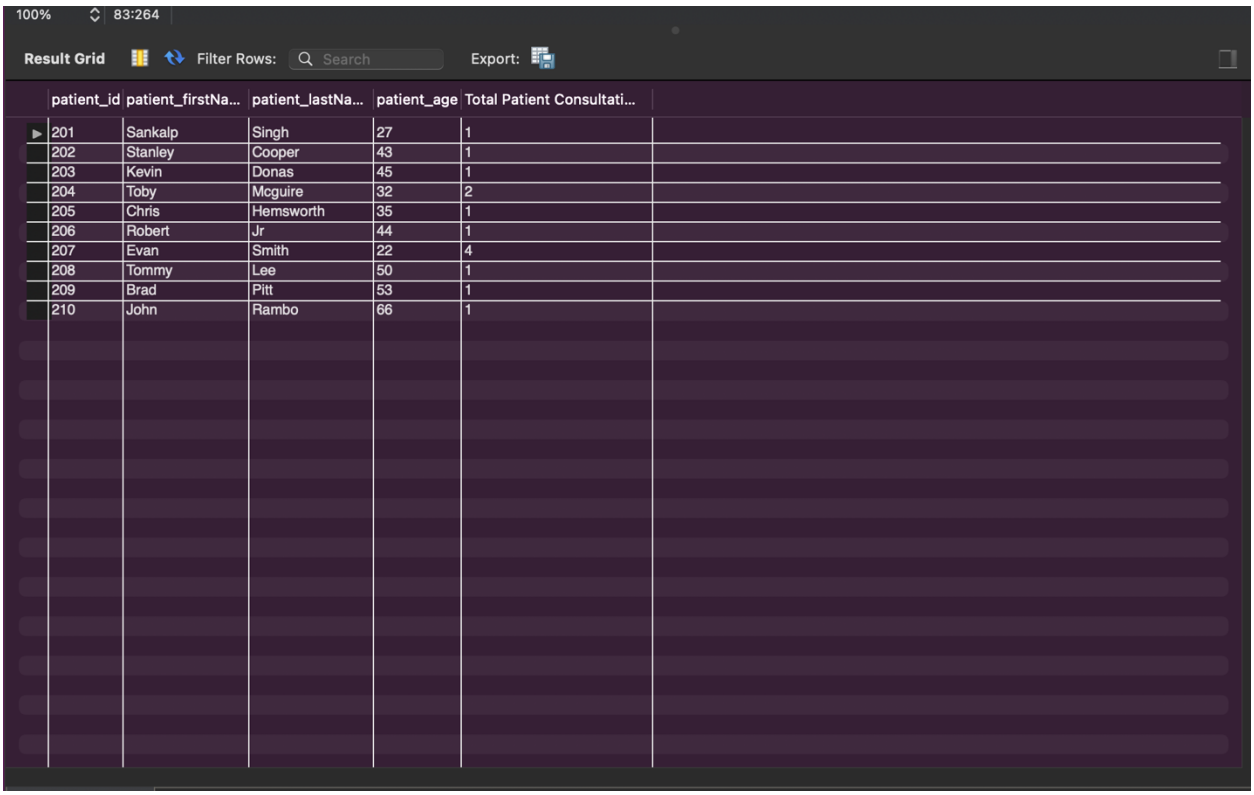
LEFT OUTER JOIN consultation cm

ON pm.patient_id=cm.patient_id

GROUP BY pm.patient_id, pm.patient_firstName, pm.patient_lastName, pm.patient_age

UNION

```
SELECT pm.patient_id, pm.patient_firstName, pm.patient_lastName,
pm.patient_age, COUNT(consultation_id) AS 'Total Patient Consultations'

FROM patient pm

RIGHT OUTER JOIN consultation cm

ON pm.patient_id=cm.patient_id

GROUP BY pm.patient_id, pm.patient_firstName, pm.patient_lastName,
pm.patient_age;
```



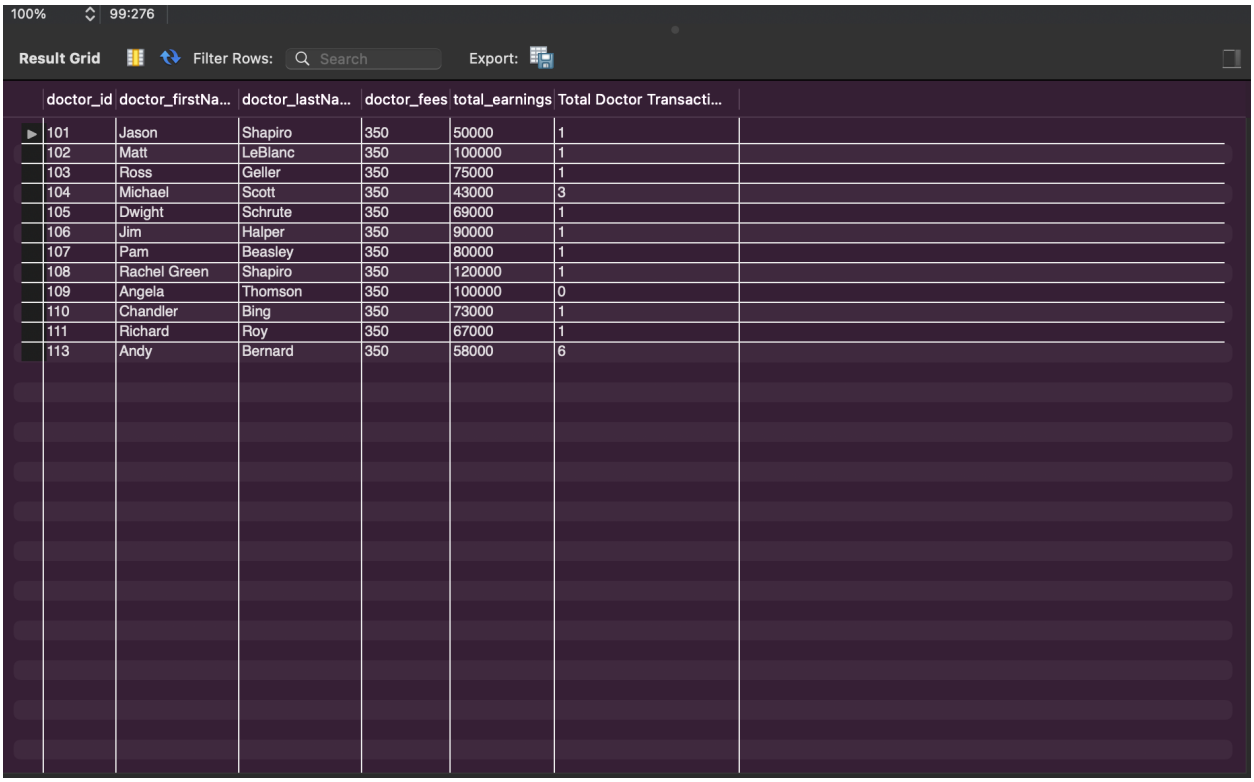
100% 83:264

Result Grid Filter Rows: Search Export:

patient_id	patient_firstNa...	patient_lastNa...	patient_age	Total Patient Consultati...
201	Sankalp	Singh	27	1
202	Stanley	Cooper	43	1
203	Kevin	Donas	45	1
204	Toby	Mcguire	32	2
205	Chris	Hemsworth	35	1
206	Robert	Jr	44	1
207	Evan	Smith	22	4
208	Tommy	Lee	50	1
209	Brad	Pitt	53	1
210	John	Rambo	66	1

```
/*Query to find out the total transactions & earnings for a
doctor*/
```

```
SELECT dm.doctor_id, dm.doctor_firstName, dm.doctor_lastName,
dm.doctor_fees, dm.total_earnings, COUNT(transaction_id) AS 'Total Doctor
Transactions'

FROM doctor dm

LEFT OUTER JOIN transaction1 tm

ON dm.doctor_id=tm.doctor_id

GROUP BY dm.doctor_id, dm.doctor_firstName, dm.doctor_lastName,
dm.doctor_fees, dm.total_earnings

UNION

SELECT dm.doctor_id, dm.doctor_firstName, dm.doctor_lastName,
dm.doctor_fees, dm.total_earnings, COUNT(transaction_id) AS 'Total Doctor
Transactions'

FROM doctor dm

RIGHT OUTER JOIN transaction1 tm

ON dm.doctor_id=tm.doctor_id

GROUP BY dm.doctor_id, dm.doctor_firstName, dm.doctor_lastName,
dm.doctor_fees, dm.total_earnings;
```



100% 99:276

Result Grid Filter Rows: Search Export:

doctor_id	doctor_firstNa...	doctor_lastNa...	doctor_fees	total_earnings	Total Doctor Transacti...
101	Jason	Shapiro	350	50000	1
102	Matt	LeBlanc	350	100000	1
103	Ross	Geller	350	75000	1
104	Michael	Scott	350	43000	3
105	Dwight	Schrute	350	69000	1
106	Jim	Halper	350	90000	1
107	Pam	Beasley	350	80000	1
108	Rachel Green	Shapiro	350	120000	1
109	Angela	Thomson	350	100000	0
110	Chandler	Bing	350	73000	1
111	Richard	Roy	350	67000	1
113	Andy	Bernard	350	58000	6

```
/*Query to find out the total transactions performed by a
patient*/
```

SELECT pm.patient_id, pm.patient_firstName, pm.patient_lastName, pm.patient_age, COUNT(transaction_id) AS 'Total Patient Transactions'

FROM patient pm

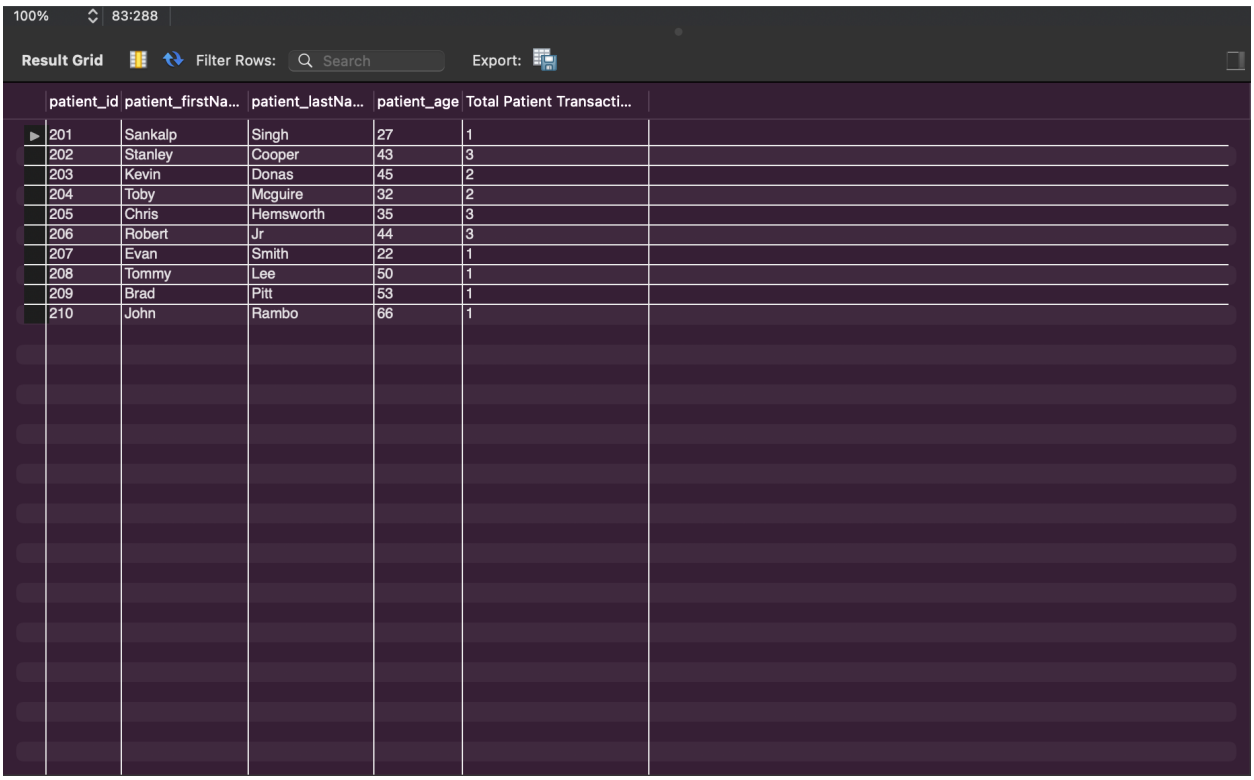
LEFT OUTER JOIN transaction1 tm

ON pm.patient_id=tm.patient_id

GROUP BY pm.patient_id, pm.patient_firstName, pm.patient_lastName, pm.patient_age

UNION

```
SELECT pm.patient_id, pm.patient_firstName, pm.patient_lastName,
pm.patient_age, COUNT(transaction_id) AS 'Total Patient Transactions'

FROM patient pm

RIGHT OUTER JOIN transaction1 tm

ON pm.patient_id=tm.patient_id

GROUP BY pm.patient_id, pm.patient_firstName, pm.patient_lastName,
pm.patient_age;
```



100% 83:288

Result Grid Filter Rows: Search Export:

patient_id	patient_firstNa...	patient_lastNa...	patient_age	Total Patient Transacti...
201	Sankalp	Singh	27	1
202	Stanley	Cooper	43	3
203	Kevin	Donas	45	2
204	Toby	Mcguire	32	2
205	Chris	Hemsworth	35	3
206	Robert	Jr	44	3
207	Evan	Smith	22	1
208	Tommy	Lee	50	1
209	Brad	Pitt	53	1
210	John	Rambo	66	1

```
/*Query to find out total medicine deliveries carried out by
a chemist*/


SELECT cm.chemist_id, cm.chemist_firstName, cm.chemist_lastName,
cm.pharmacy_name, cm.pharmacy_address, COUNT(medicineDelivery_id) AS
'Total Medicine Deliveries'

FROM chemist cm

LEFT OUTER JOIN medicineDelivery mdm ON cm.chemist_id=mdm.chemist_id

GROUP BY cm.chemist_id, cm.chemist_firstName, cm.chemist_lastName,
cm.pharmacy_name, cm.pharmacy_address

UNION

SELECT cm.chemist_id, cm.chemist_firstName, cm.chemist_lastName,
cm.pharmacy_name, cm.pharmacy_address, COUNT(medicineDelivery_id) AS
'Total Medicine Deliveries'

FROM chemist cm

RIGHT OUTER JOIN medicineDelivery mdm ON
cm.chemist_id=mdm.chemist_id

GROUP BY cm.chemist_id, cm.chemist_firstName, cm.chemist_lastName,
cm.pharmacy_name, cm.pharmacy_address;
```

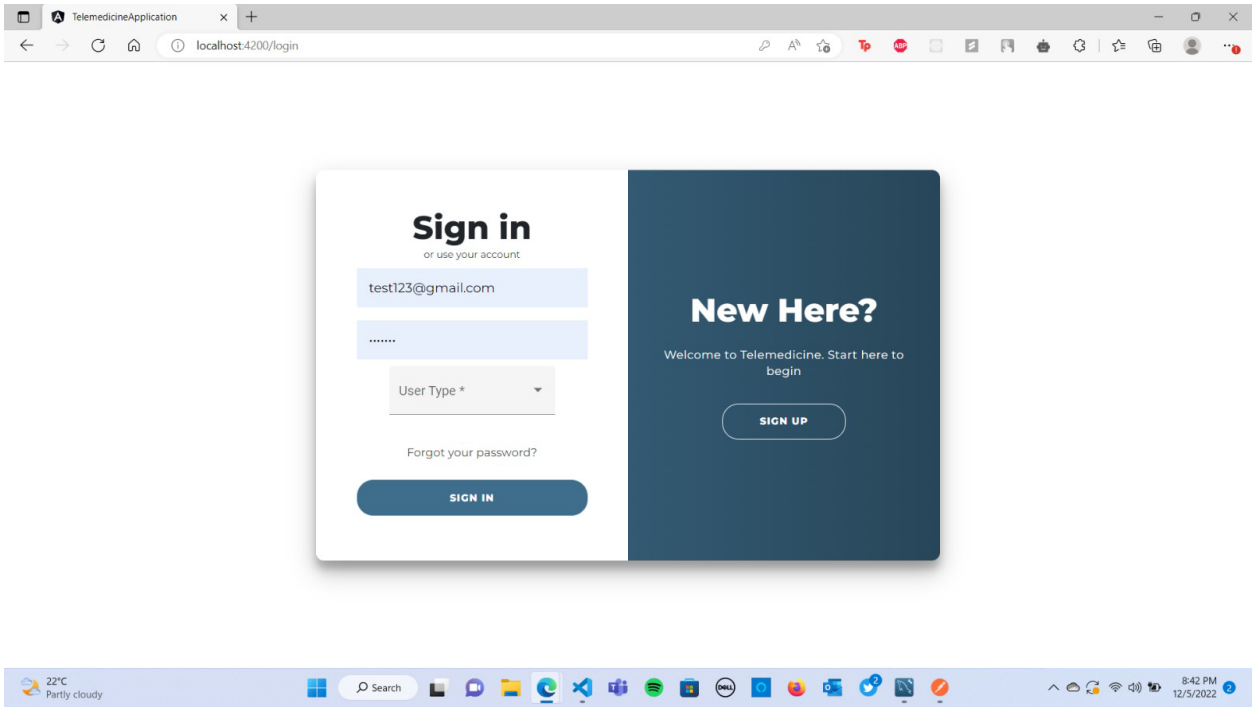
100% 1:299

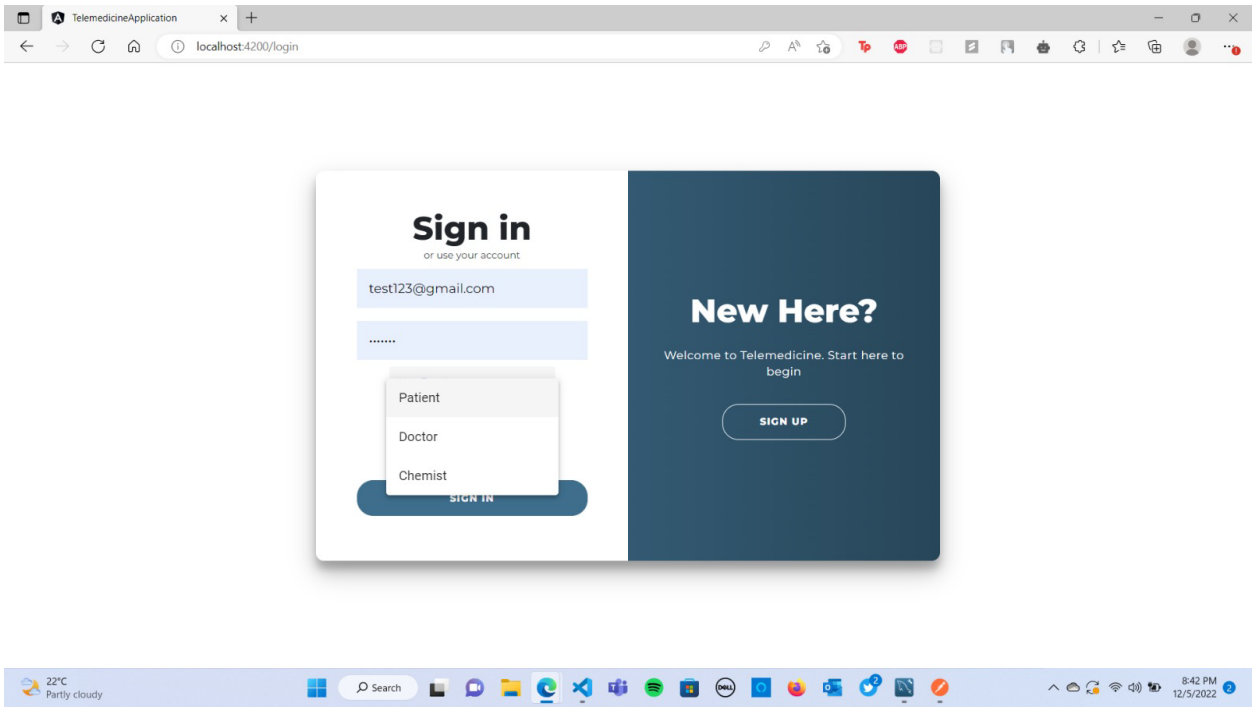
Result Grid Filter Rows: Search Export:

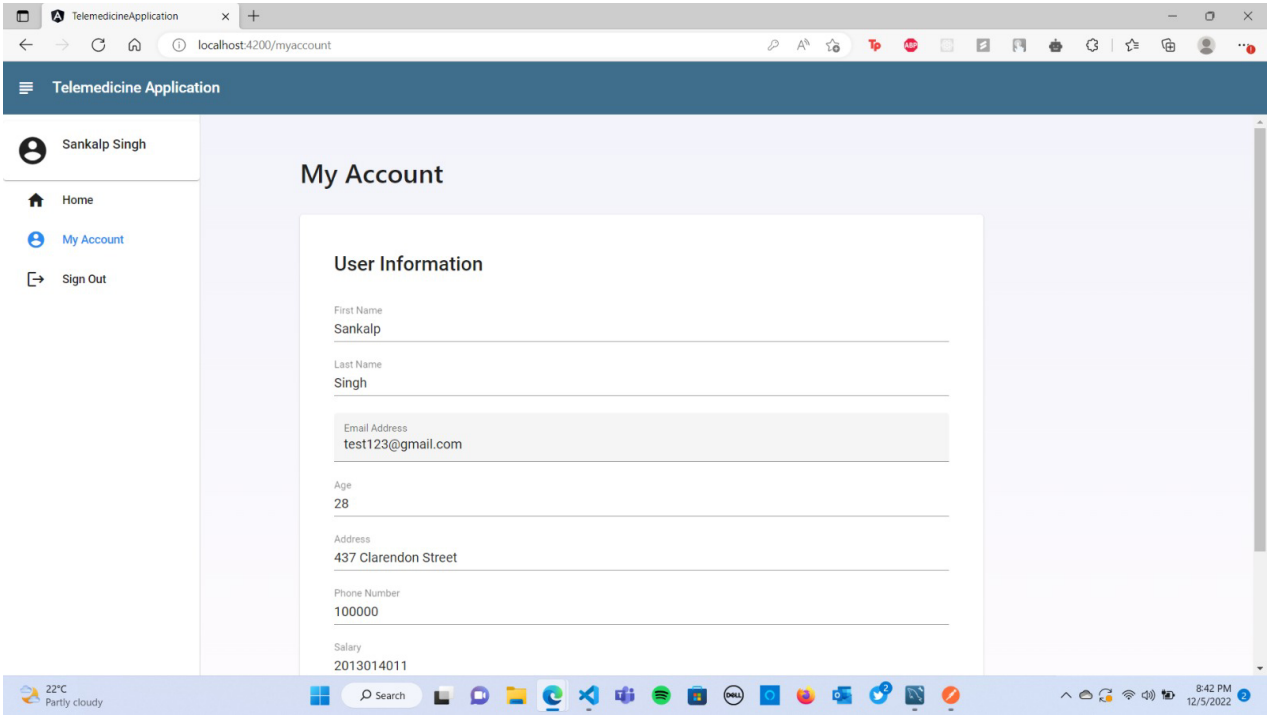
chemist_id	chemist_firstNa...	chemist_lastNa...	pharmacy_name	pharmacy_addre...	Total Medicine Deliver...
301	Katie	Holmes	CVS Pharmacy	Marshall Street	1
302	Tom	Cruise	Chase Pharmacy	Ackerman Street	1
303	Barney	Stinson	Jet Pharmacy	Lancaster Street	1
304	Lily	Dawson	Blue Pharmacy	Clarendon Street	1
305	Robin	Dannings	Green Pharmacy	Euclid Street	3
306	Ted	Mosby	Yellow Pharmacy	Ostrom Street	1
307	Marshall	Anderson	Red Cross Pharmacy	Comstock Street	4
308	Kevin	Donas	Health Plus Pharmacy	Gilbert Street	1
309	Margeret	Robbie	Trust Pharmacy	Westcott Street	1
310	James	Blunt	CVS Pharmacy	Lyra Street	1

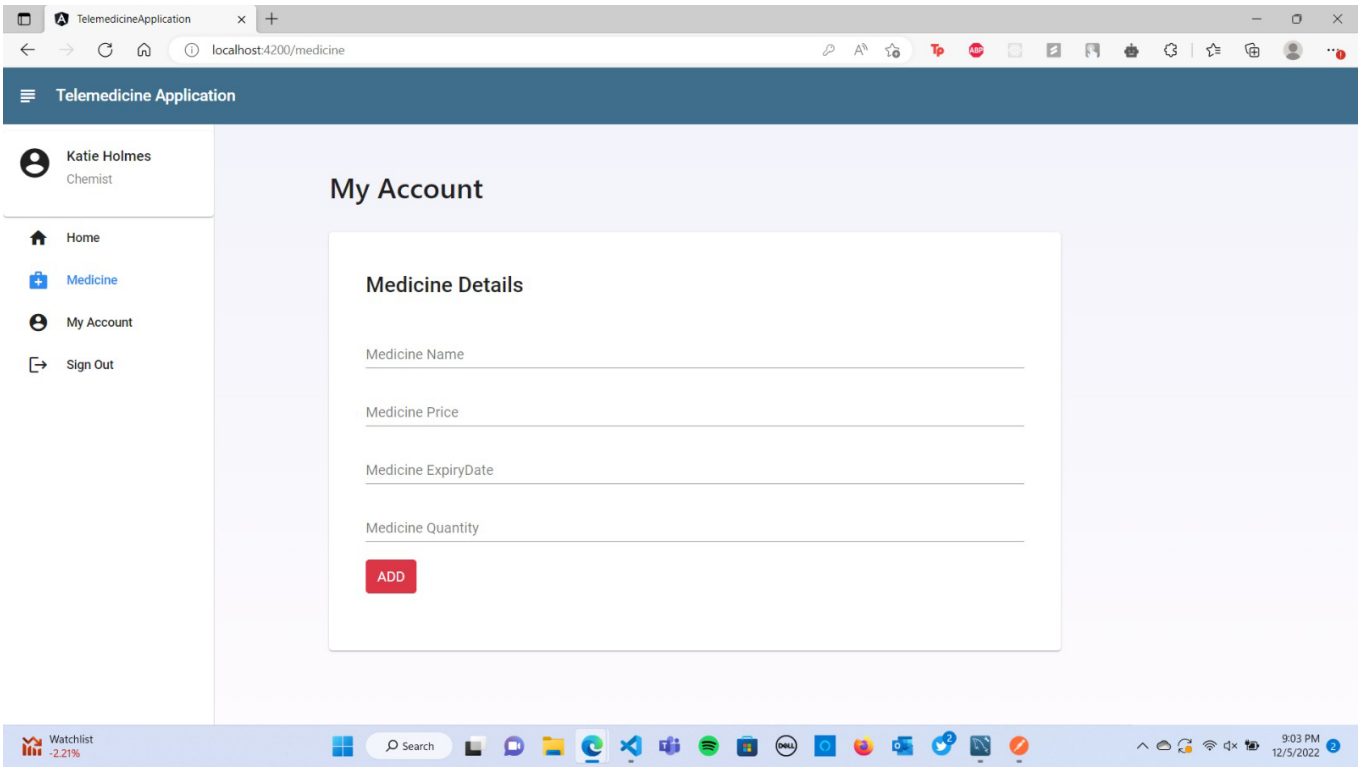
THE DATABASE SYSTEM / FRONTEND: -

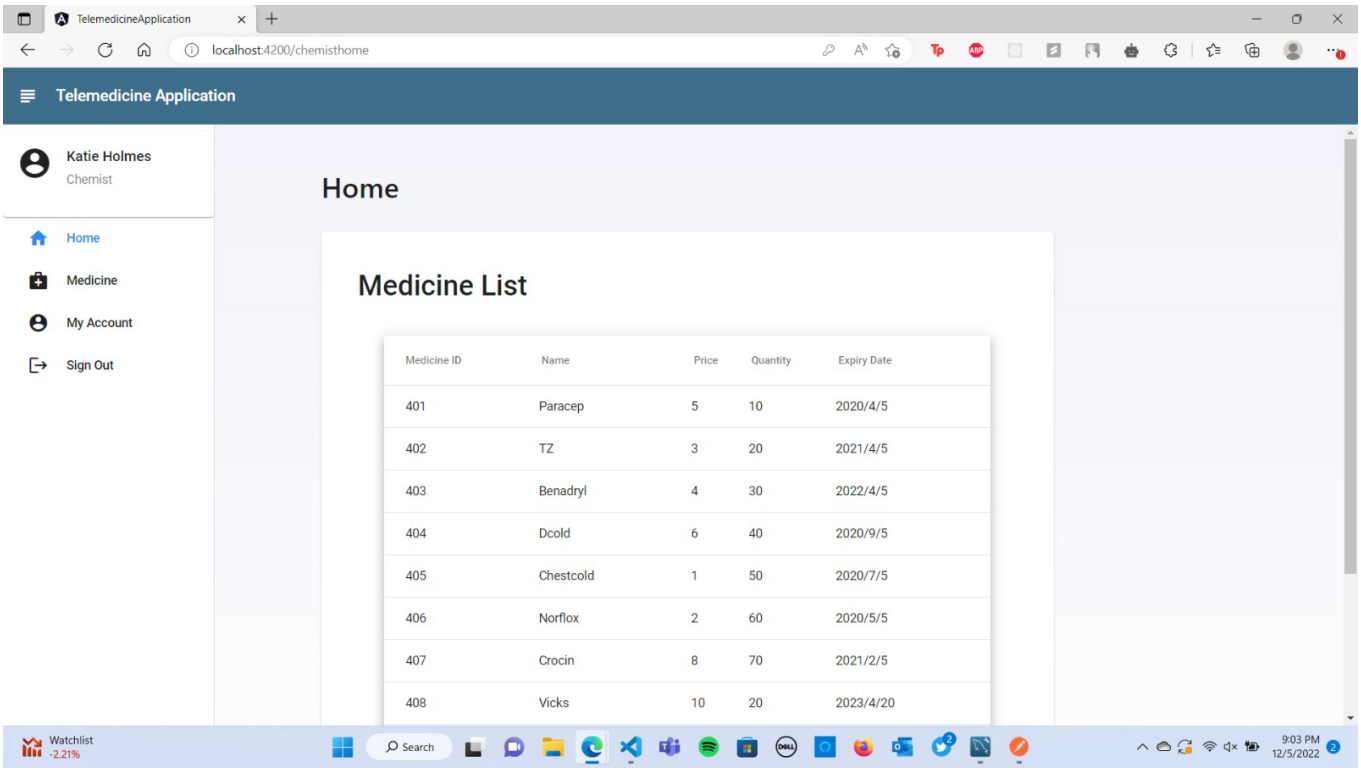
Click on the role type to get the login page specific to that role. The login interface for all the roles is the same as shown below.

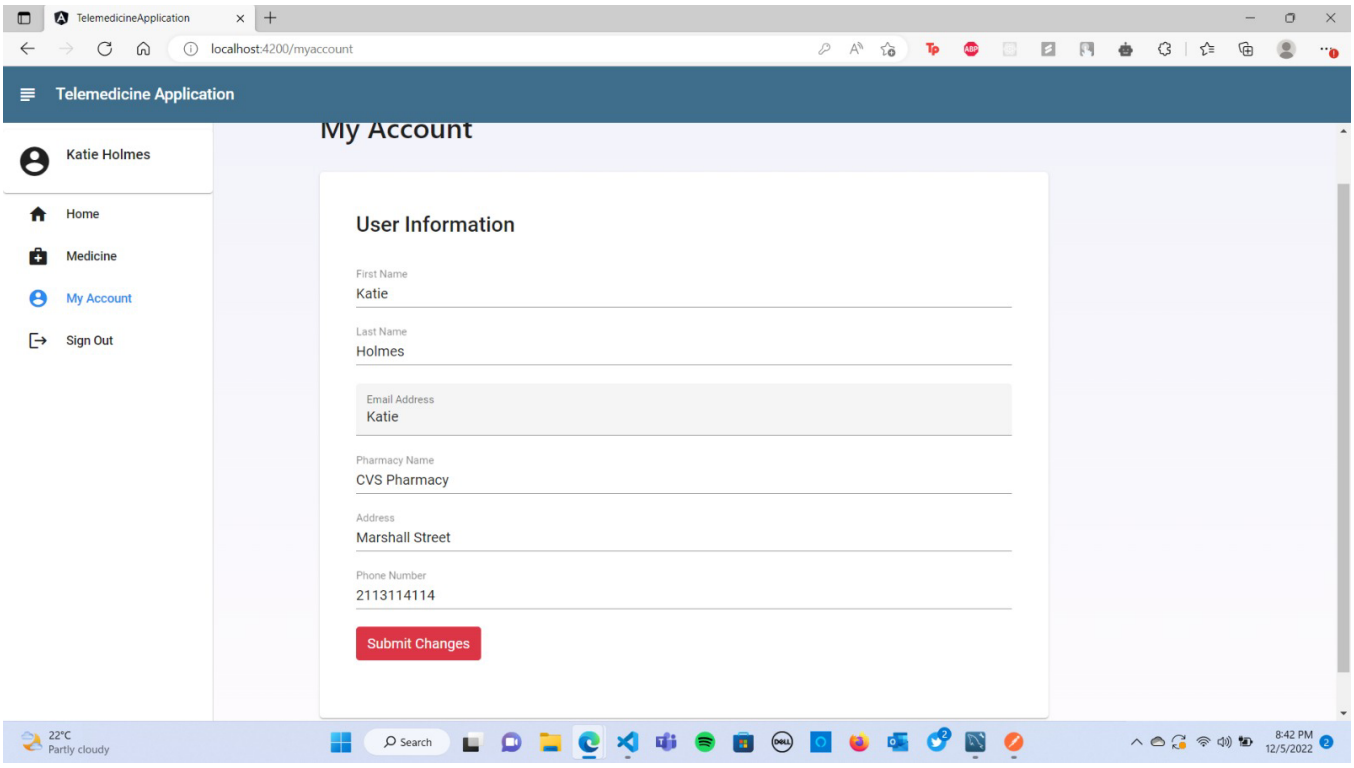


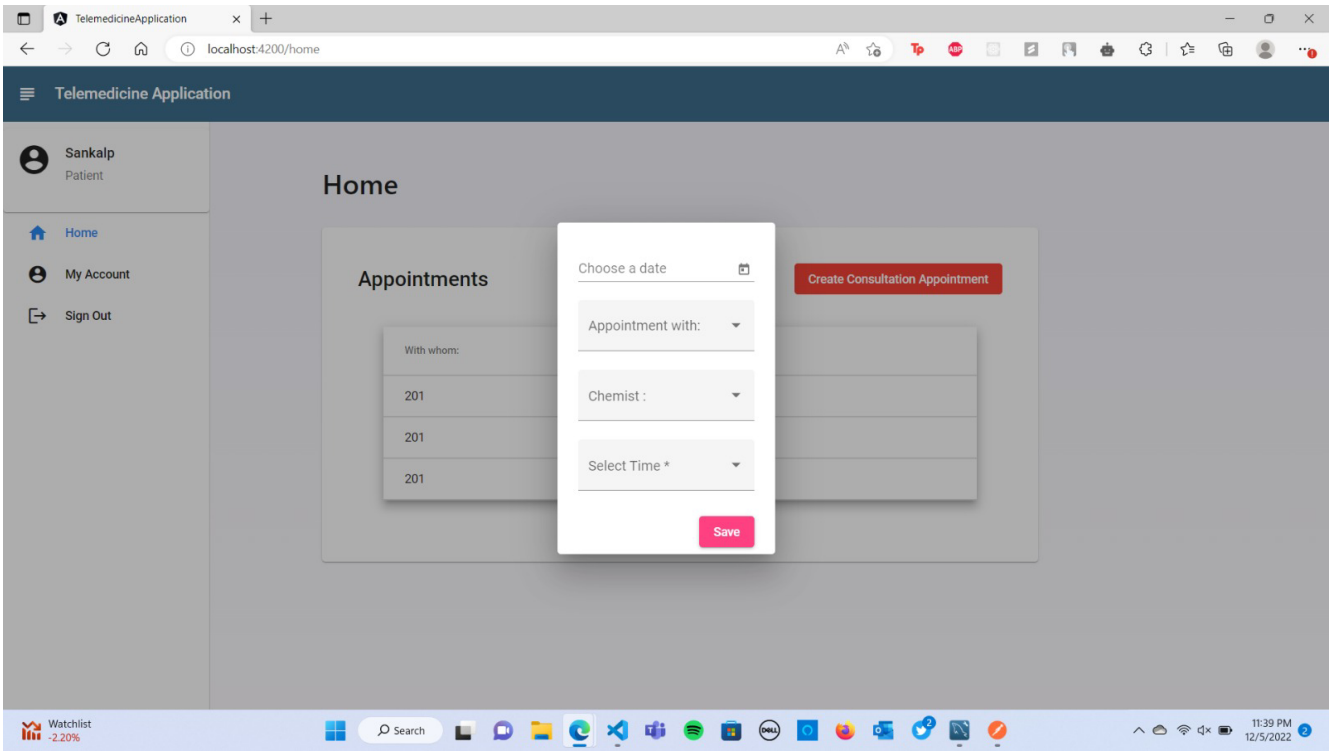

TelemedicineApplication
localhost:4200/login
Sign in
or use your account
test123@gmail.com
.....
Patient
Doctor
Chemist
SIGN IN
New Here?
Welcome to Telemedicine. Start here to begin
SIGN UP
22°C
Partly cloudy
Search
8:42 PM
12/5/2022

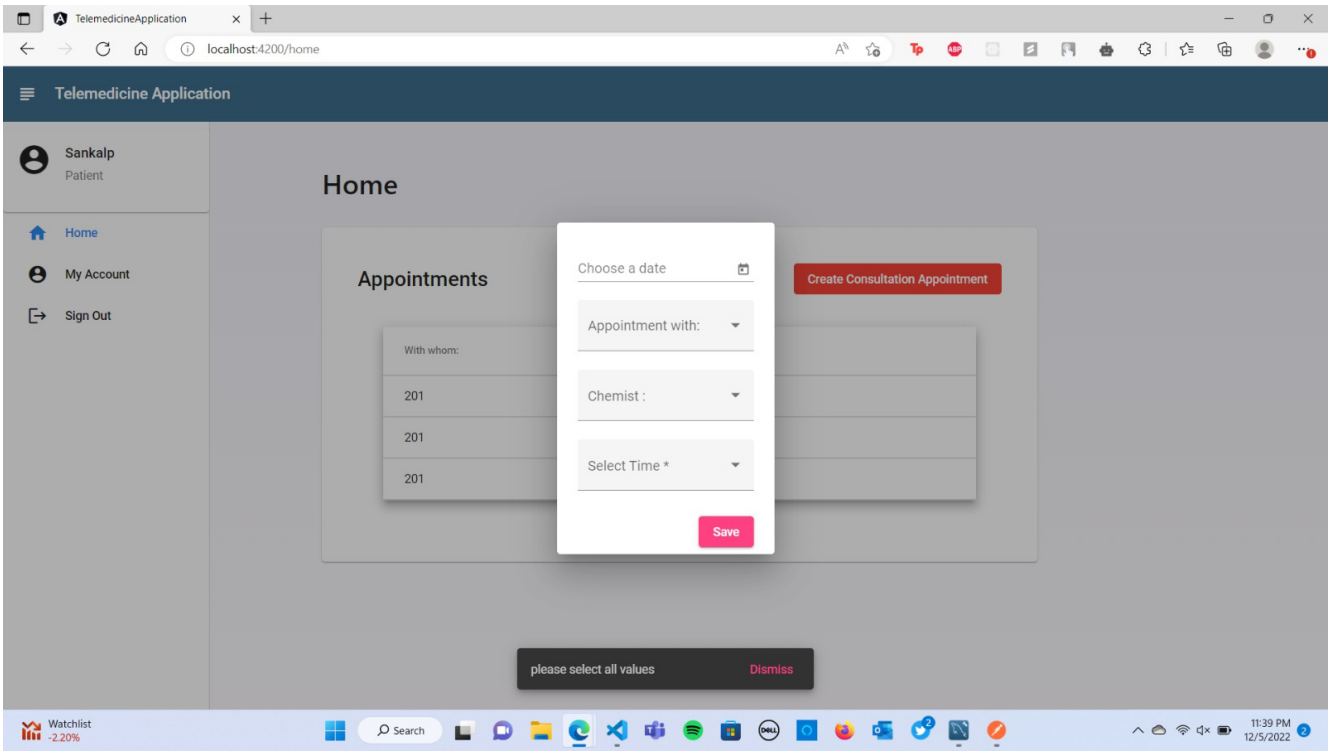

TelemedicineApplication
localhost:4200/myaccount
Telemedicine Application
Sankalp Singh
Home
My Account
Sign Out
My Account
User Information
First Name
Sankalp
Last Name
Singh
Email Address
test123@gmail.com
Age
28
Address
437 Clarendon Street
Phone Number
100000
Salary
2013014011
22°C
Partly cloudy
Search
8:42 PM
12/5/2022


TelemedicineApplication
localhost:4200/medicine
Telemedicine Application
Katie Holmes
Chemist
Home
Medicine
My Account
Sign Out
My Account
Medicine Details
Medicine Name
Medicine Price
Medicine ExpiryDate
Medicine Quantity
ADD
Watchlist
-2.21%
Search
9:03 PM
12/5/2022


TelemedicineApplication
localhost:4200/chemisthome
Telemedicine Application
Katie Holmes
Chemist
Home
Medicine
My Account
Sign Out
Home
Medicine List
Medicine ID
Name
Price
Quantity
Expiry Date
401 Paracep 5 10 2020/4/5
402 TZ 3 20 2021/4/5
403 Benadryl 4 30 2022/4/5
404 Dcold 6 40 2020/9/5
405 Chestcold 1 50 2020/7/5
406 Norflox 2 60 2020/5/5
407 Crocin 8 70 2021/2/5
408 Vicks 10 20 2023/4/20
Watchlist
-2.21%
Search
9:03 PM
12/5/2022


TelemedicineApplication
localhost:4200/myaccount
Telemedicine Application
Katie Holmes
Home
Medicine
My Account
Sign Out
My Account
User Information
First Name
Katie
Last Name
Holmes
Email Address
Katie
Pharmacy Name
CVS Pharmacy
Address
Marshall Street
Phone Number
2113114114
Submit Changes
22°C
Partly cloudy
Search
8:42 PM
12/5/2022


TelemedicineApplication
localhost:4200/home
Telemedicine Application
Sankalp
Patient
Home
My Account
Sign Out
Home
Appointments
Choose a date
Create Consultation Appointment
Appointment with:
With whom:
201
201
201
Chemist :
Select Time *
Save
Watchlist
-2.20%
Search
11:39 PM
12/5/2022


TelemedicineApplication
localhost:4200/home
Telemedicine Application
Sankalp
Patient
Home
My Account
Sign Out
Home
Appointments
Choose a date
Create Consultation Appointment
Appointment with:
With whom:
201
201
201
Chemist :
Select Time *
Save
please select all values
Dismiss
Watchlist
-2.20%
Search
11:39 PM
12/5/2022

CONCLUSION AND FUTURE WORK

- This telemedicine project was built using Angular, Node, Css, HTML, JavaScript, NodeJS and MySQL.
- MySQL is used to design the database where the whole data is stored, and Java swing is used for the front end making it very responsive.
- NodeJS is used for making the back end and the connection to the database given by using mysqli.
- For future work on this project, there can be more functionalities added like a pharmaceuticals database which can also provide the medicines for the prescription that is given by the consulted doctor. Which can take the order for the prescription and deliver it to the respective destination.
- We can improve the design of the GUI made using React while adding functionalities to the system.

REFERENCES

- www.geekforgeeks.com
- www.W3school.com
- www.javapoint.com

APPENDIX

https://github.com/prijeshb/DBProject/tree/master/Telemedicine-Application