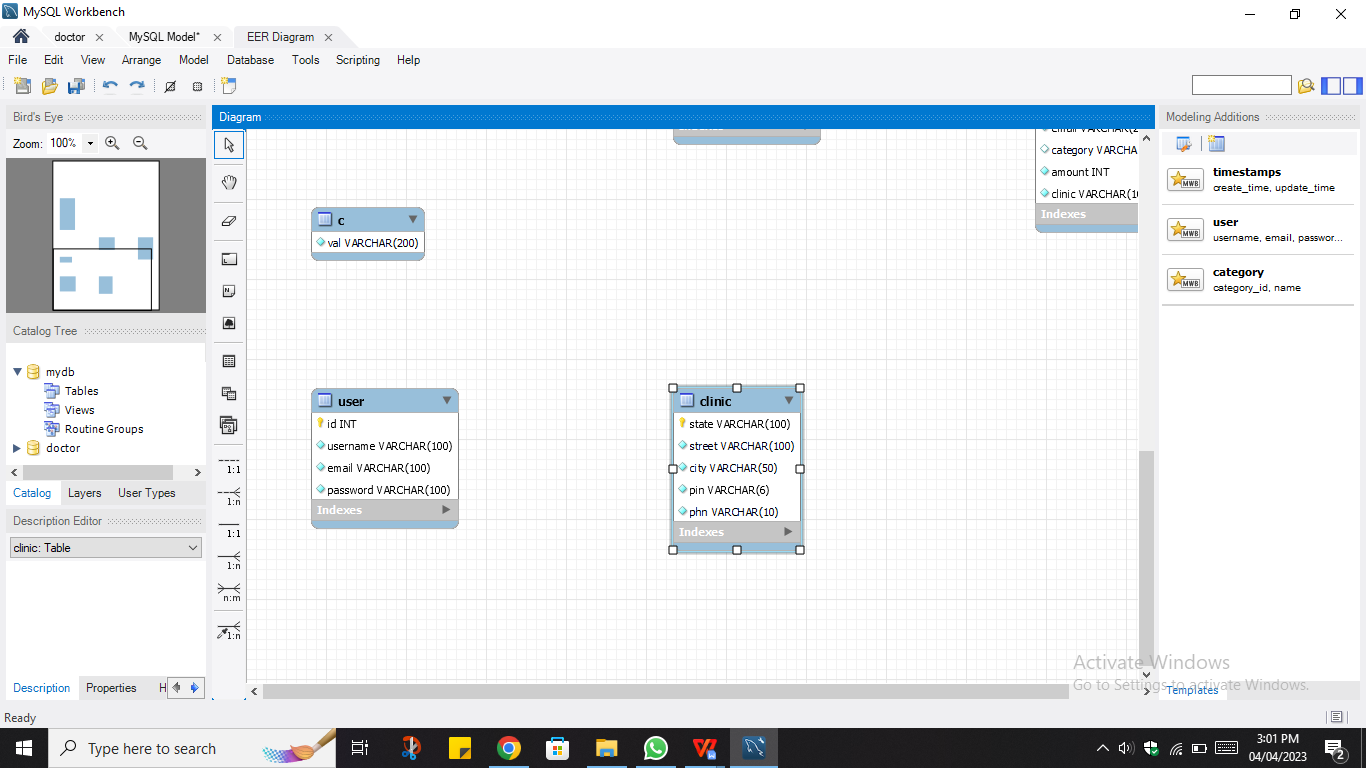
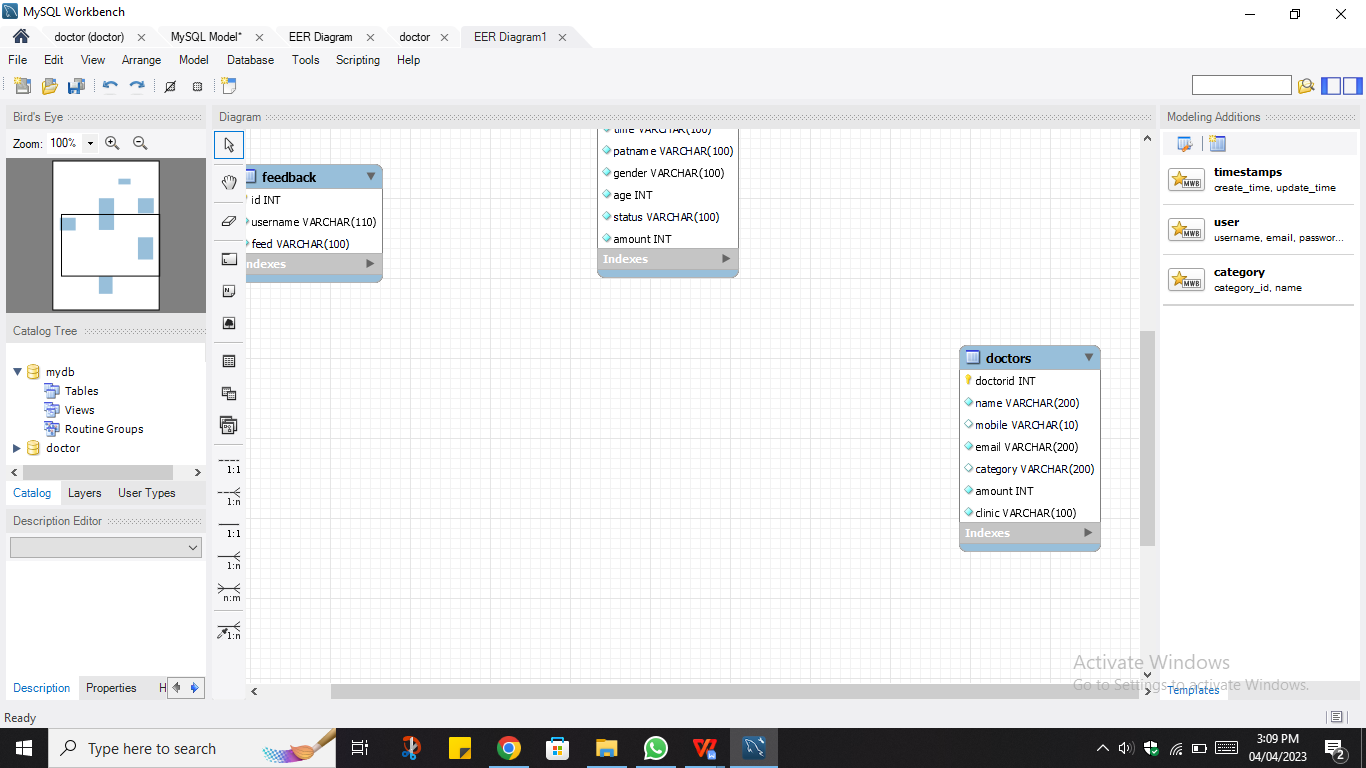
**DATABASE DESIGN**

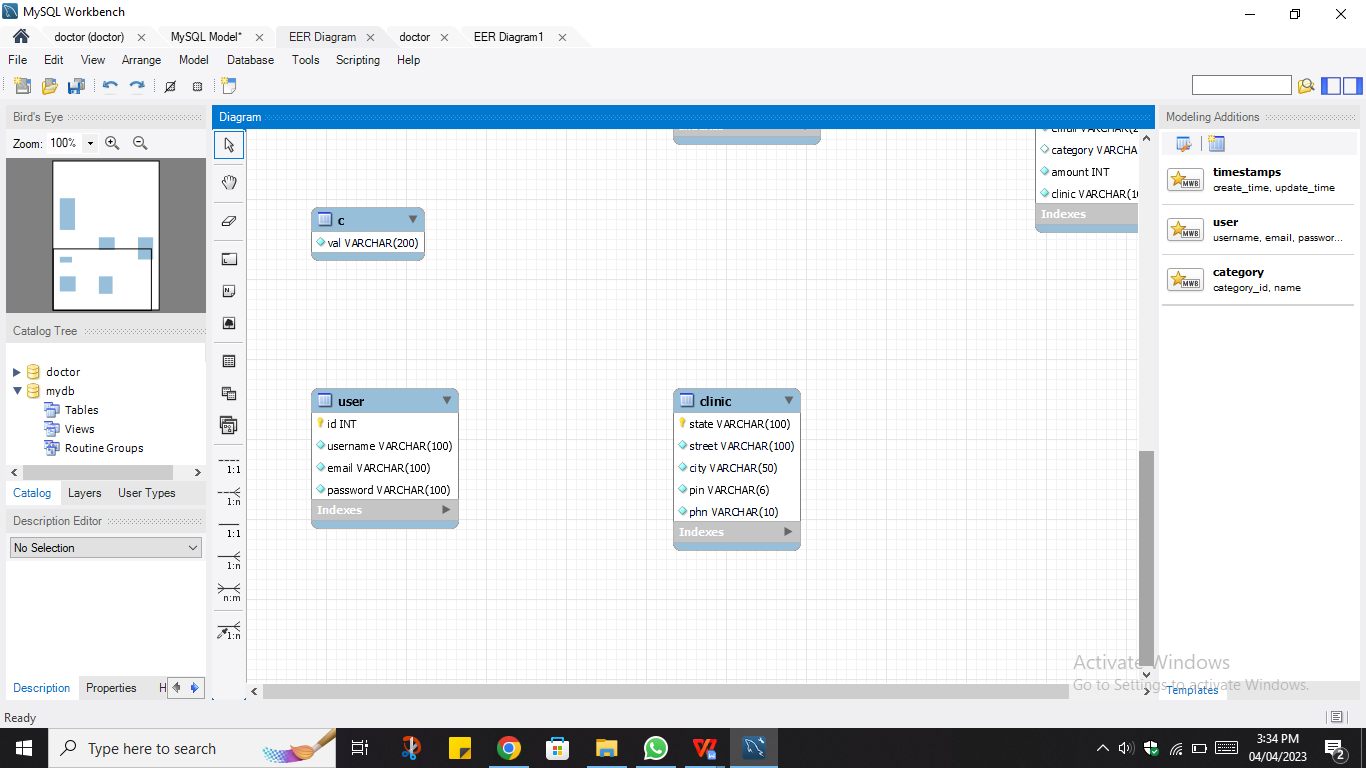
In my Doctor Appointment Booking System the database consists of 5 tables: doctors, booking, user, feedback and clinic. Lets go through each of then individually.

User Table:

* id(int): This is a primary key of the table which auto increments itself when a new user registers.
* Username(varchar): This stores the username.
* Email(varchar): This field is a unique field and it stores the email id of the users.
* Password(varchar): This field stores the password of the users.

Doctors Table:

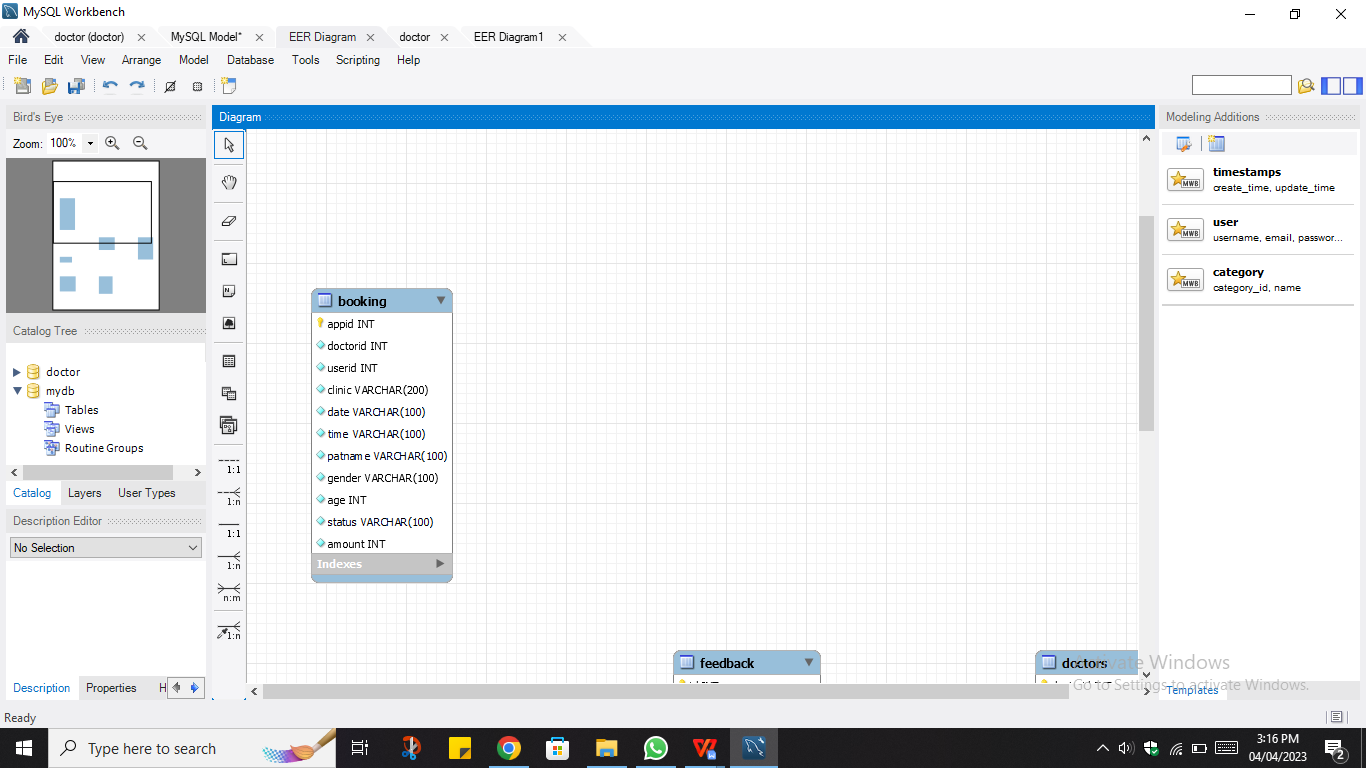
* doctorid(int): This is primary key of the table which auto increments its self when a new doctor is added by the admin.
* name(varchar): This stores the doctor name.
* mobile(varchar): This field is unique. This stores the contact no. of the doctor.
* email(varchar): This field is also unique. This stores the email id of the doctor in case he is not available on mobile.
* Category(varchar): This field stores information about the doctors educational degree and his specialization.
* Amount(int): This field stores the appointment fee of the doctor.
* Clinic(varchar): As in our system we have clinics in 3 cities so this field stores the name of the clinic area to which the doctor visits.

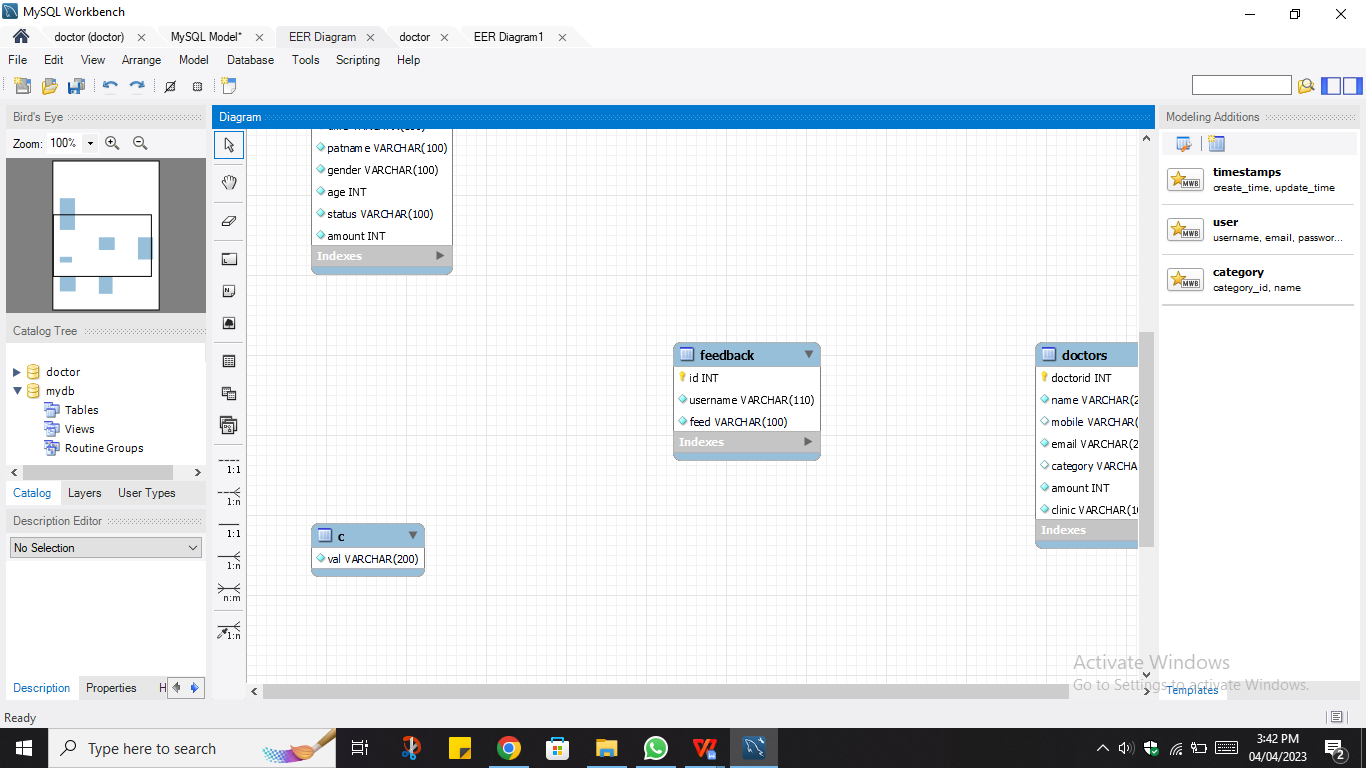
Clinic Table:

We have clinics in 3 different locations. So this table contains the exact location details of the clinics.

* state(varchar): This field is the primary key. For booking an appointment the user first has to enter nearby state. Here we have 3 options: Delhi, Karnataka and Maharashtra.
* street(varchar): This field stores the street name where the clinic is located.
* city(varchar): This field stores the city where the clinic is located.
* pin(varchar): This field stores the pin code of the area.
* phn(varchar): This field stores the contact no. of the clinic.

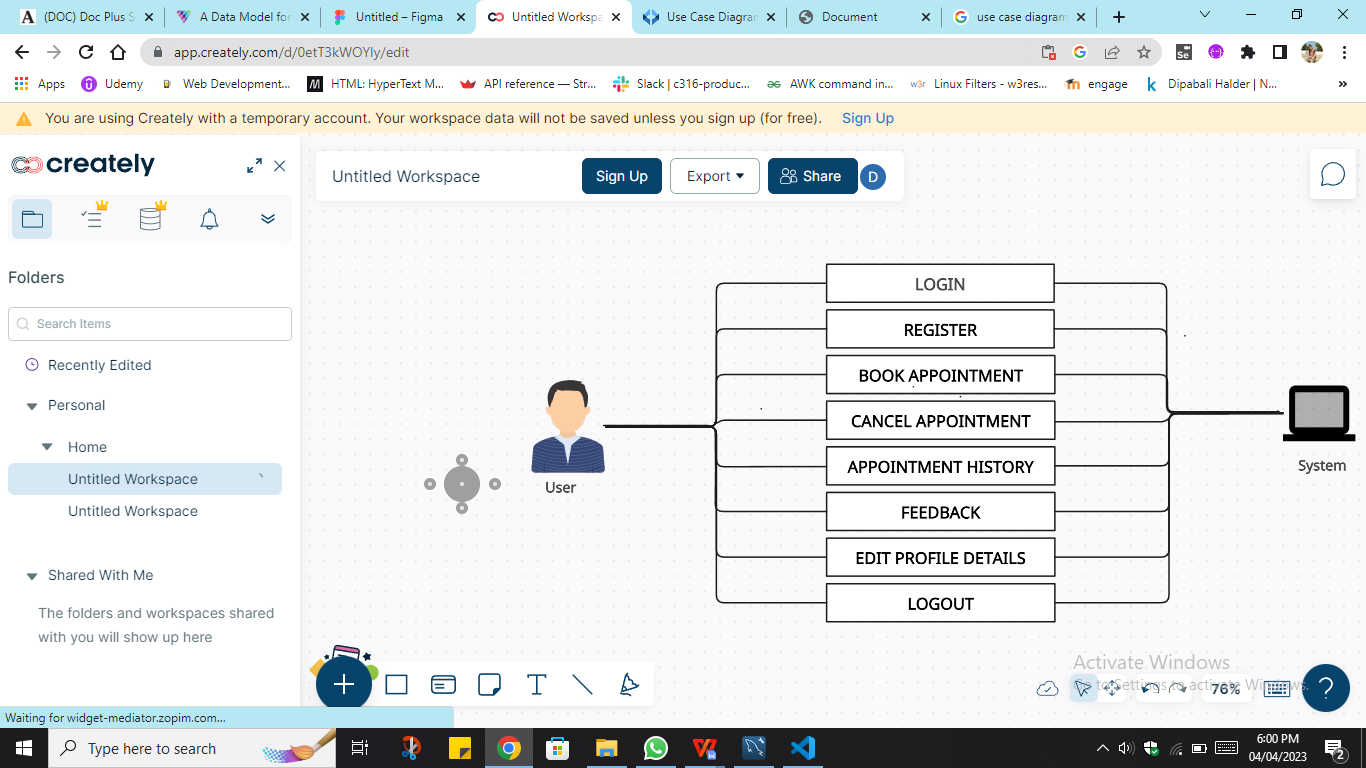
Booking Table:

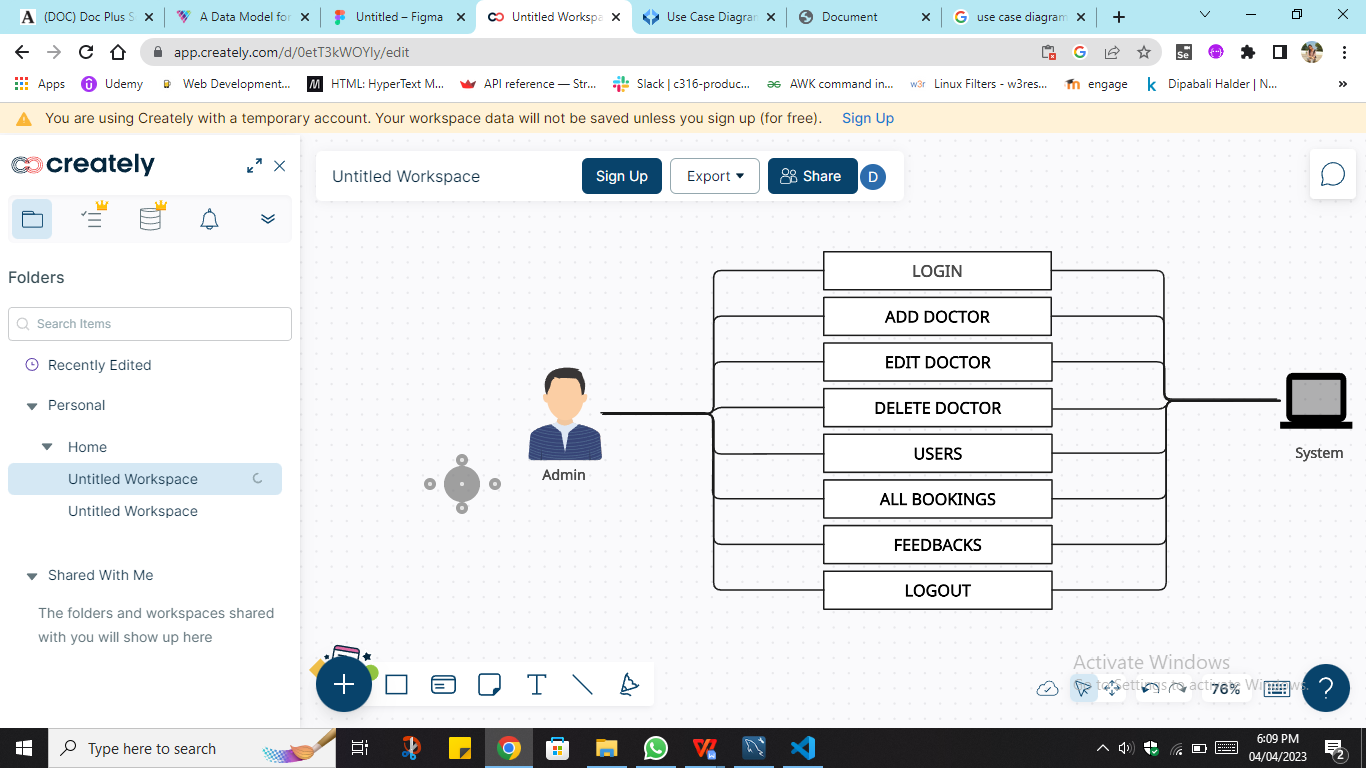
* appid(int): This is the primary key of the table which keeps a count on the appointments received and also helps to individually identify an appointment.
* doctorid(int): This field stores the doctor id. This key is used as a **reference of the doctorid key in the doctors table.**
* userid(int): This field stores the user id. This key is used as a **reference of the id key in the user table.**
* Clinic(varchar): As in our system we have clinics in 3 cities so this field stores the name of the clinic area to which the patient wants to visit. Also the options provided for the doctors depends on this field.
* date(varchar): This field stores the date of the appointment.
* Time(varchar): This field stores the time slot for the appointment.
* patname(varchar): A user may book an appointment for their mother, father or even any other relatives. So this field stores the patient’s name.
* gender(varchar): This field stores the gender of the patient.
* age(int): This field stores the age of the patient.
* status(varchar): This field stores the appointment status. It can have 2 values: “App. Booked” or “Canceled”
* Amount(int): This field stores the total appointment fee.

Feedback Table:

* id(int): This field is the primary key of the table and auto-increments itself whenever a new feedback is received by the user.
* username(varchar): This field refers to the **username field in the user table**. This field is used to uniquely identify the user who has provided a particular feedback.
* feed(varchar): This field stores the feedback content.

**USE CASE DIAGRAMS**





**USER FLOW DIAGRAMS**

Book Appointment:

No

ENTER DOCTOR NAME, DATE, TIME AND PATIENT DETAILS

No

Yes

Yes

STOP

DISPLAY APPOINTMENT DETAILS

BOOK

DISPLAYS A PAGE TO ENTER THE BOOKING DETAILS

GO

ENTER BOOKING LOCATION

START

Cancel Appointment:

STOP

If appointment status for the id entered is not “App. Booked”

If appointment status for the id entered is “App. Booked”

ERROR MESSAGE DISPLAYED

APPOINTMENT CANCELED MESSAGE DISPLAYED

CANCEL

ENTER APPOINTMENT ID

START