**EPIDEMIC MANAGEMENT SYSTEM**

**Abstract**

**Introduction**

The past few years in the 21st century has visually shown humanity where it was lacking when hit with such a huge catastrophe in the form of a pandemic. Realization dawned as infectious diseases started emerging without warning, disrupting trade, travel and livelihood of millions on the planet. The aim of this project is to design and develop a database for the public health administration and other multi-lateral agencies supporting epidemic response. These sectors can now use the efficient and organized data to tackle high level threats. The plan is to create a nexus with details such as patient records (such as contact and registration), availability of assets, migrant details along with an idea on the current statistics of the epidemic.

**Functionalities of the system**

The system recognizes 4 different types of users, admin, doctors/lab staffs, patients and visitors. Visitors are the people who are looking to get tested. These tests are taken at testing facilities and the visitors are free to choose a center as they desire using the system. The samples collected at these centers are tested and the results are then uploaded to the database by the doctors or lab staffs and then verified by the admin. People who have tested positive for a disease are added under the migrants list and their movements are tracked. People can keep a note of their test results using the system and can use it to check for the availability of hospitals and quarantine centers .They can then admit themselves to a hospital of their choice as a patient. The list and status of hospitals is added/modified by the doctors. Critical patients admitted in hospitals are of primary focus. The system can also generated patient reports, asset availability and guidelines report for the patients. For doctors and admin it can generate a migrants report and a critical case report along with the ones mentioned above.

**Modules in the system**

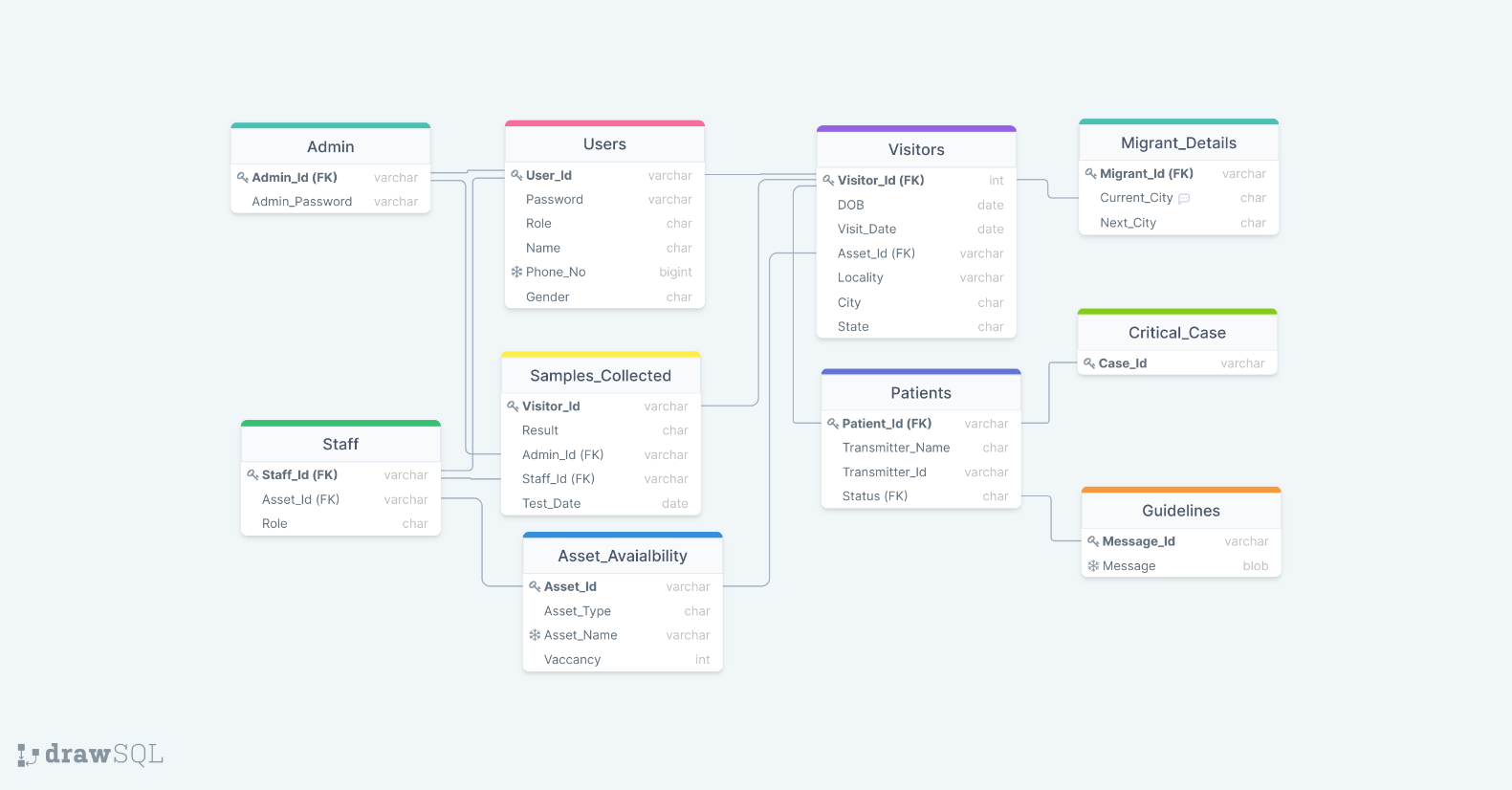
The system include the following modules

* A register and login portal for visitors, patients and doctors
* Real-time Dashboard on the current statistics of the epidemic.
* Checking for the availability of assets (for patients and doctors) and modifying/adding assets (for doctors)
* Interface for doctors to upload their test results and for patients to view them
* Report generation

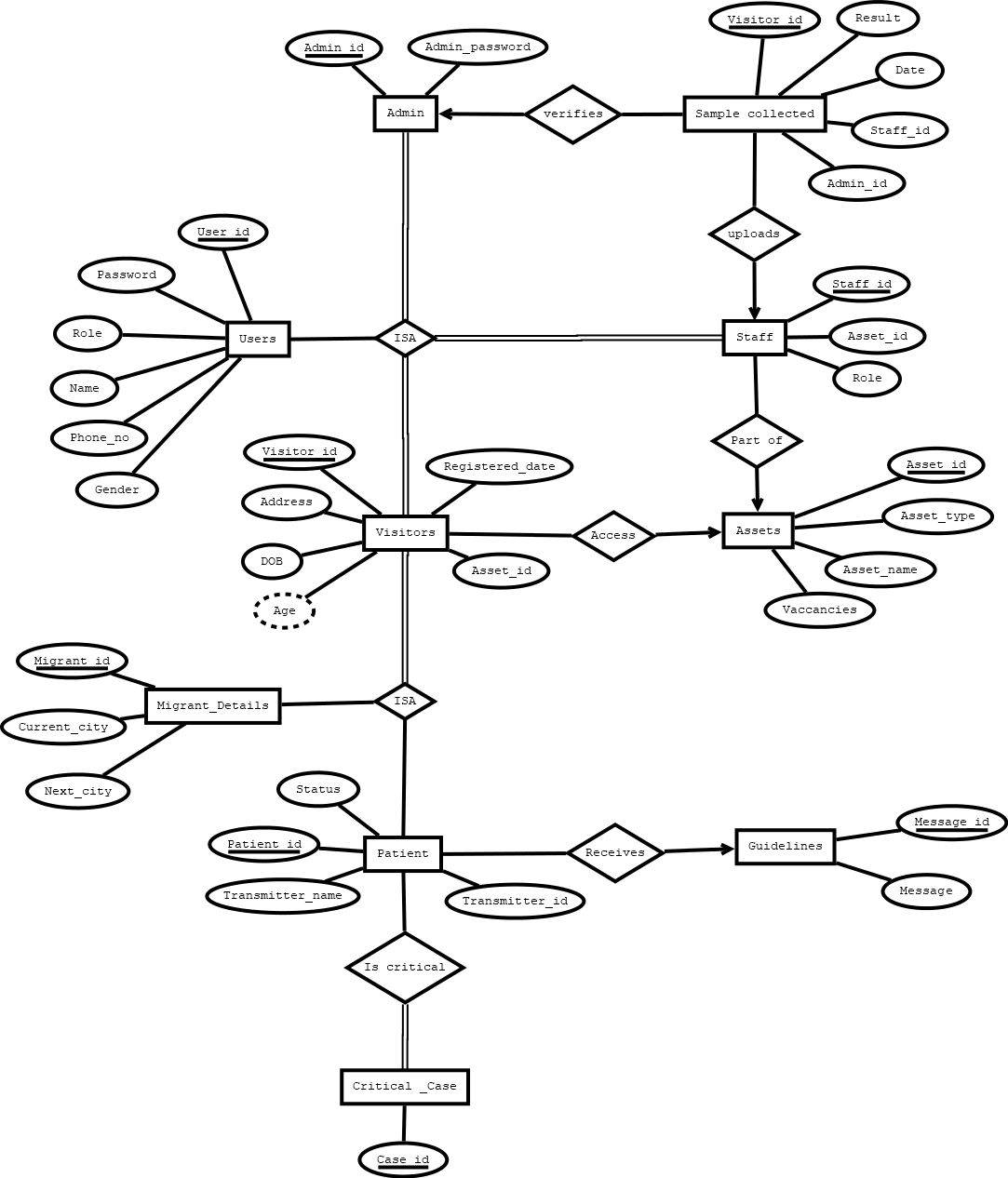
**Benefits of the system**

This system supports the management, doctors and patients by providing them necessary information. This system also can generate different reports based on the requirement. It also helps in foretelling the arrival of a pandemic and keep tracks of migrants.

**Schema Diagram**

****

**ER Diagram**

****

**Normalizations procedure**

**1st Normalization**

First Normal Form: If the relation is flat i.e. it has no composite and multivalued attribute, then it is in 1st Normal Form

The relation **Admin, Staff, Critical Case, Guidelines, Asset availability, Sample collected, Migrant Details** hav**e** no multivalued or composite attributes.

1. The **Users** table has a multivalued attribute Phone no

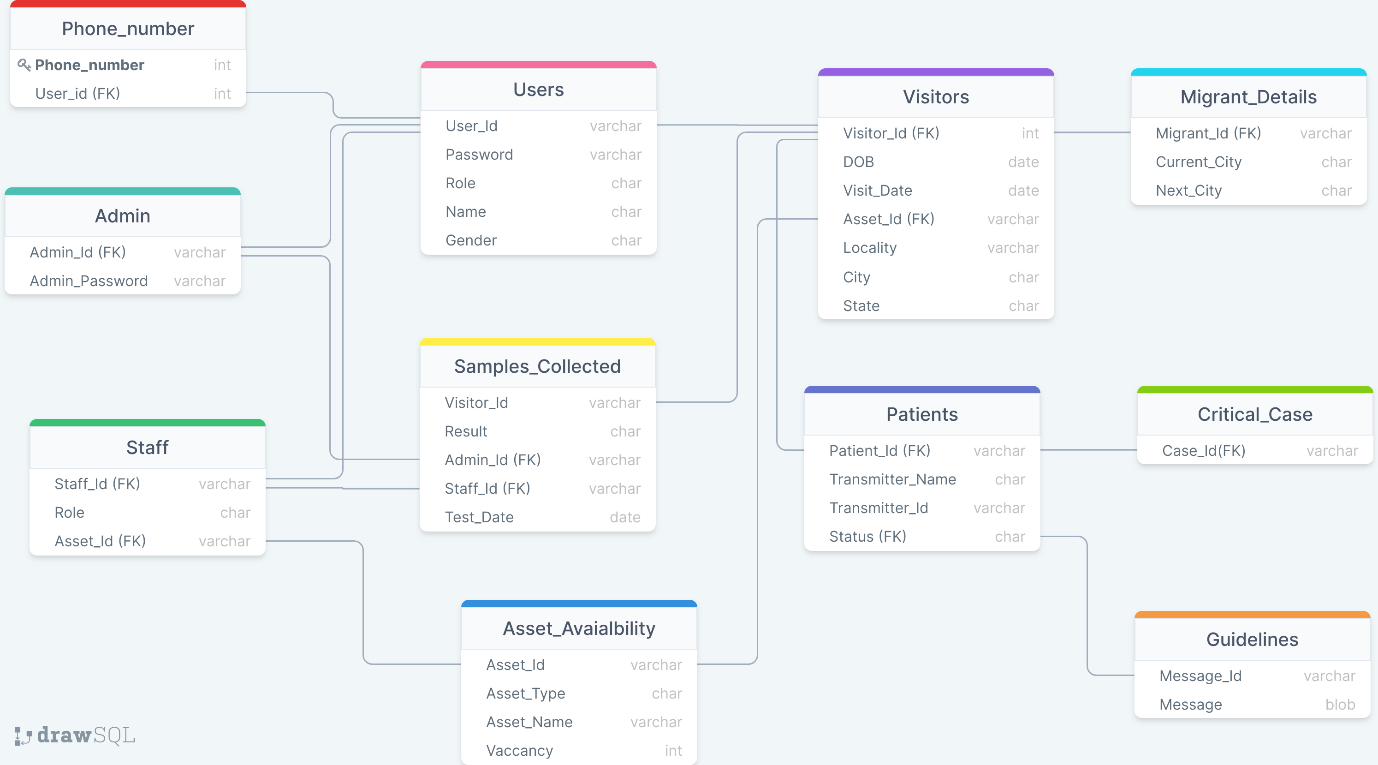
Users (User\_id, Password, Role, Name, Phone no, Gender)

* Users (User\_id, Password, Role, Name, Gender) and Phone Number(User\_Id ,Phone Number)

Relations in the database:

**Admin, Staff, Critical Case, Guidelines, Asset availability, Sample collected, Migrant Details, Users, Patients, Visitors**

+ **Phone Number**

****

**Functional dependencies (Trivial dependencies omitted unless specified)**

1. Admin

* Admin\_ID 🡪 {Admin\_Password, Admin\_id}
* Admin\_Password 🡪{ Admin\_ID, Admin password}

1. Critical case

* Critical case 🡪 critical case (Trivial)

1. Guidelines
   * Message\_id 🡪{ Message, Message\_id}
   * Message 🡪 {Message\_id, Message}
2. Asset Availability
   * Asset\_id 🡪 {Asset\_id, Asset type , Asset name ,Vacancy}
3. Staff
   * Staff\_id 🡪{Asset\_id , Role ,Staff\_id}
4. Samples collected

* Visitor\_id ,Test date 🡪 {Result, Admin\_id ,Staff\_id, Visitor\_id, Test date}

1. Phone number
   * Phone number 🡪 {User ID , Phone Number}
2. Users

* User\_id 🡪{User\_id , Password ,Role, Name, Gender}
* Password 🡪{Password , User\_id, Role, Name, Gender}

1. Visitors

* Visitor\_id 🡪 DOB, Locality, City, State
* Locality 🡪 City
* City 🡪 State
* Visitor\_id, Asset\_id, Visit\_date 🡪 {Visitor\_id, DOB, Locality, City, State, Asset\_id, Visit\_date}

1. Patients
   * Patient\_id 🡪 status
   * Transmitter\_id 🡪 Transmitter name
   * Patient\_id ,Transmitter\_id 🡪{ status ,Transmitter name , Patient ,Transmitter\_id}
2. Migrants

* Migrant\_id ,Current City , Next City 🡪 { Migrant\_id ,Current City , Next City}

**2nd Normalization**

A table is supposed to be in second normal form if, It is in 1st normal form and It does not have any partial dependency.

The relation **Admin, Staff, Phone number, Critical Case, Guidelines, Asset availability, Users, Samples collected, Migrants** have no partial dependencies.

1. Visitors relation has a partial dependency

Visitor\_id, Asset\_id, Visit\_date 🡪 {Visitor\_id, DOB, Locality, City, State, Asset\_id, Visit\_date}

Visitor\_id 🡪 DOB, Locality, City, State

Prime attributes: Visitor\_id, Asset\_id, Visit\_date

Non-prime Attributes: DOB, Locality, State, City

A part of the key derive a non-prime attribute. Hence it is not in 2NF form

Performing decomposition:

Visitors {Visitor\_id, DOB, Locality, City, State, Asset\_id, Visit\_date}

Logs (Visitor\_id, Visit\_date, Asset\_id) Visitor (visitor\_id, DOB, Locality, City, State)

Checking whether the decomposition is lossless

* Attr(Logs) union Attr(Visitor) = Attr(Visitor(Old))
* Attr(Logs) intersection Attr(Visitor) != NULL
* Attr(Logs) intersection Attr(Visitor) 🡪 Attr(Visitor)

Thus it is lossless decomposition

2) Patients table has partial dependencies

Patient\_id, Transmitter\_id 🡪{status, Transmitter name, Patient, Transmitter\_id}

Patient\_id 🡪 status

Transmitter\_id 🡪 Transmitter name

Performing decomposition

Patient {status, Transmitter name, Patient, Transmitter\_id}

Transmitters (Patient\_id, Transmitter name Patient (Patient\_id, Status)

Transmitter\_id)

Checking whether the decomposition is lossless

* Attr(Transmitter) union Attr(Patient) = Attr(Patient(Old))
* Attr(Transmitter) intersection Attr(Patient) != NULL
* Attr(Transmitter) intersection Attr(Patient) 🡪 Attr(Patient)

3) The newly formed Transmitters table has a partial dependency

Patient\_id, Transmitter\_id 🡪{Transmitter name, Patient, Transmitter\_id}

Transmitter\_id 🡪 Transmitter name

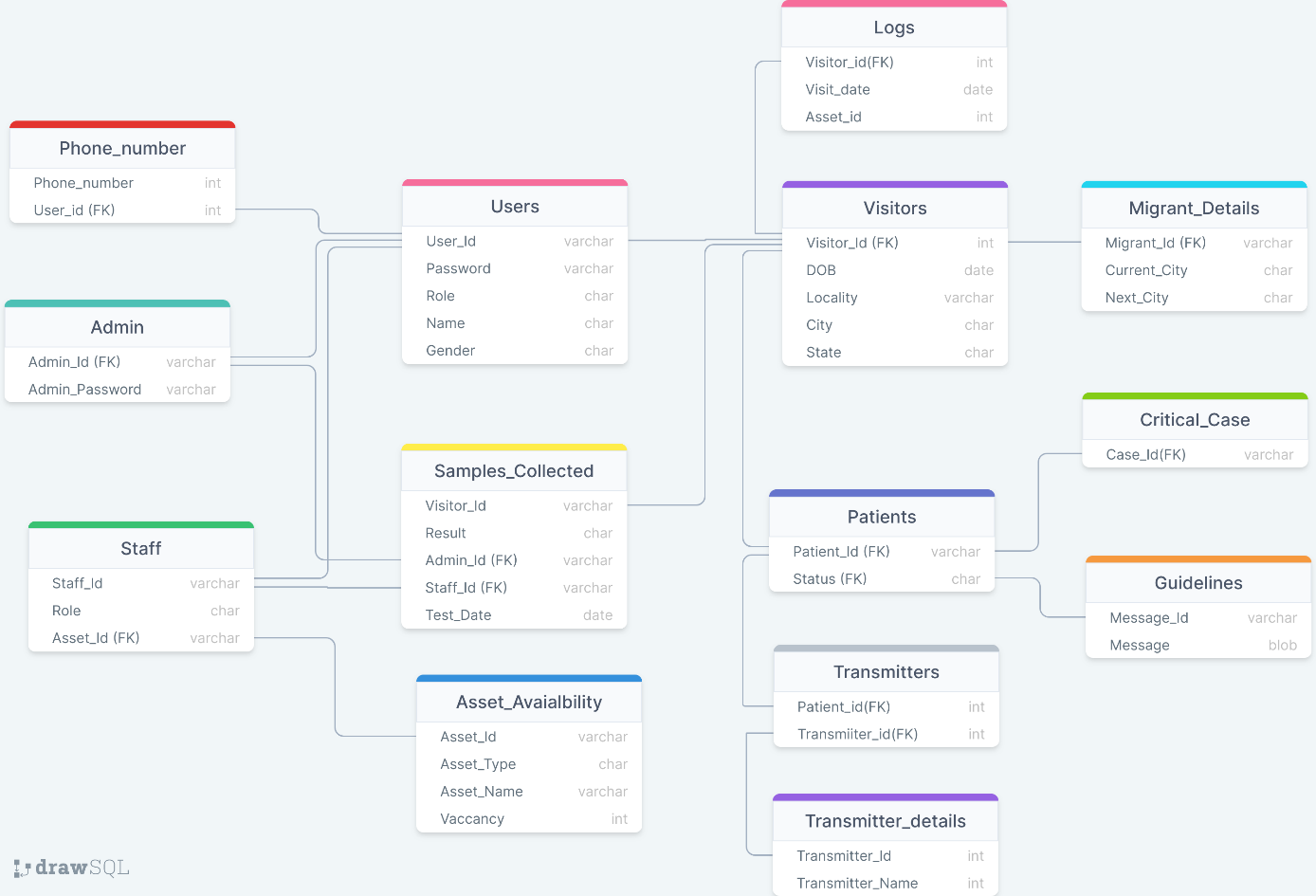
Performing decomposition:

Transmitters {Patient\_id, Transmitter name, Patient, Transmitter\_id}

Transmitters (Patient\_id, Transmitter\_id) Trasmitter\_details (Transmitter\_id, Transmiiter\_name)

Checking whether the decomposition is lossless

* Attr(Transmitter) union Attr(Trasmitter\_details) = Attr(Transmitter (Old))
* Attr(Transmitter) intersection Attr(Trasmitter\_details) != NULL
* Attr(Transmitter) intersection Attr(Trasmitter\_details) 🡪 Attr(Trasmitter\_details)



The relations after 2nd normalizing are

**Admin, Staff, Critical Case, Guidelines, Asset availability, Sample collected, Migrant Details, Users, Patients, Visitors**, **Phone Number**

**+ Transmitters, Transmitter details, Logs**

**3rd Normalization**

A relation will be in 3rd Normal Form if it is in 2nd Normal Form and not contain any transitive partial dependency. 3NF is used to reduce the data duplication. It is also used to achieve the data integrity.

If there is no transitive dependency for non-prime attributes, then the relation must be in third normal form.

After the 2nd normalization, the Visitor relation has transitivity.

1. Visitor (Visitor\_Id, DOB, Locality, City, State)

City is dependent on Locality and state is dependent on City.

We also see that Locality, City and State are individually dependent on Visitor\_Id.

Hence there is more than one instance of transitivity.

Visitor\_Id 🡪 Locality

Locality 🡪 City

Visitor\_Id 🡪 City

City 🡪 State

Hence we decompose the table:

Visitor (Visitor\_Id, DOB, Locality, City, State)

Visitor (Visitor\_Id, DOB, Locality) Locality (Locality, City, State)

Locality (Locality, City) City (City, State)

* Visitor\_Id 🡪 {Visitor\_Id, DOB, Locality}
* Locality 🡪 {Locality, City}
* City 🡪 {City, State}

Checking whether the decomposition is lossless (part 1)

* Attr(Visitor) union Attr(Locality) = Attr(Visitor (Old))
* Attr(Visitor) intersection Attr(Locality) !=NULL
* Attr(Visitor) intersection Attr(Locality) 🡪 Attr(Locality)

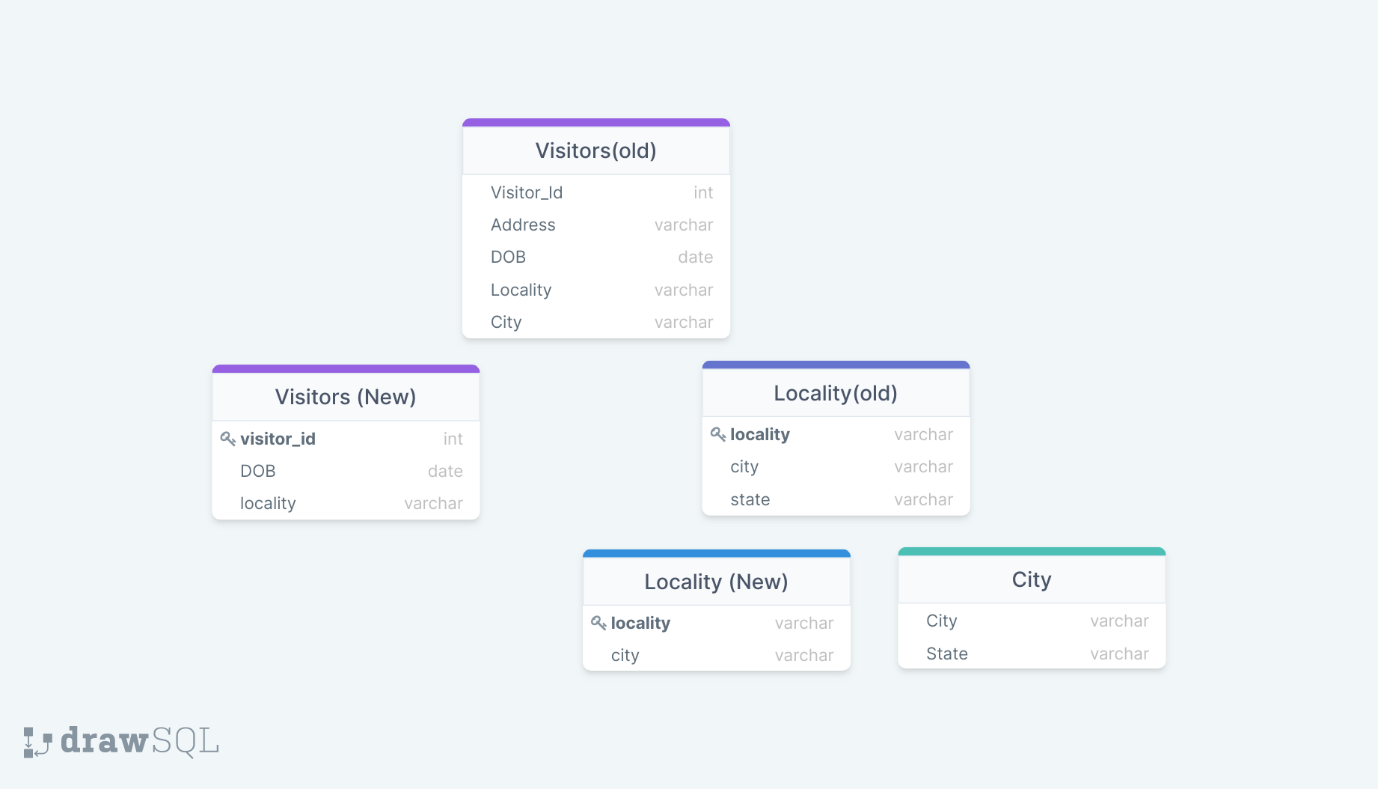
Checking whether the decomposition is lossless (part 2)

* Attr(Locality) union Attr(City) = Attr(Locality(Old))
* Attr(Locality) intersection Attr(State) !=NULL
* Attr(Locality) intersection Attr(State) 🡪 Attr(City)

Hence the relations after the 3rd Normalization are:

**Admin, Staff, Critical Case, Guidelines, Asset availability, Sample collected, Migrant Details, Users, Patients, Visitor**, **Phone Number, Transmitter, Logs**

**+ Locality, City**



**Comments used for creation of tables**

1) User table

create table Users(User\_Id varchar(10),Password varchar(20),Role char(10),Name varchar(20),Gender char(2),Primary key(User\_Id));

2) Phone number

Create table phone\_number (phone\_number bigint primary key, user\_id varchar (10), foreign key (User\_id) references users (User\_id));

3) Admin

create table Admin(Admin\_Id varchar(10),Admin\_Password varchar(20),Primary key(Admin\_Id),Foreign key(Admin\_Id) references Users(User\_Id));

4) City

create table City(City varchar(20),State varchar(20),Primary Key(City));

5) Locality

create table Locality(Locality varchar(20),City varchar(20),Primary Key(Locality),Foreign key(City) references City(City));

6) Visitor

create table Visitor(Visitor\_Id varchar(10),DOB date,Locality varchar(20),Primary Key(Visitor\_Id),Foreign key(Visitor\_Id) references Users(User\_Id),Foreign key(Locality) references Locality(Locality));

7) Logs

create table Logs(Visitor\_Id varchar(10),Visit Date date,Asset\_Id varchar(10),Primary key(Visitor\_Id,Asset\_id,Visit\_Date),Foreign key(Visitor\_Id) references Visitor(Visitor\_id));

8) Migrant details

create table Migrant\_Details(Migrant\_Id varchar(10),Current City varchar(20),Next City varchar(20),Primary key(Migrant\_Id, Current city, Next City),Foreign key(Migrant\_Id) references Visitor(Visitor\_Id));

9) Guidelines

Create table Guidelines (Message\_Id varchar (10), Message Blob, Primary key (Message\_Id));

10) Patients

create table Patients(Patient\_Id varchar(10),Status varchar(10),Primary key(Patient\_Id),Foreign key(Patient\_Id) references Visitor(Visitor\_Id),Foreign key(Status) references Guidelines(Message\_Id));

11) Transmitter\_dets

Create table Transmitter\_Dets (Transmitter\_Id varchar (10), Transmitter Name varchar (20), Primary key(Transmitter\_Id));

12) Transmitters

create table transmitters(Patient\_id varchar(10),Transmitter\_Id varchar(10),Primary key(Patient\_id,Transmitter\_id),Foreign key(Patient\_Id) references Patients(Patient\_ID),Foreign key(Transmitter\_Id) references Transmitter\_Dets(Transmitter\_ID)););

13) Asset Availability

Create table Asset\_Availability (Asset\_Id varchar (10),Asset Type varchar(20),Asset Name varchar(20),Vacancy int not null, Primary key(Asset\_Id));

14) Staff

create table Staff(Staff\_Id varchar(10),Role varchar(10),Asset\_Id varchar(10),Primary key(Staff\_Id),Foreign key(Asset\_Id) references Asset\_Availability(Asset\_ID));

15) Samples collected

create table Samples\_Collected(Visitor\_Id varchar(10),Result varchar(10),Admin\_Id varchar(10),Staff\_Id varchar(10),Test\_Date date,Primary key(Visitor\_Id, Test\_Date),Foreign key(Admin\_Id) references Admin(Admin\_ID),Foreign key(Staff\_Id) references Staff(Staff\_ID),Foreign key(Visitor\_Id) references Visitor(Visitor\_ID));

16) Critical Case

Create table Critical Case(Case\_Id varchar(10),Primary key(Case\_Id),Foreign key(Case\_Id) references Patients(Patient\_Id));

Comments used for populating tables with relevant records

1)Users

insert into users values("V105","asasasre","Visitor","mukesh","M");

insert into users values("V106","mjiuiukuki","Visitor","mukesh","M");

insert into users values("V107","portrtrgfg","Visitor","jacob","M");

insert into users values("V108","cvzxsder","Visitor","Hema","F");

insert into users values("V109","ghgh#$#43","Visitor","Anirudh","M");

insert into users values("V110","tyuy67hfg3$","Visitor","logesh","M");

insert into users values("V111","vbv676789","Visitor","Deepa","F");

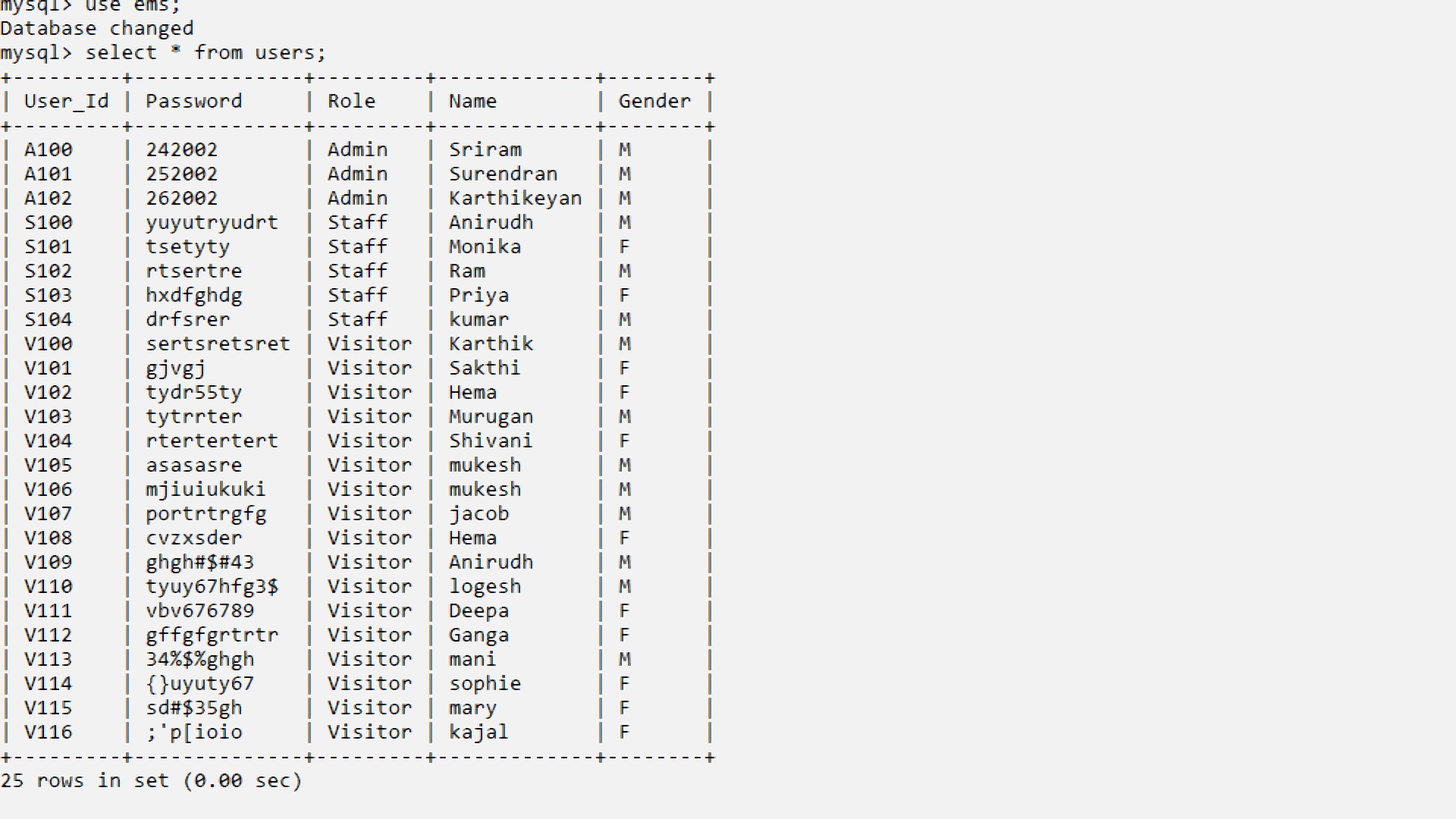
insert into users values("V112","gffgfgrtrtr","Visitor","Ganga","F");

insert into users values("V113","34%$%ghgh","Visitor","mani","M");

insert into users values("V114","{}uyuty67","Visitor","sophie","F");

insert into users values("V115","sd#$35gh","Visitor","mary","F");

insert into users values("V116",";'p[ioio","Visitor","kajal","F")

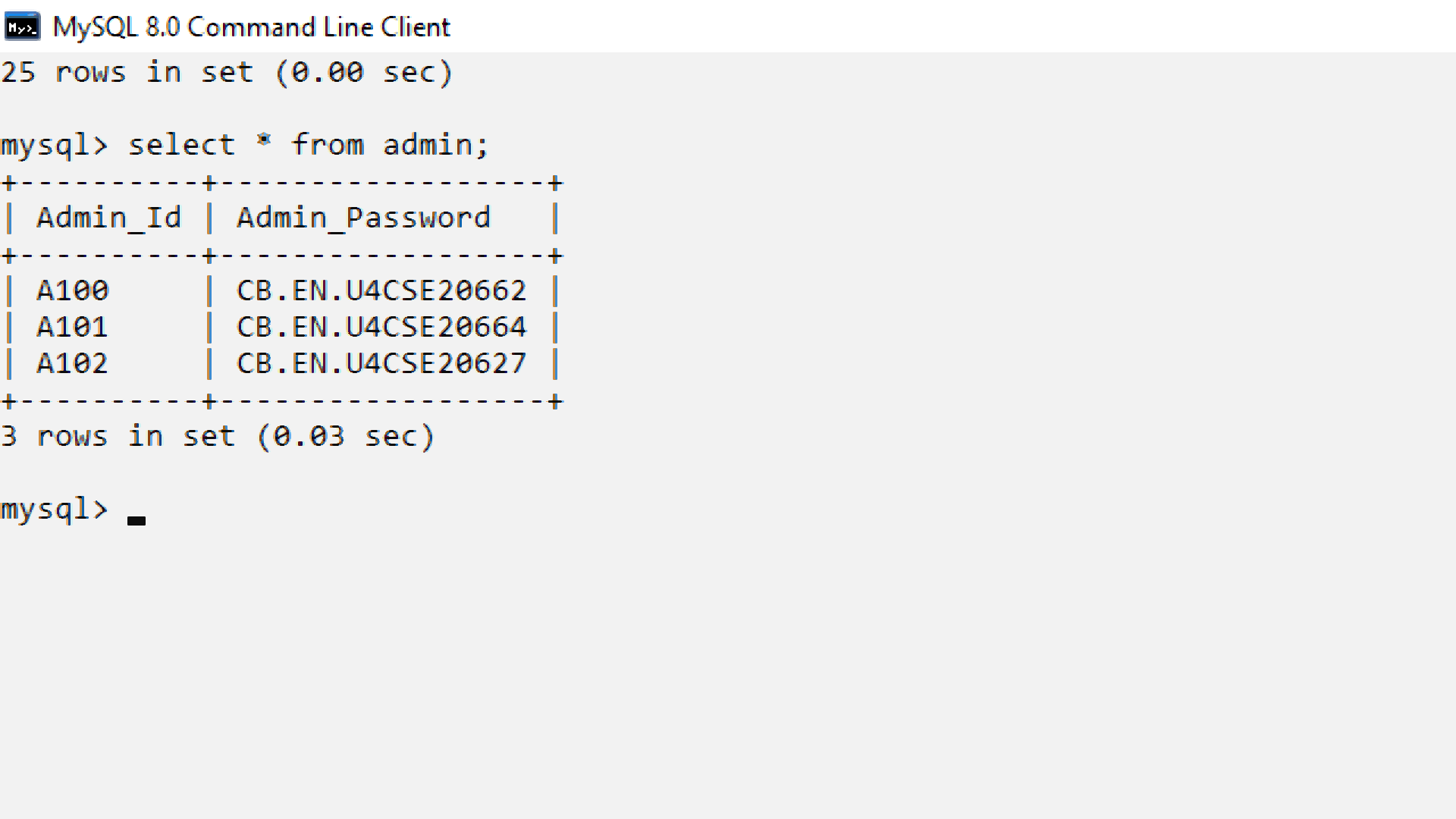


2)Admin

insert into admin values("A100","CB.EN.U4CSE20662");

insert into admin values("A101","CB.EN.U4CSE20664");

insert into admin values("A102","CB.EN.U4CSE20627");



3)Phone\_number

insert into phone\_number values(1234123412,"S100");

insert into phone\_number values(1234123672,"S101");

insert into phone\_number values(1234123452,"S102");

insert into phone\_number values(1234123415,"S103");

insert into phone\_number values(1234923414,"S104");

insert into phone\_number values(2234123412,"V100");

insert into phone\_number values(3234123412,"V101");

insert into phone\_number values(4234123412,"V102");

insert into phone\_number values(5234123412,"V103");

insert into phone\_number values(6234123412,"V104");

insert into phone\_number values(7234123412,"V105");

insert into phone\_number values(8234123412,"V106");

insert into phone\_number values(9234123412,"V107");

insert into phone\_number values(1134123412,"V108");

insert into phone\_number values(1034123412,"V109");

insert into phone\_number values(1334123412,"V110");

insert into phone\_number values(1434123412,"V111");

insert into phone\_number values(1534123412,"V112");

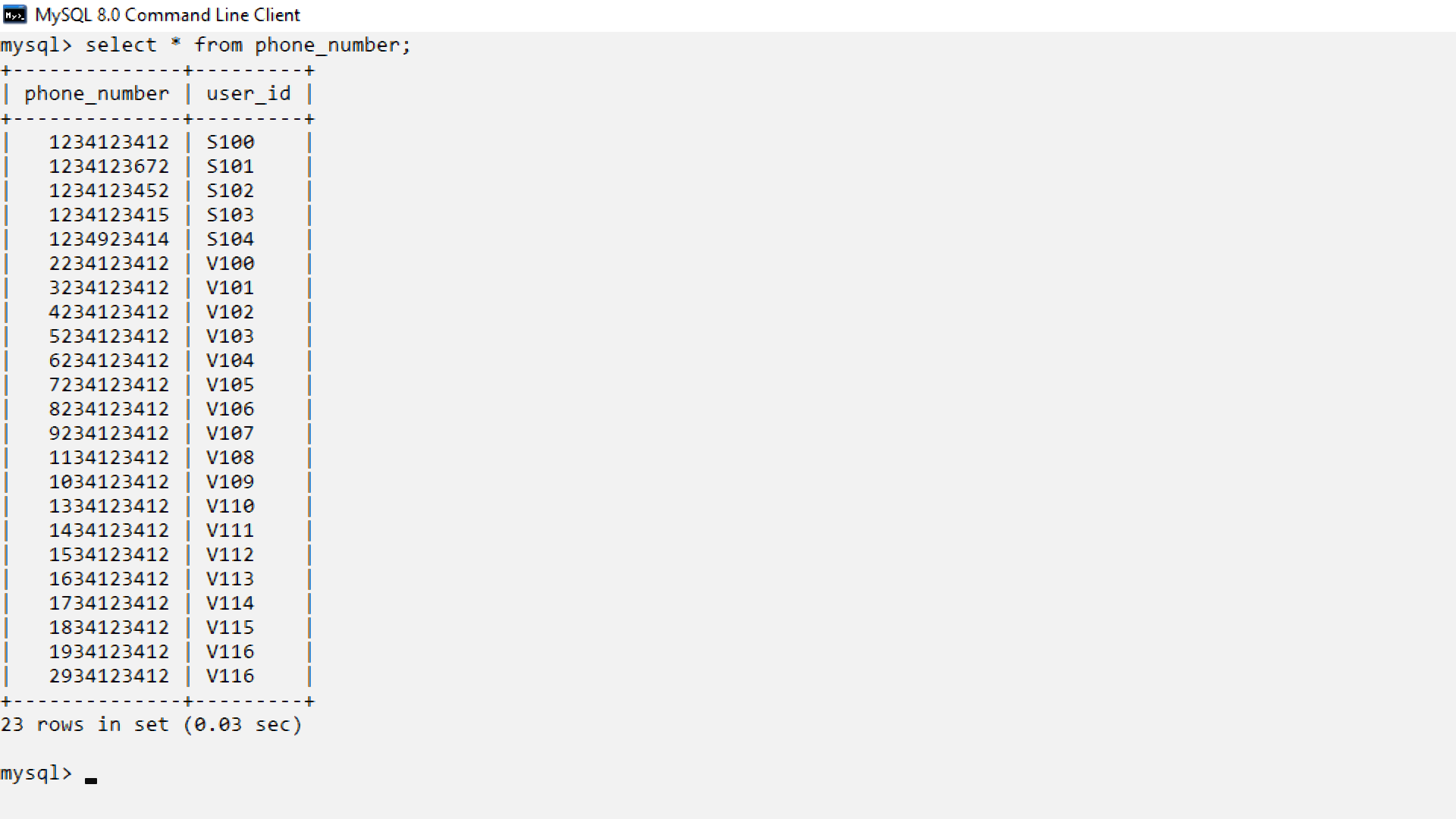
insert into phone\_number values(1634123412,"V113");

insert into phone\_number values(1734123412,"V114");

insert into phone\_number values(1834123412,"V115");

insert into phone\_number values(1934123412,"V116");

insert into phone\_number values(2934123412,"V116");



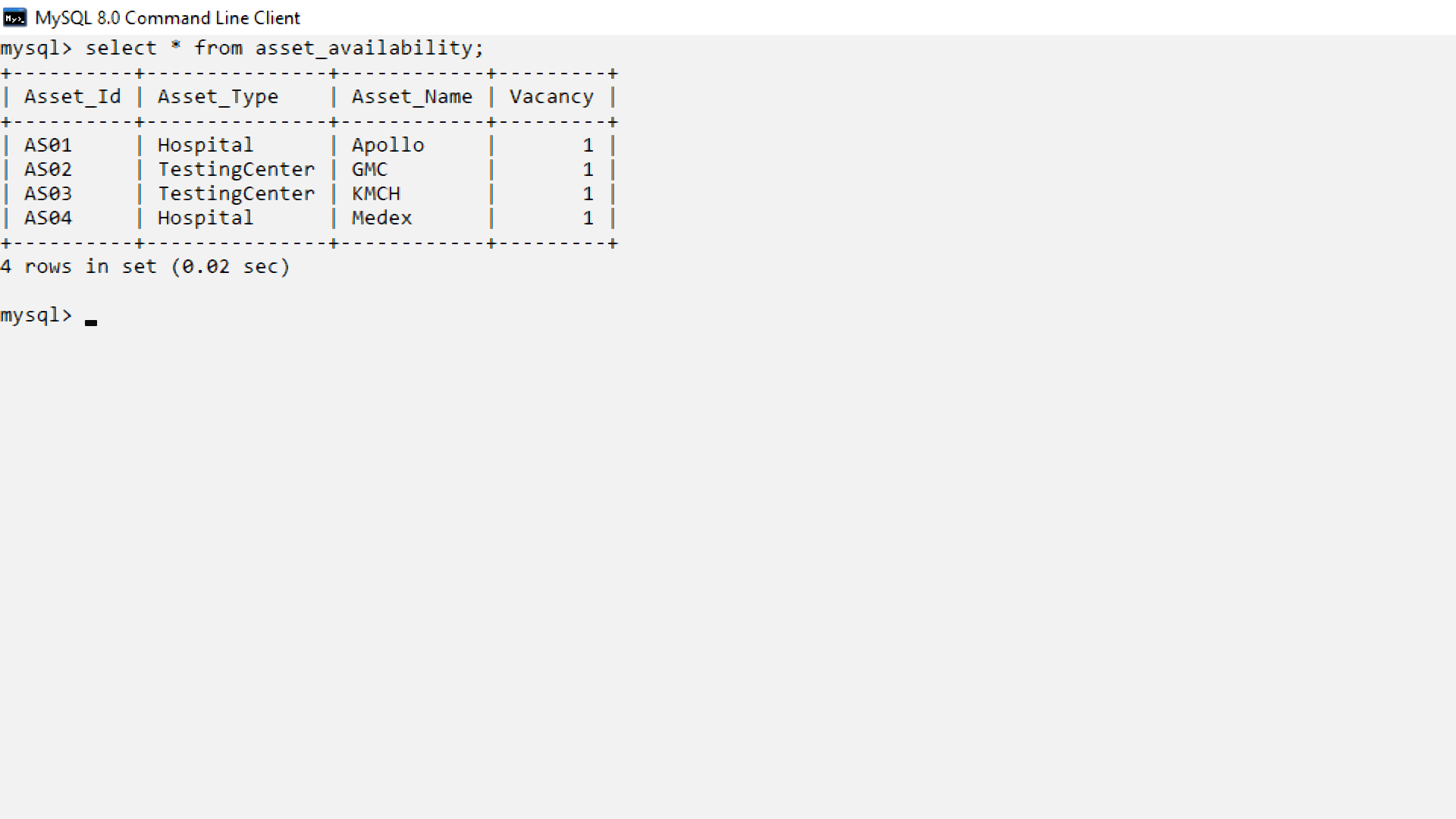
4) asset\_Availability

insert into asset\_availability values("AS01","Hospital","Apollo",1);

insert into asset\_availability values("AS02","TestingCenter","GMC",1);

insert into asset\_availability values("AS03","TestingCenter","KMCH",1);

insert into asset\_availability values("AS04","Hospital","Medex",1);



5)Staff

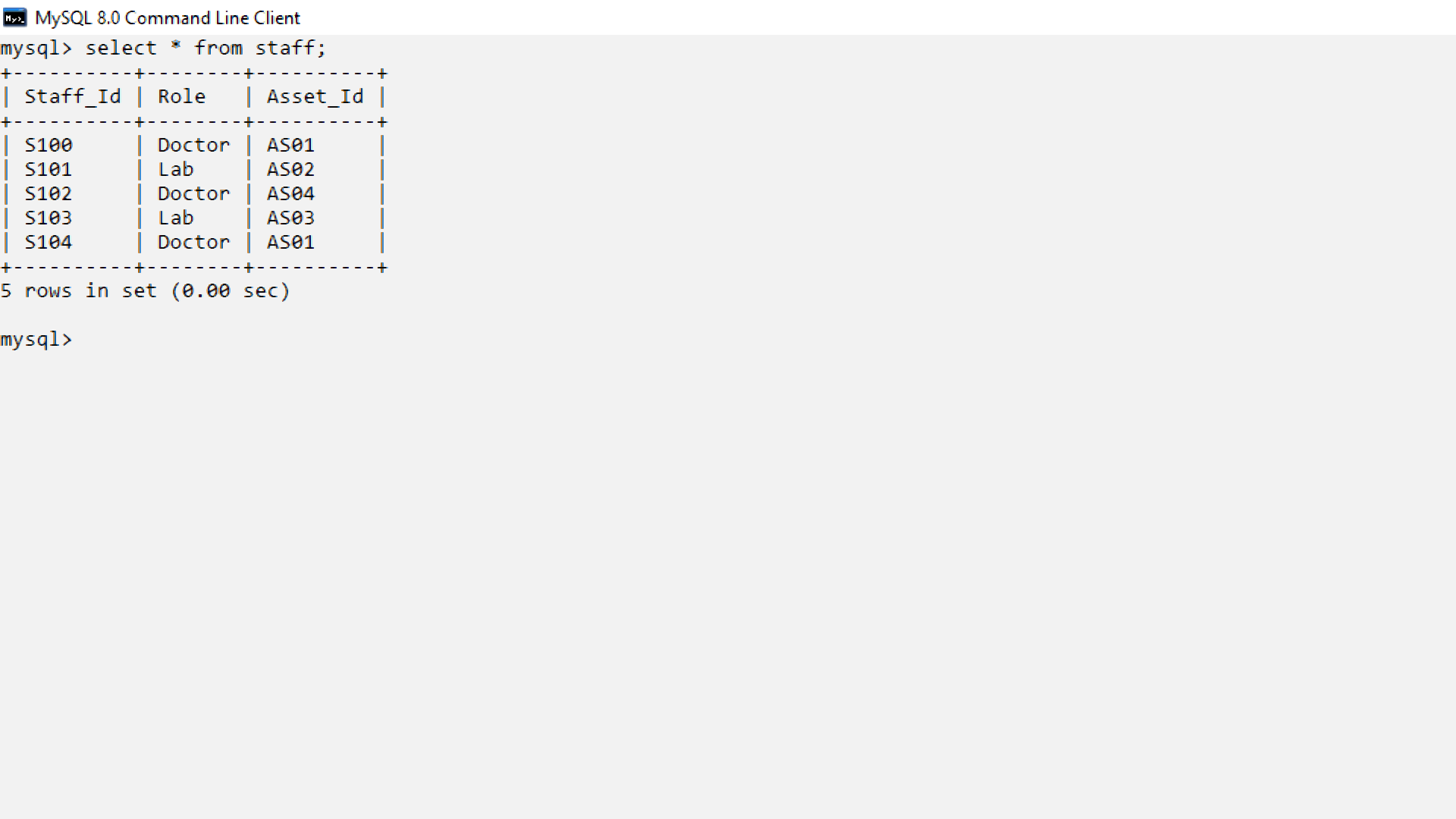
insert into staff values("S100","Doctor","AS01");

insert into staff values("S101","Lab","AS02");

insert into staff values("S102","Doctor","AS04");

insert into staff values("S103","Lab","AS03");

insert into staff values("S104","Doctor","AS01");



6) Visitors

insert into visitor values("V100","2000-09-23","T nagar");

insert into visitor values("V101","1998=09-22","Andheri");

insert into visitor values("V102","1996-01-12","Gurugram");

insert into visitor values("V103","1998-01-01","Alipore");

insert into visitor values("V104","1999-11-01","Gurugram");

insert into visitor values("V105","2002-04-11","T nagar");

insert into visitor values("V106","2001-11-19","T nagar");

insert into visitor values("V107","1990-01-23","Gurugram");

insert into visitor values("V108","1991-02-12","Alipore");

insert into visitor values("V109","1992-04-09","Andheri");

insert into visitor values("V110","1995-05-08","Alipore");

insert into visitor values("V111","1988-08-30","Gurugram");

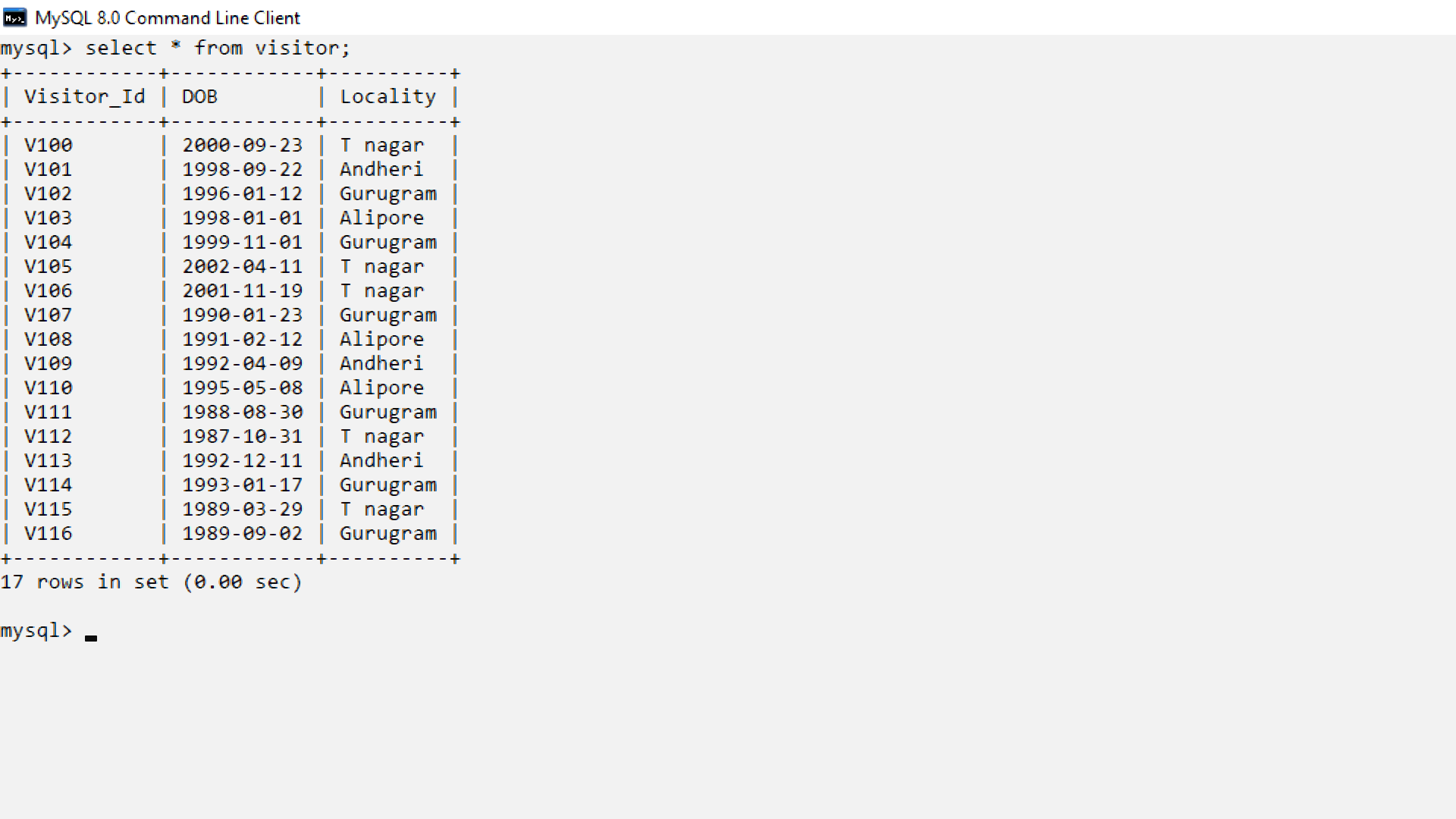
insert into visitor values("V112","1987-10-31","T nagar");

insert into visitor values("V113","1992-12-11","Andheri");

insert into visitor values("V114","1993-01-17","Gurugram");

insert into visitor values("V115","1989-03-29","T nagar");

insert into visitor values("V116","1989-09-02","Gurugram");



7)Locality

insert into locality values("T nagar","Chennai");

insert into locality values("Andheri","Mumbai");

insert into locality values("Gurugram","Delhi");

insert into locality values("Alipore","Kolkata");

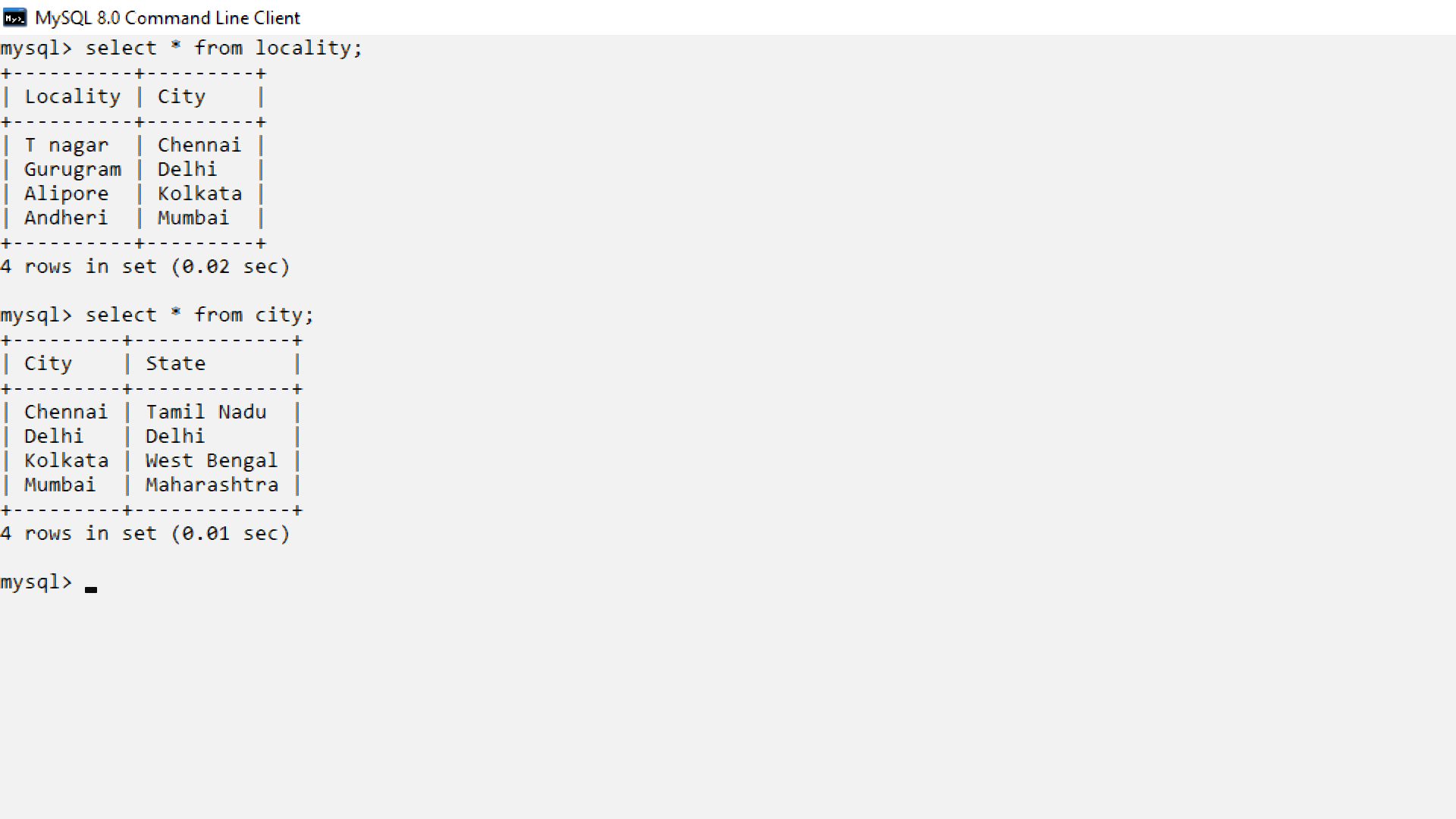
8) City

insert into city values("Chennai","Tamil Nadu");

insert into city values("Mumbai","Maharashtra");

insert into city values("Delhi","Delhi");

insert into city values("Kolkata","West Bengal");



9)Logs

insert into logs values("V100","2021-12-12","AS01");

insert into logs values("V101","2021-12-13","AS02");

insert into logs values("V103","2021-12-13","AS01");

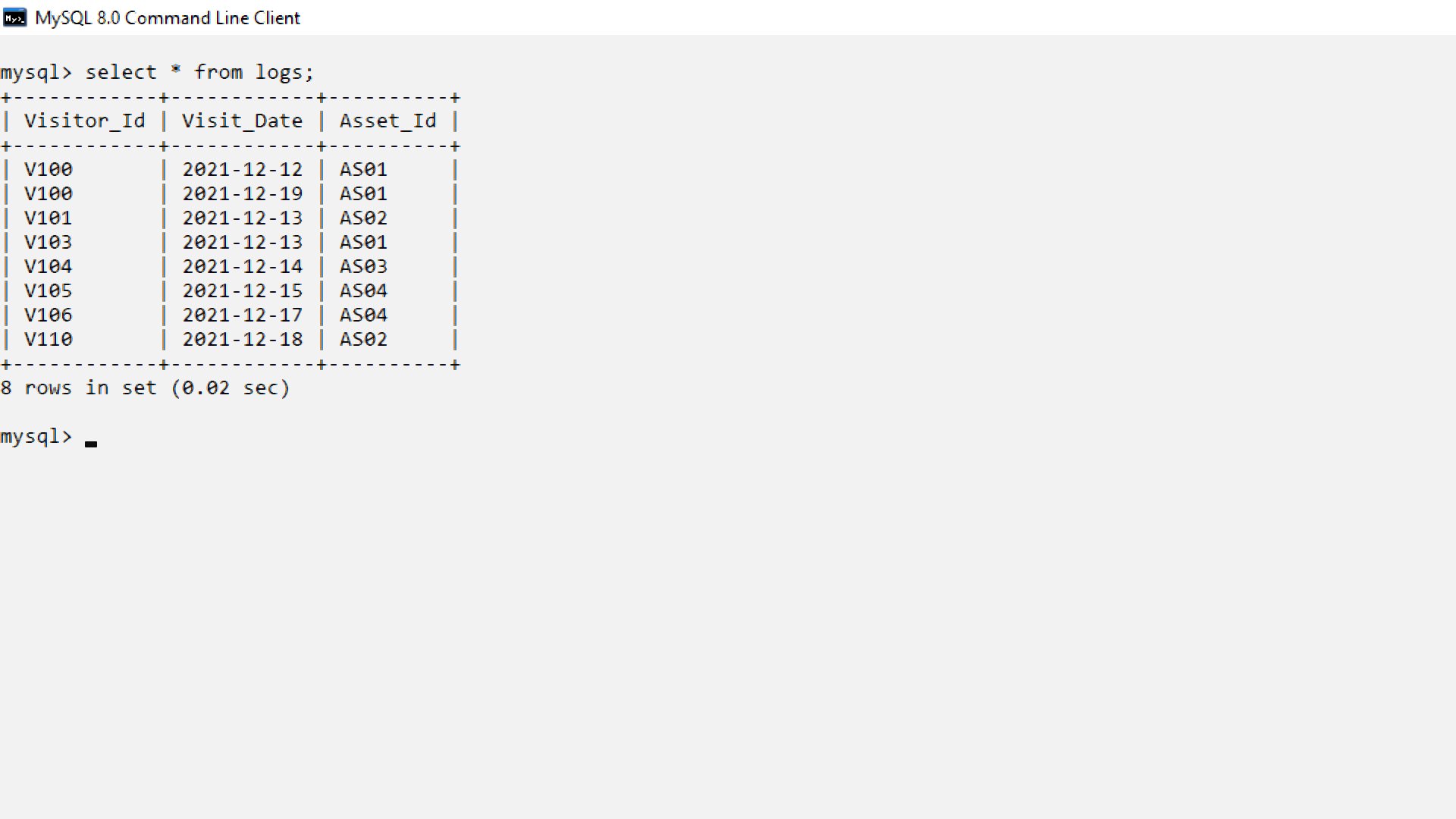
insert into logs values("V104","2021-12-14","AS03");

insert into logs values("V105","2021-12-15","AS04");

insert into logs values("V106","2021-12-17","AS04");

insert into logs values("V110","2021-12-18","AS02");

insert into logs values("V100","2021-12-19","AS01");



10)migrant\_details

insert into migrant\_details values("V100","Chennai","Coimbatore");

insert into migrant\_details values("V100","Coimbatore","Madurai");

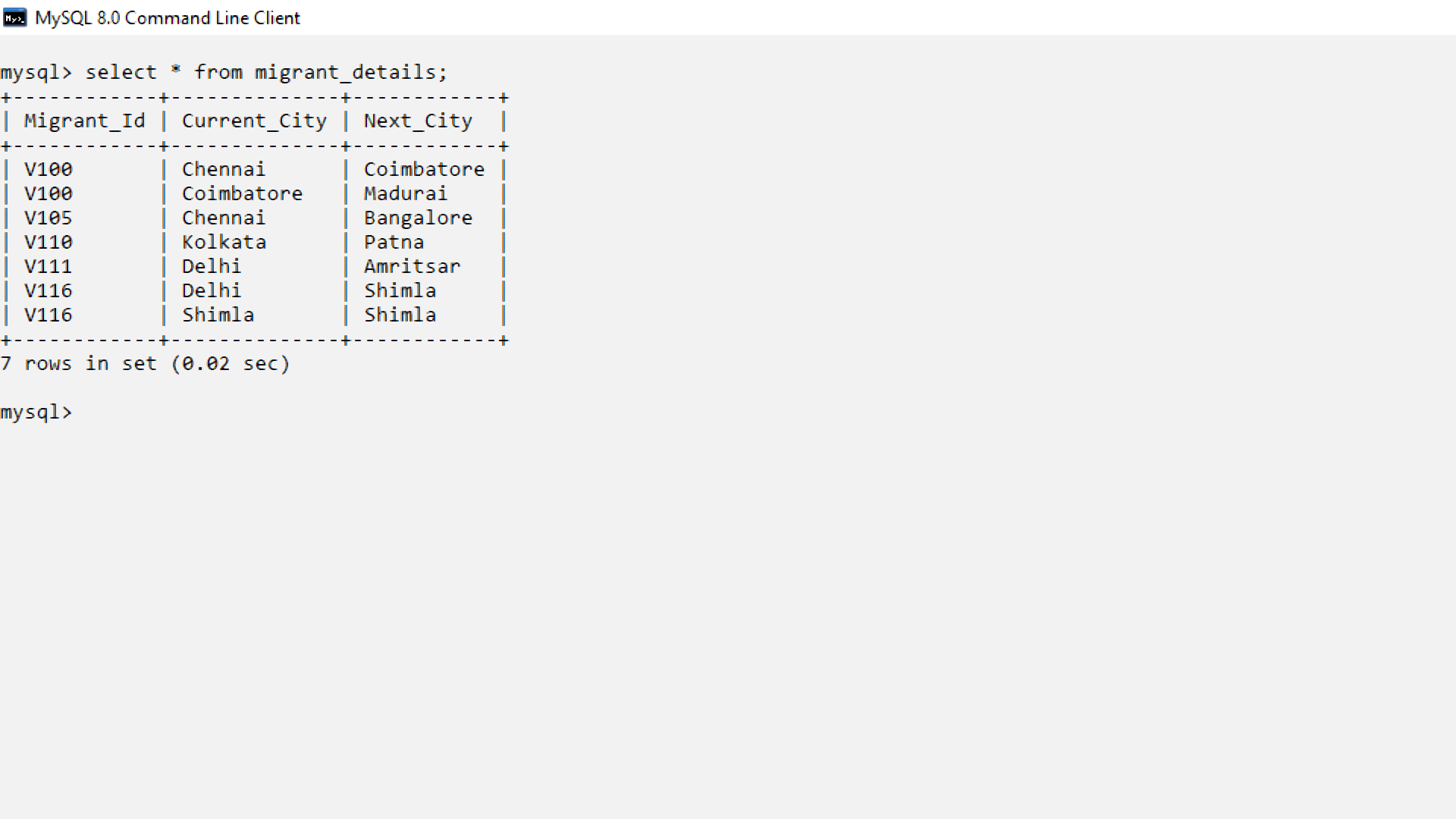
insert into migrant\_details values("V105","Chennai","Bangalore");

insert into migrant\_details values("V110","Kolkata","Patna");

insert into migrant\_details values("V111","Delhi","Amritsar");

insert into migrant\_details values("V116","Delhi","Shimla");

insert into migrant\_details values("V116","Shimla","Shimla");



11)Samples\_collected

insert into samples\_collected values("V100","Negative","A100","S101","2021-12-12");

insert into samples\_collected values("V101","Negative","A100","S101","2021-12-13");

insert into samples\_collected values("V103","Negative","A100","S101","2021-12-13");

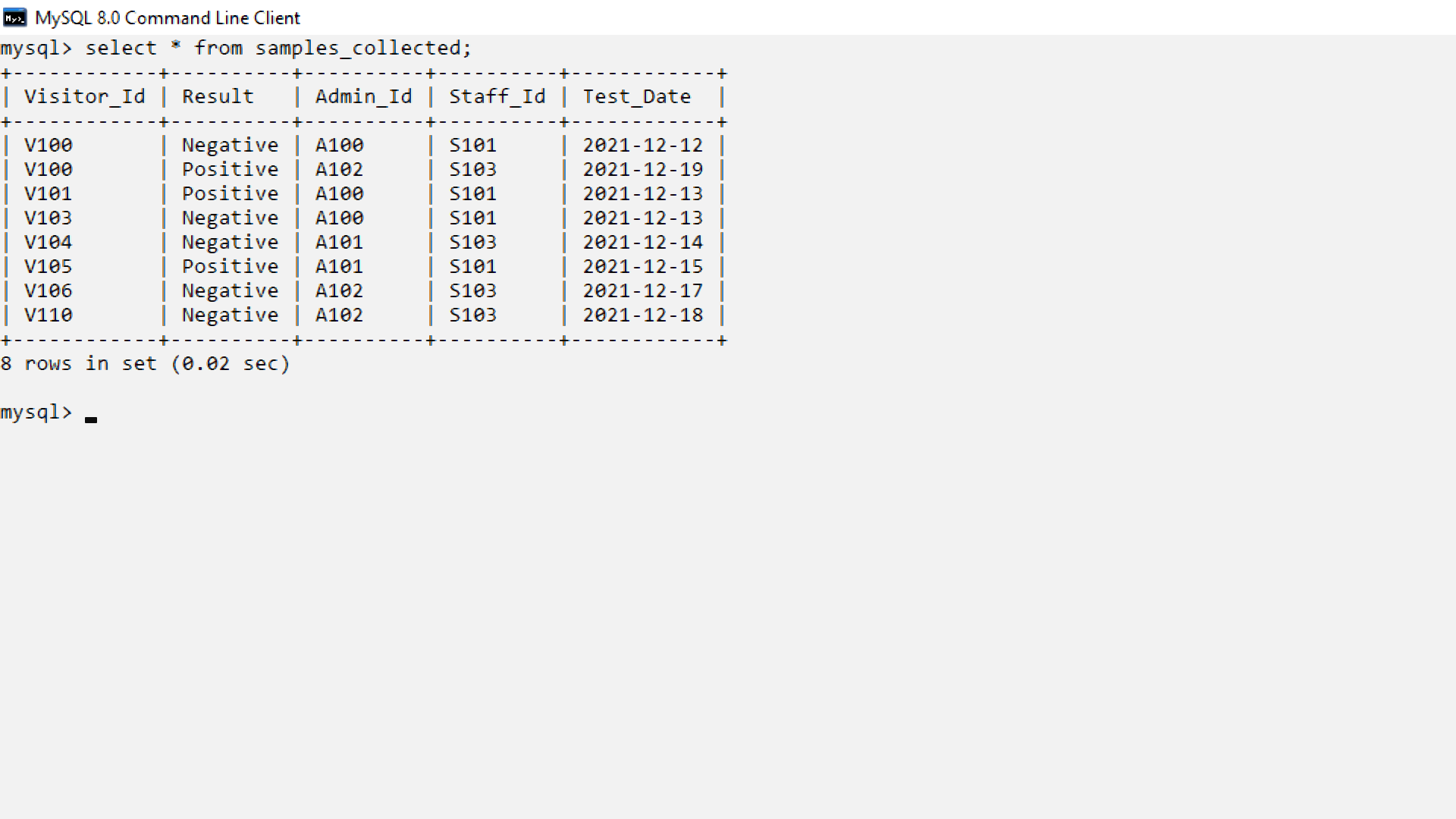
insert into samples\_collected values("V104","Negative","A101","S103","2021-12-14");

insert into samples\_collected values("V105","Positive","A101","S101","2021-12-15");

insert into samples\_collected values("V106","Negative","A102","S103","2021-12-17");

insert into samples\_collected values("V110","Negative","A102","S103","2021-12-18");

insert into samples\_collected values("V100","Positive","A102","S103","2021-12-19");



12)Patients

insert into patients values("V100","Stable");

insert into patients values("V101","Severe");

insert into patients values("V105","Critical");

13)Guidelines

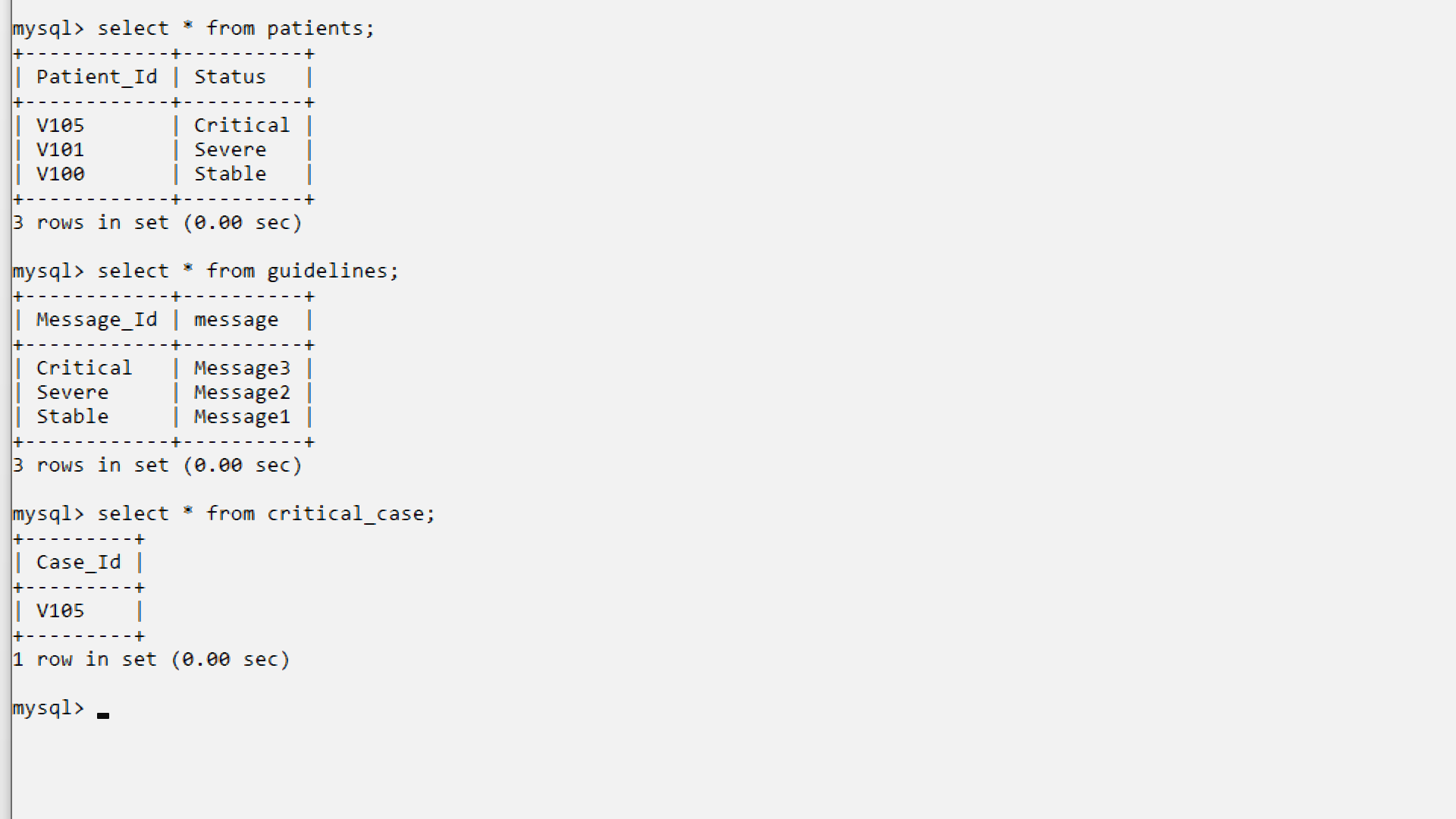
insert into guidelines values("Stable","Message1");

insert into guidelines values("Severe","Message2");

insert into guidelines values("Critical","Message3");

14)Critical case

insert into critical\_case values("V105");



15) Transmitter\_dets

insert into transmitter\_dets values("T001","Rithesh");

insert into transmitter\_dets values("T002","Sneha");

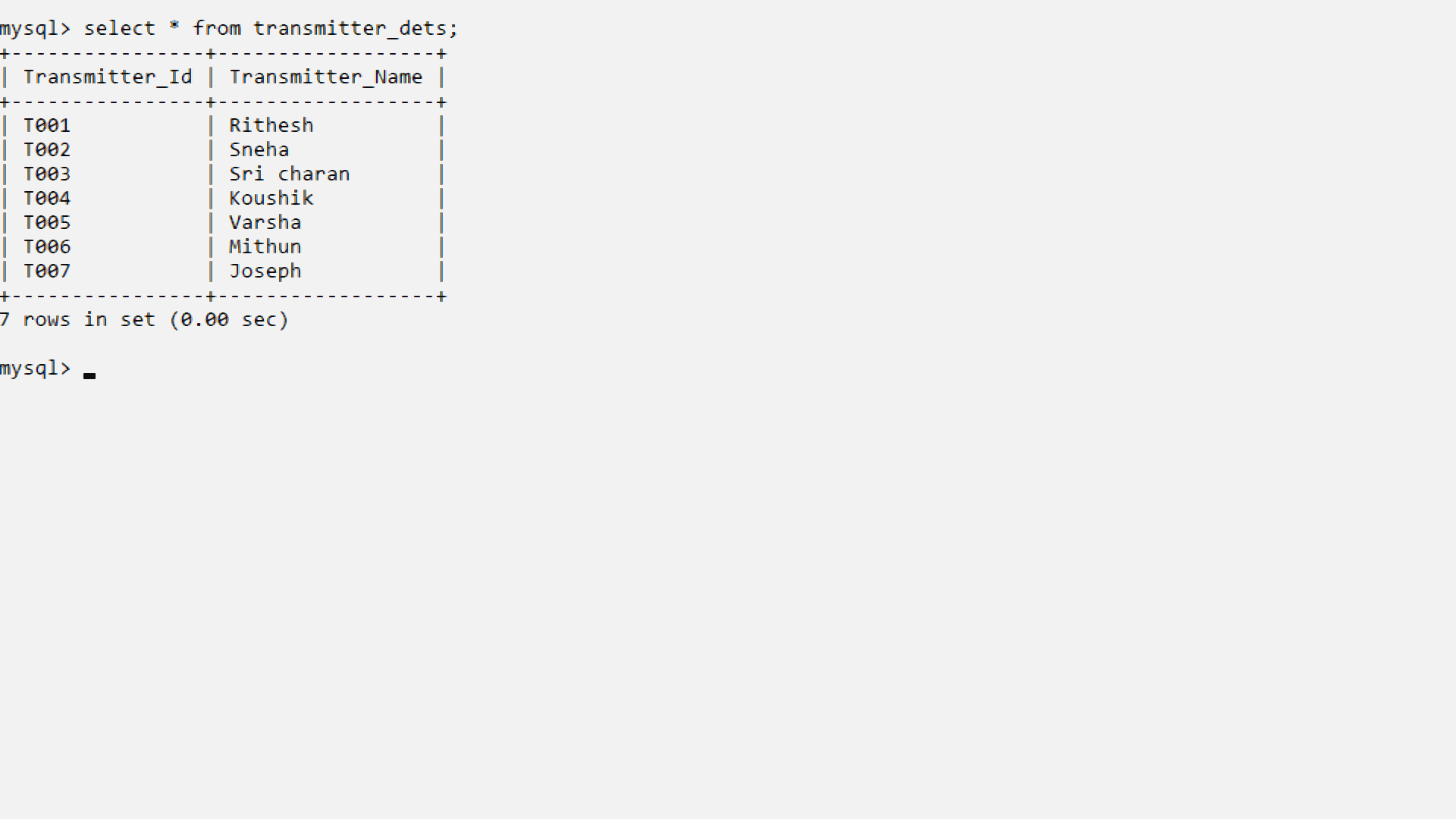
insert into transmitter\_dets values("T003","Sri charan");

insert into transmitter\_dets values("T004","Koushik");

insert into transmitter\_dets values("T005","Varsha");

insert into transmitter\_dets values("T006","Mithun");

insert into transmitter\_dets values("T007","Joseph");



16)Transmitters

insert into transmitters values("V100","T001");

insert into transmitters values("V100","T002");

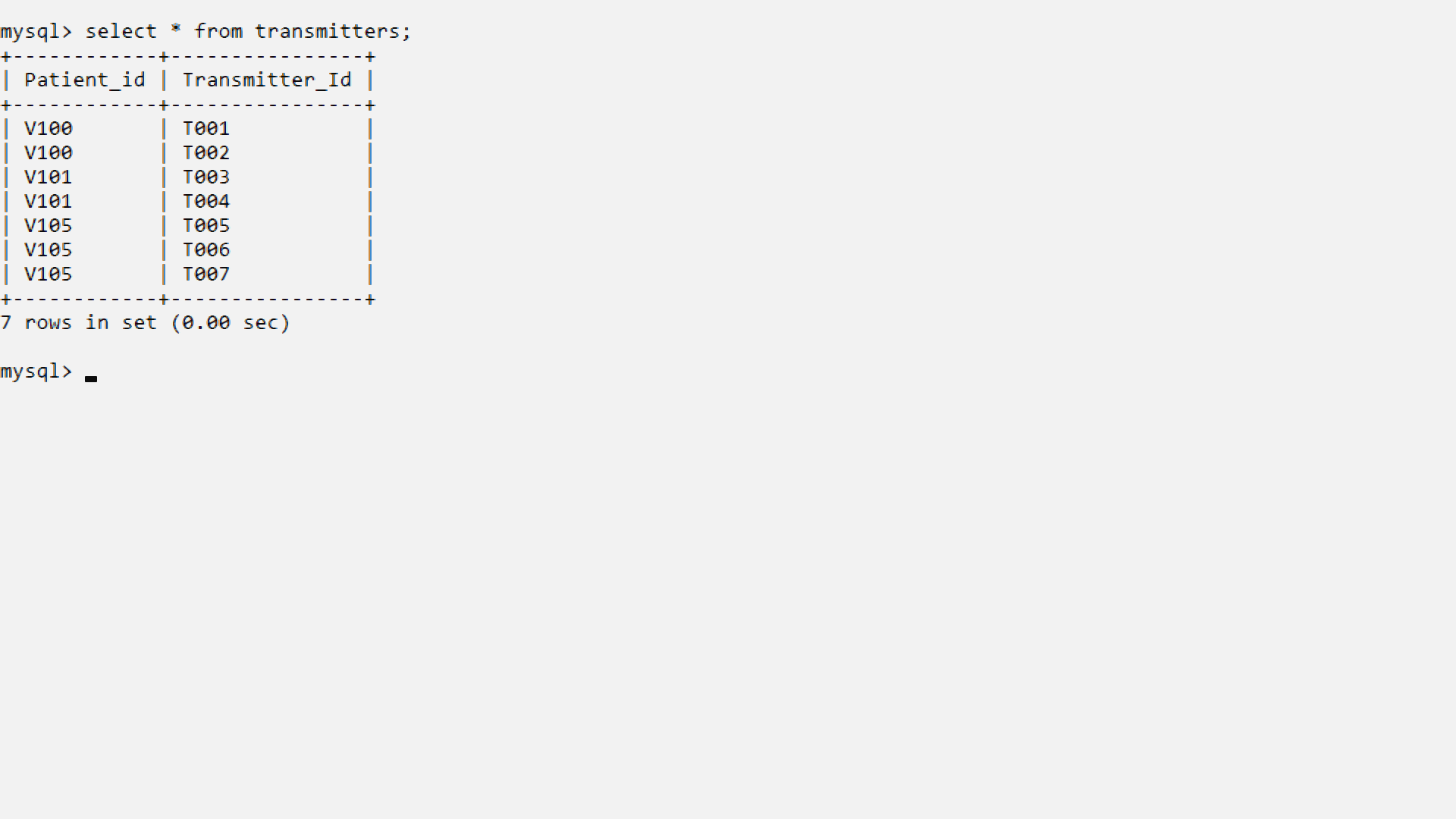
insert into transmitters values("V101","T003");

insert into transmitters values("V101","T004");

insert into transmitters values("V105","T005");

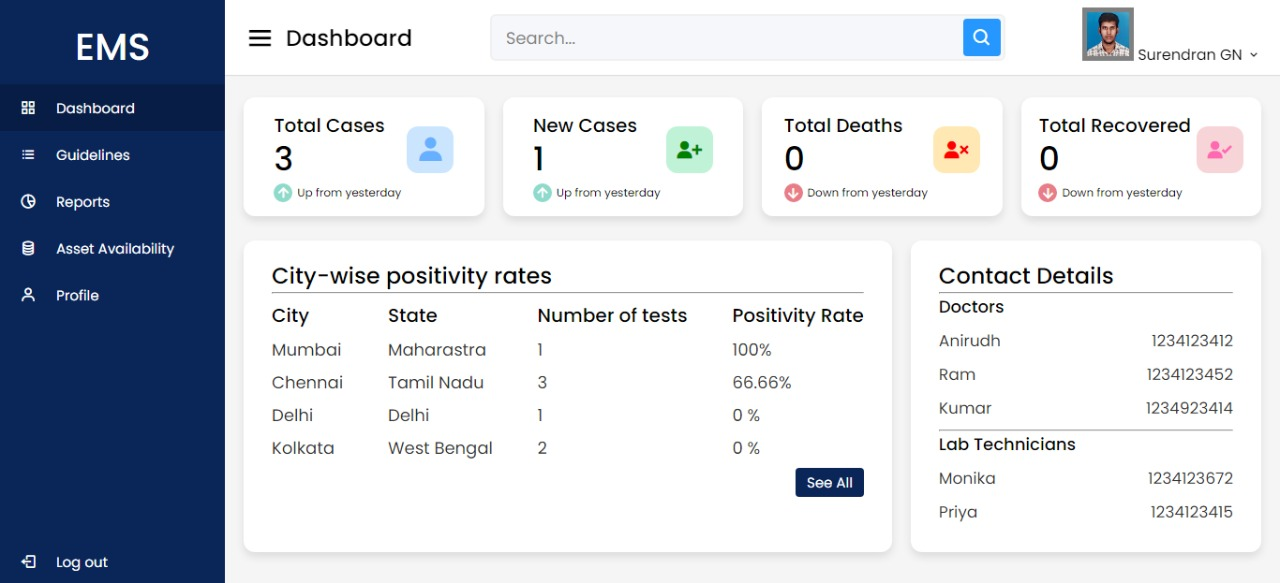
insert into transmitters values("V105","T006");

insert into transmitters values("V105","T007");

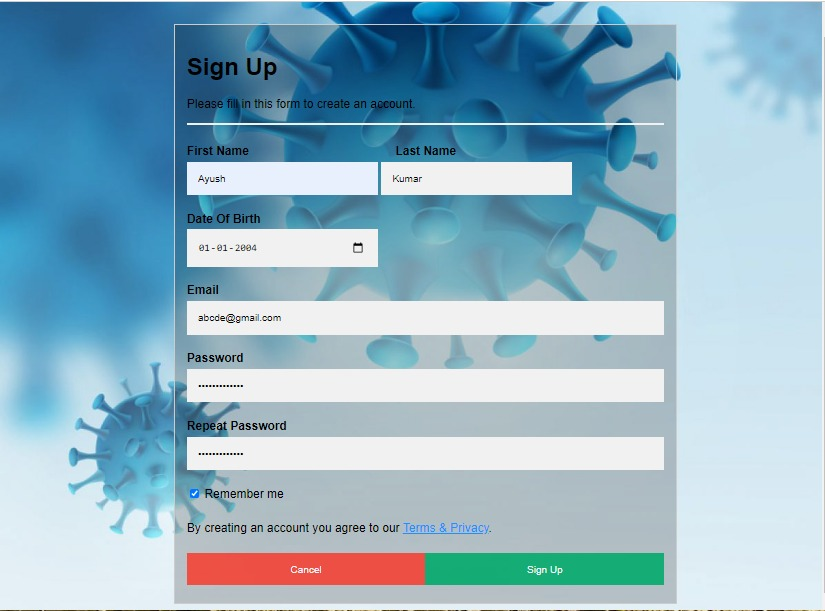


**Sample User interface design screens**

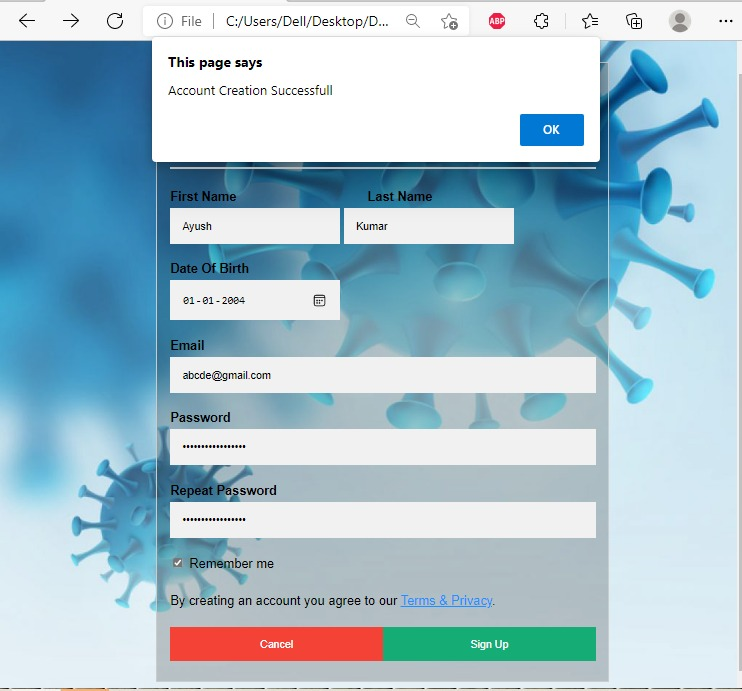
**Dashboard**



**New User**



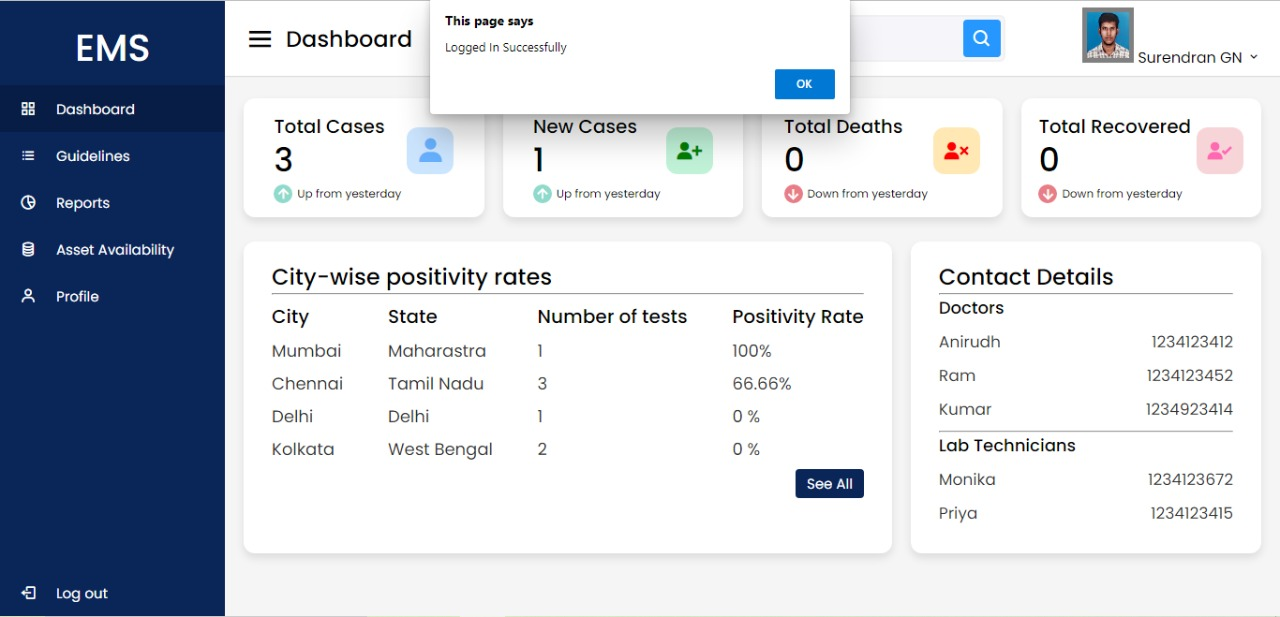
**Account Creation successful**



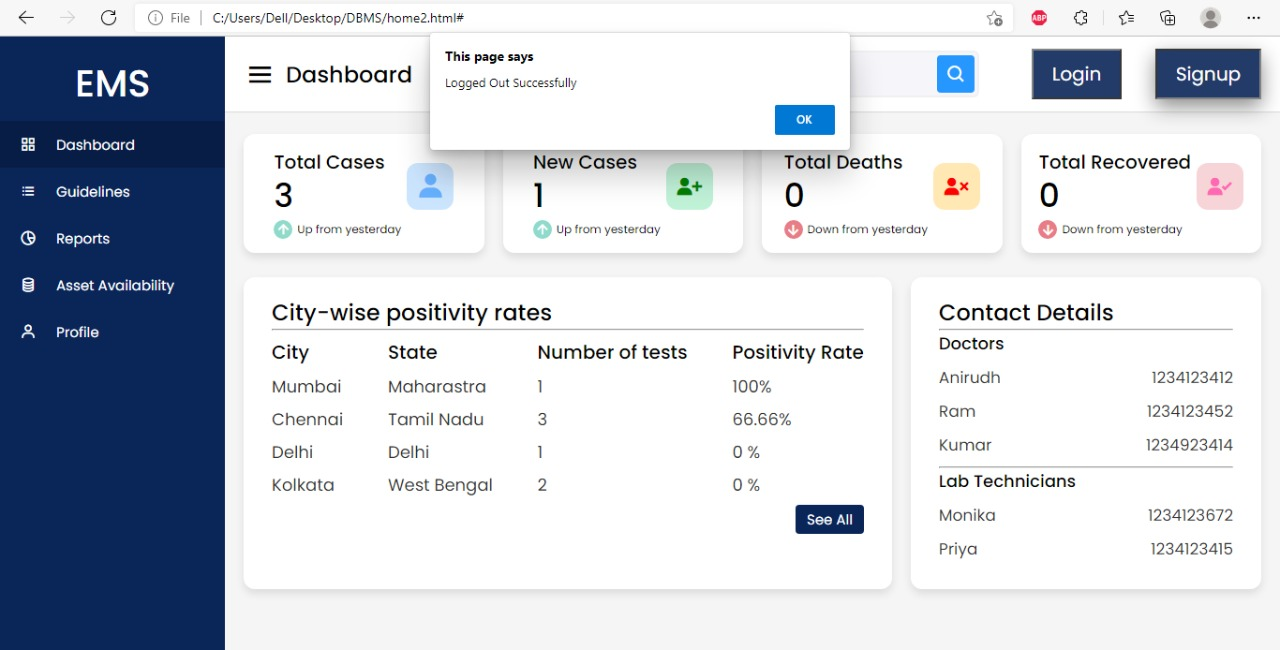
**Login page**



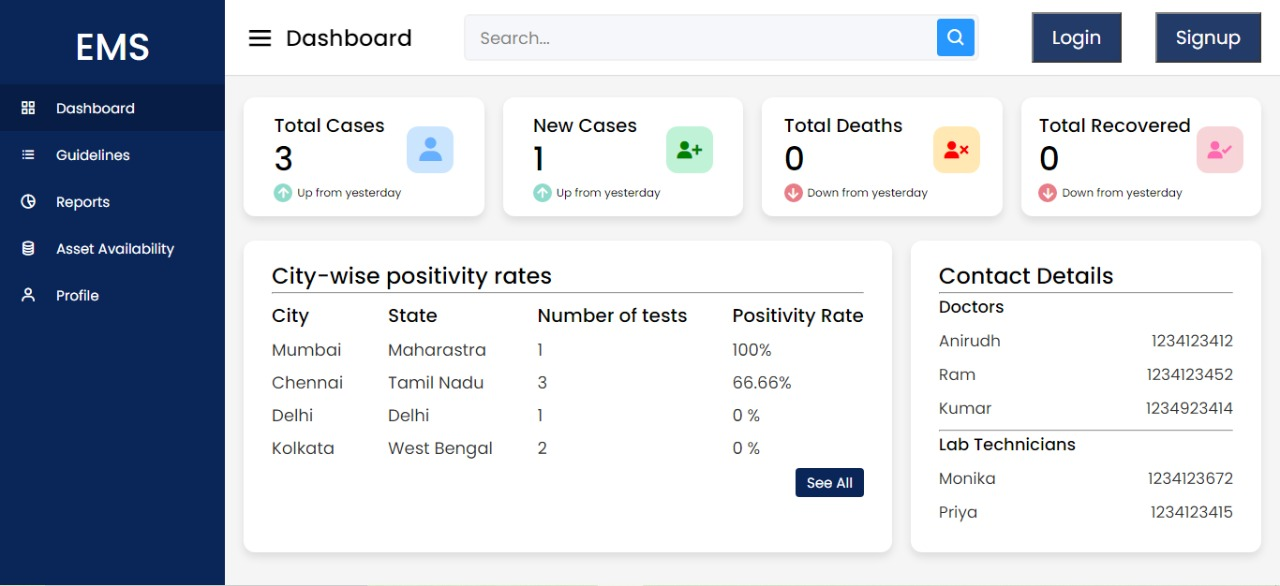
**Successful login**



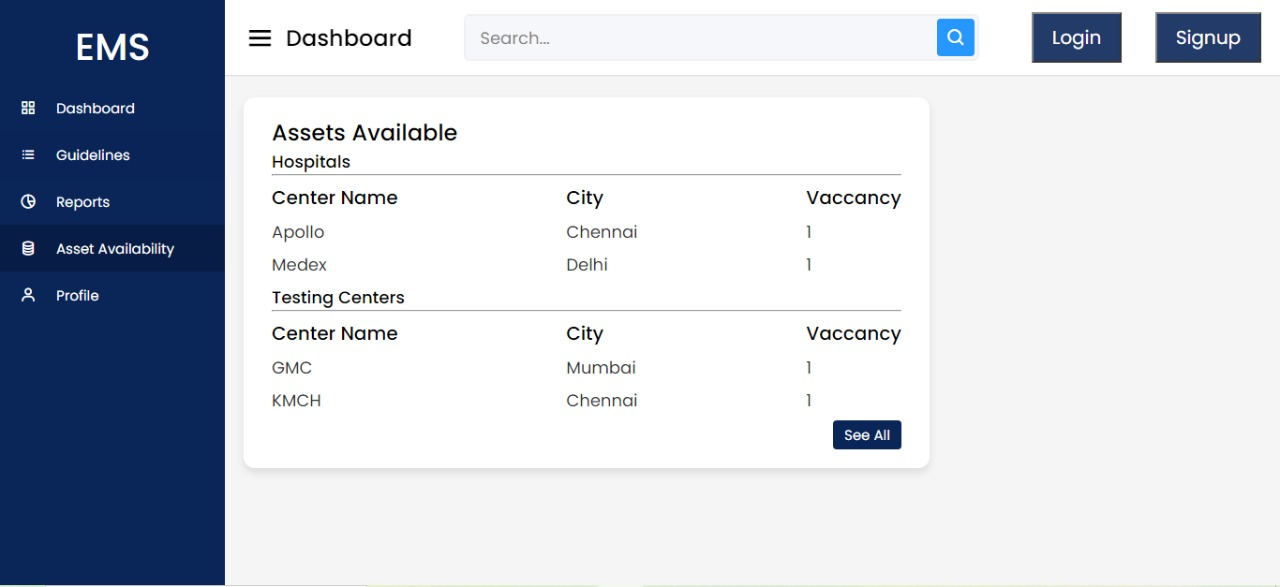
**Logged out**



**Before Login\public visibility**

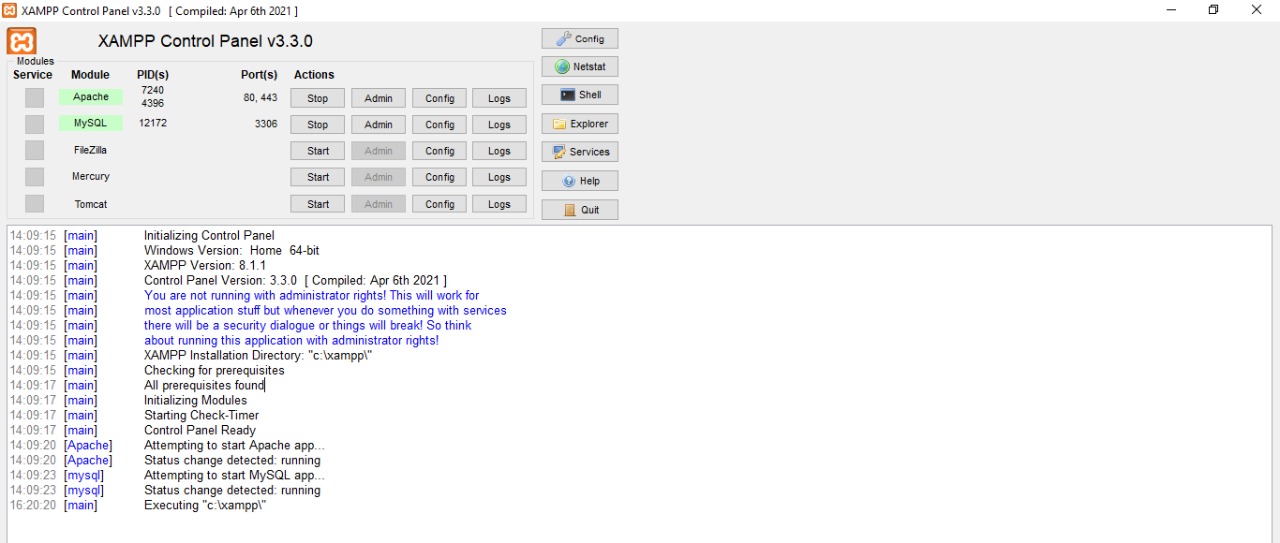


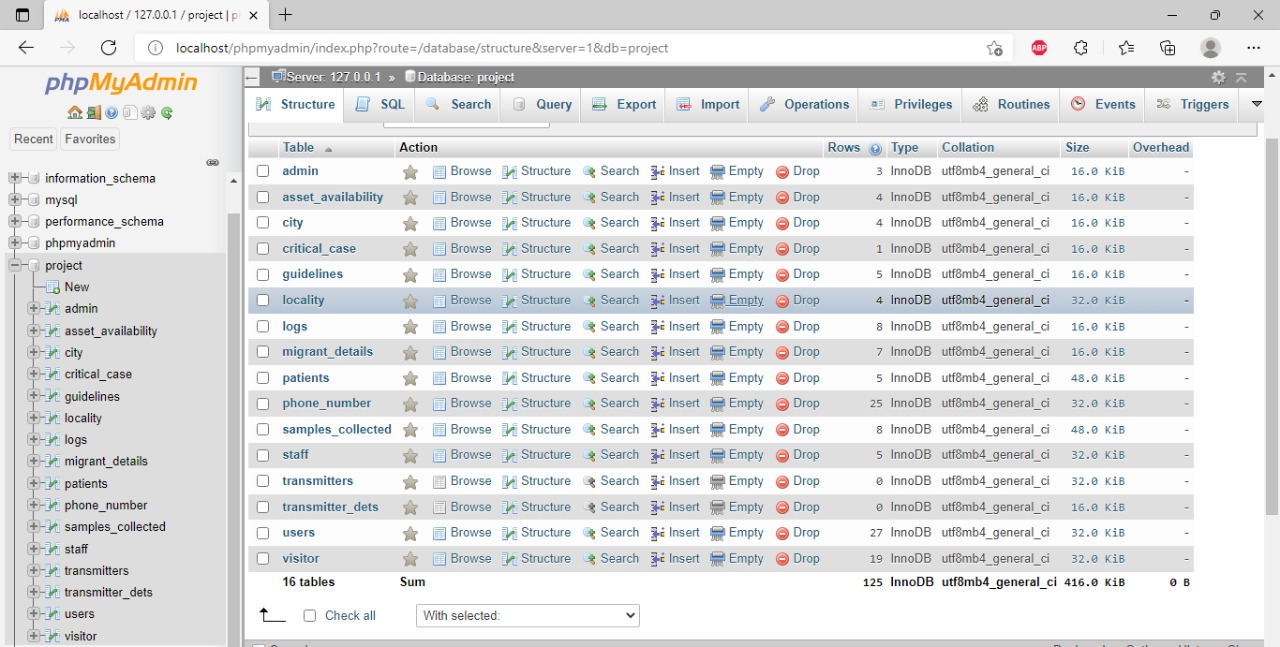
**Asset Availability**



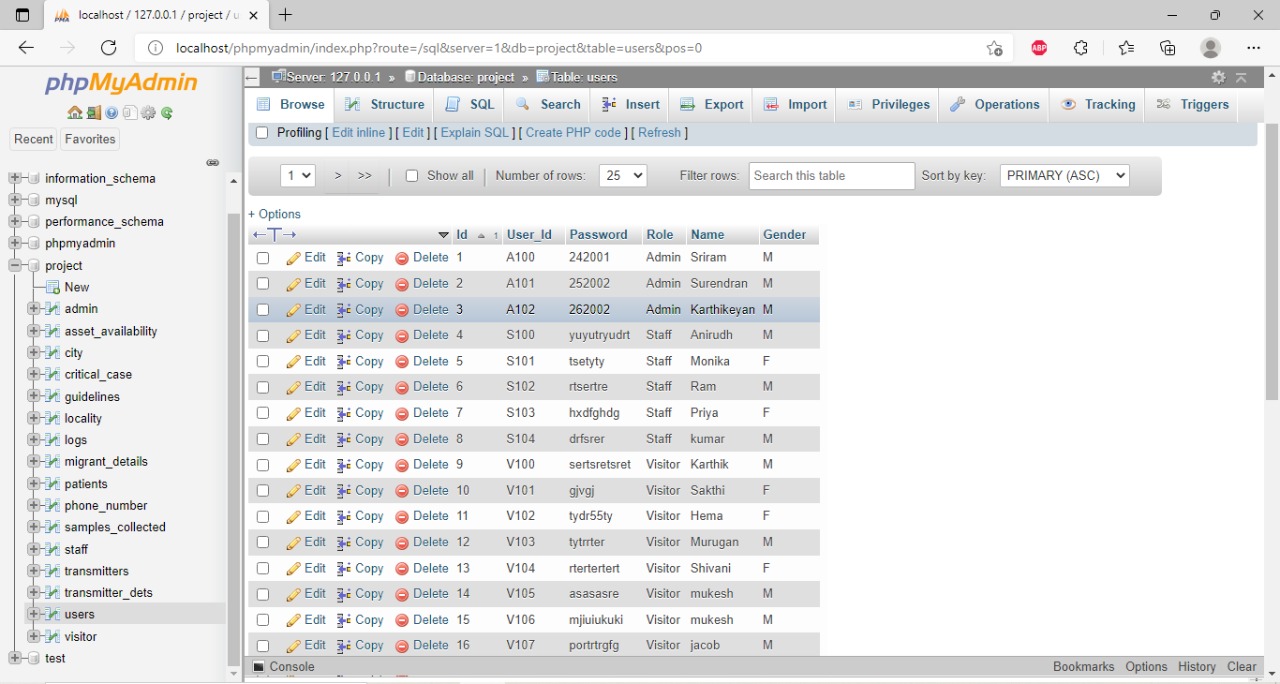
**PhpMyAdmin**

**All Tables:**

****

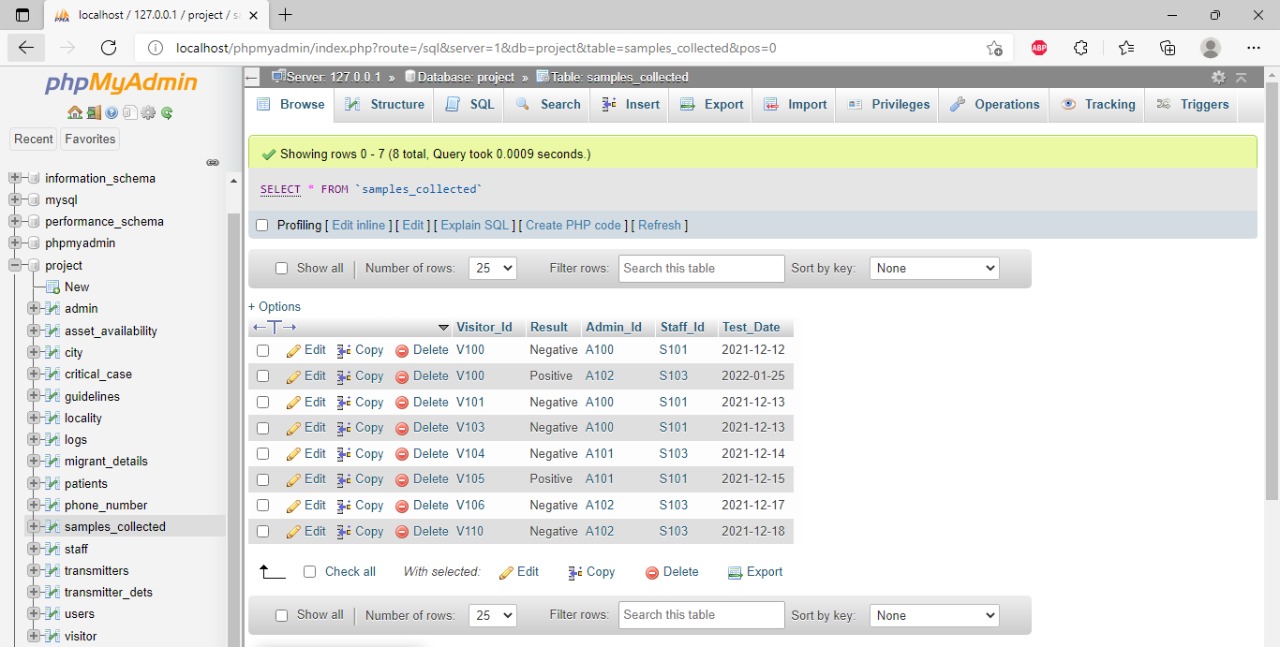
****

**Users Table:**

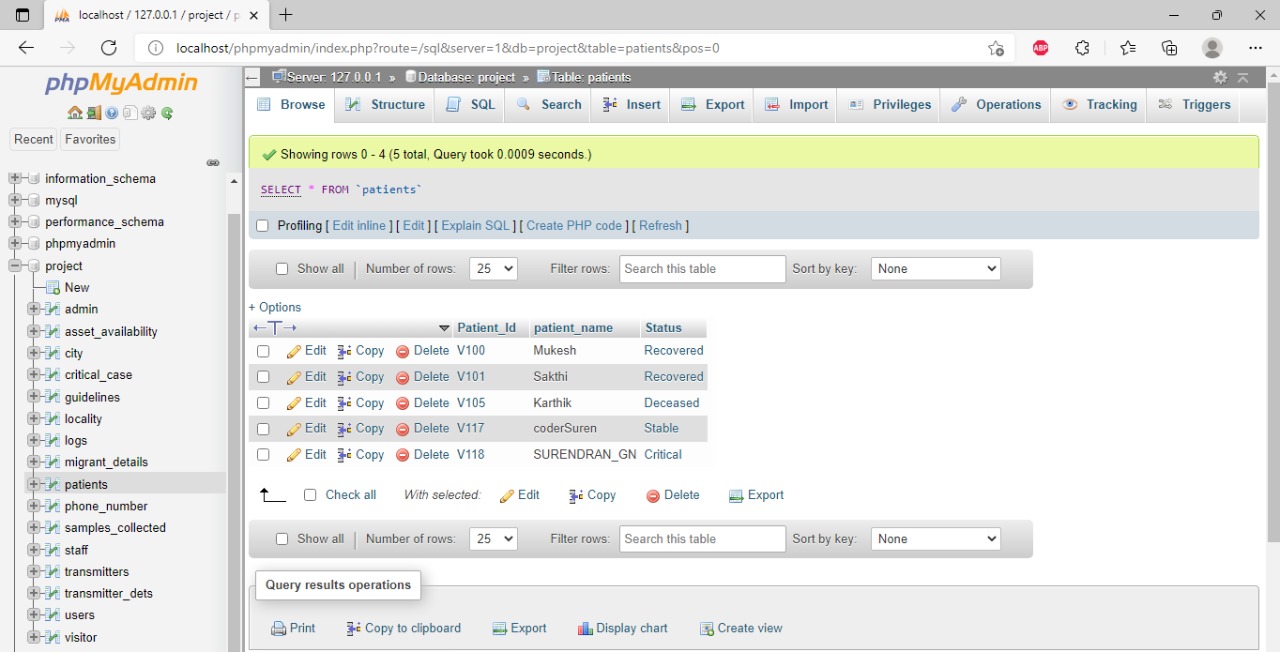
****

**Tables where Doctor/Staff upload their results and other data:**

**Samples\_Collected Table:**

****

**Patients Table:**

****

**Some of the Php Files Used in this project:**

**Server:**

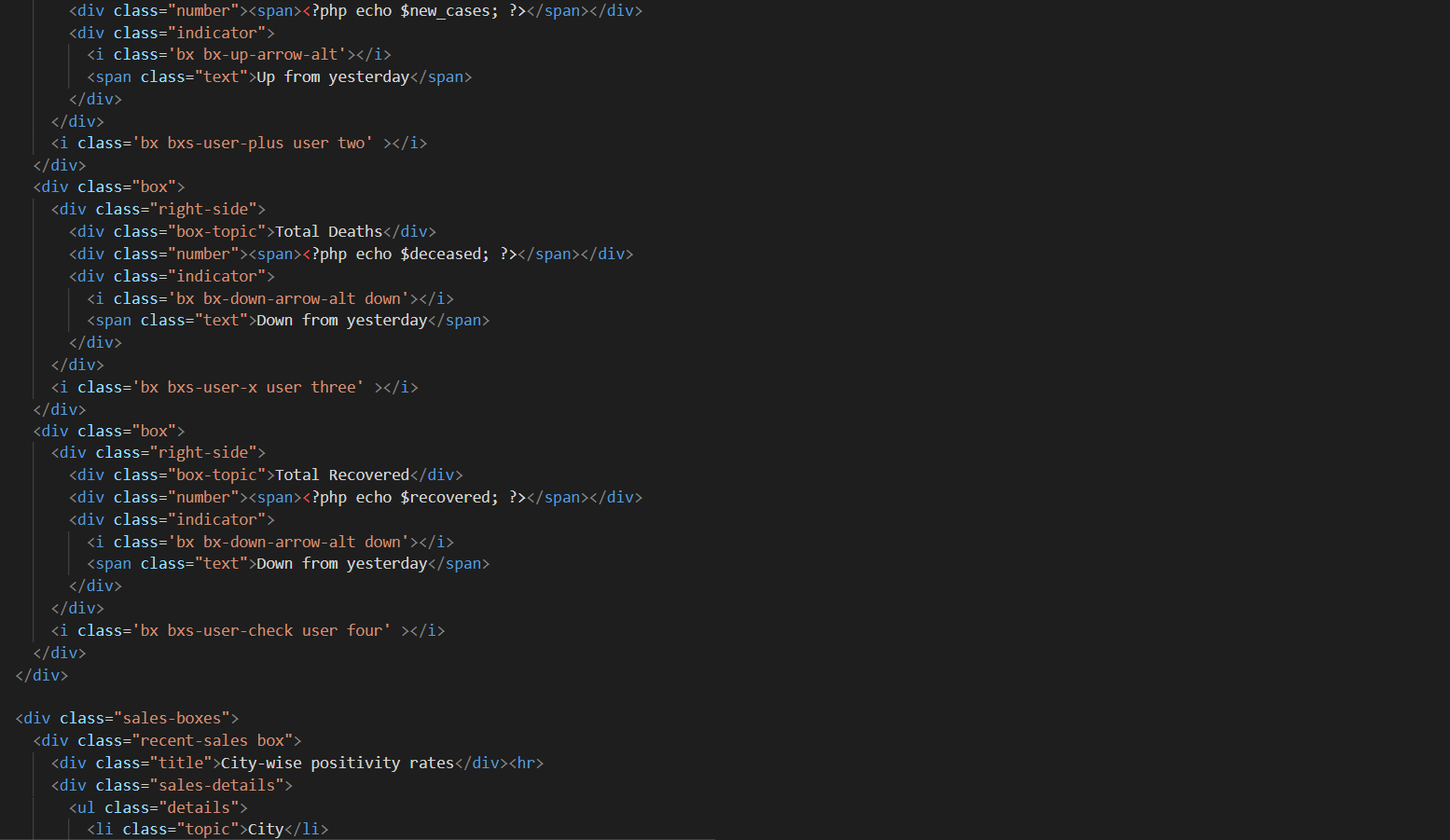
****

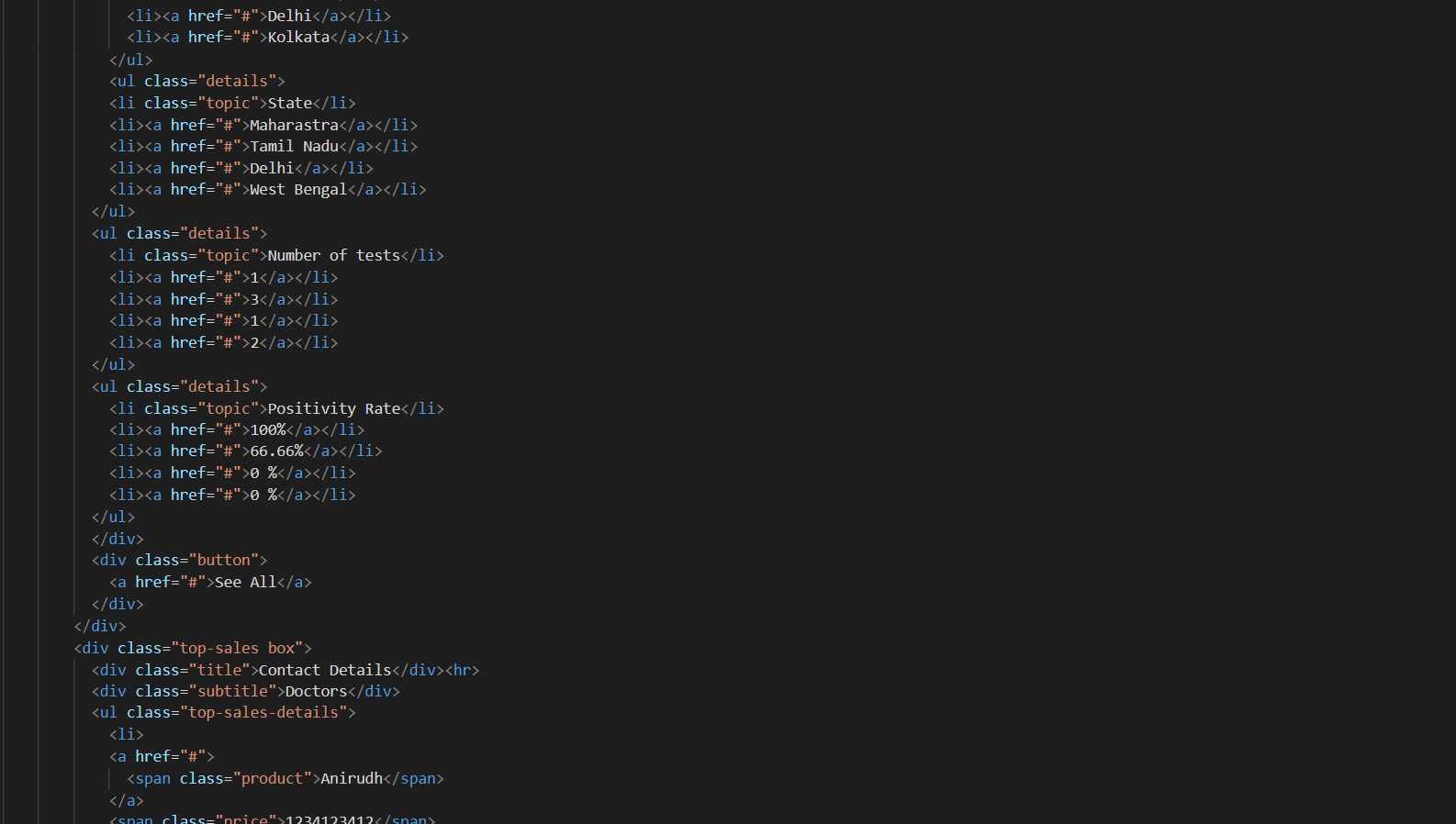
**Home:**

****

****

****

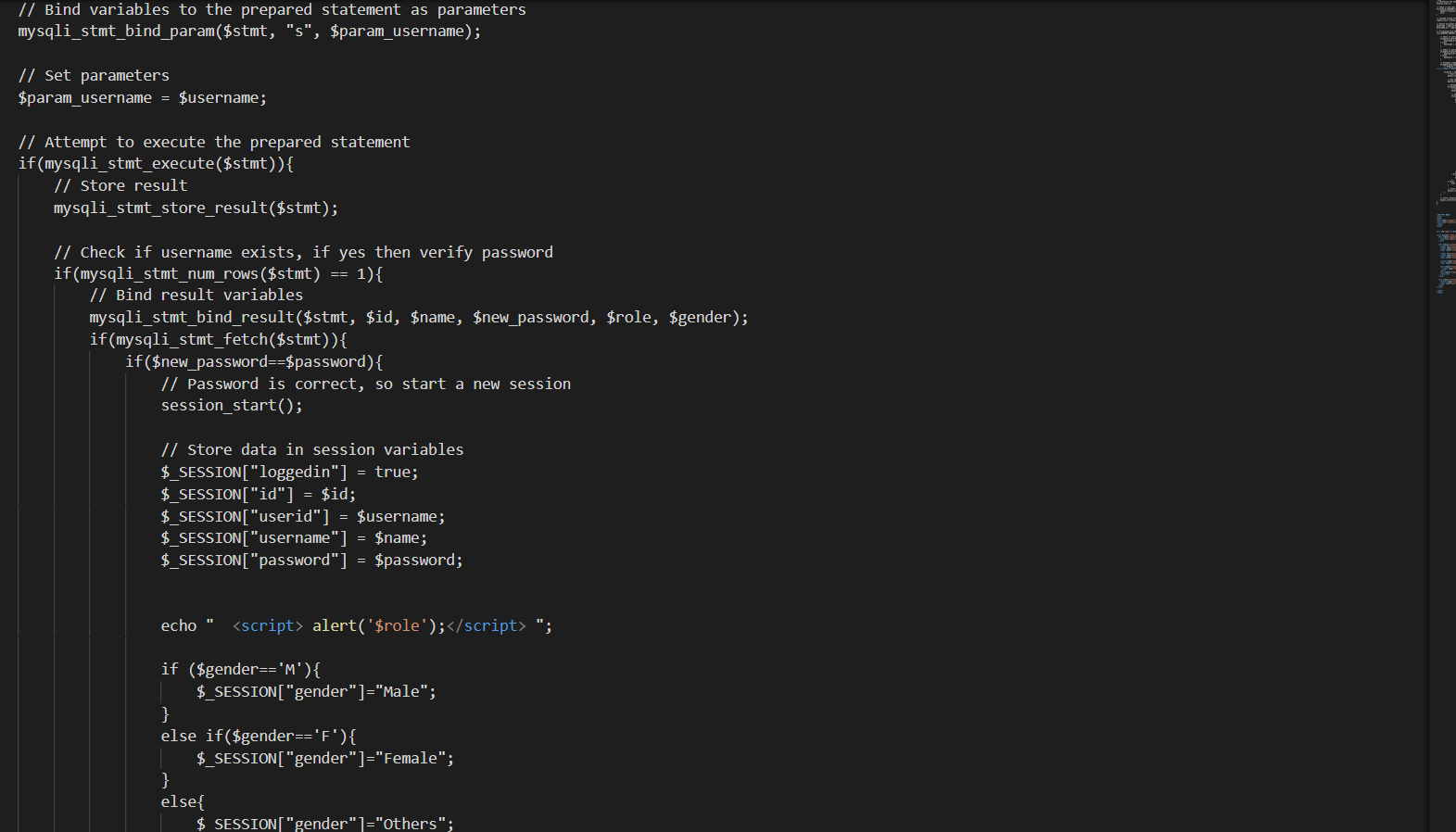
****

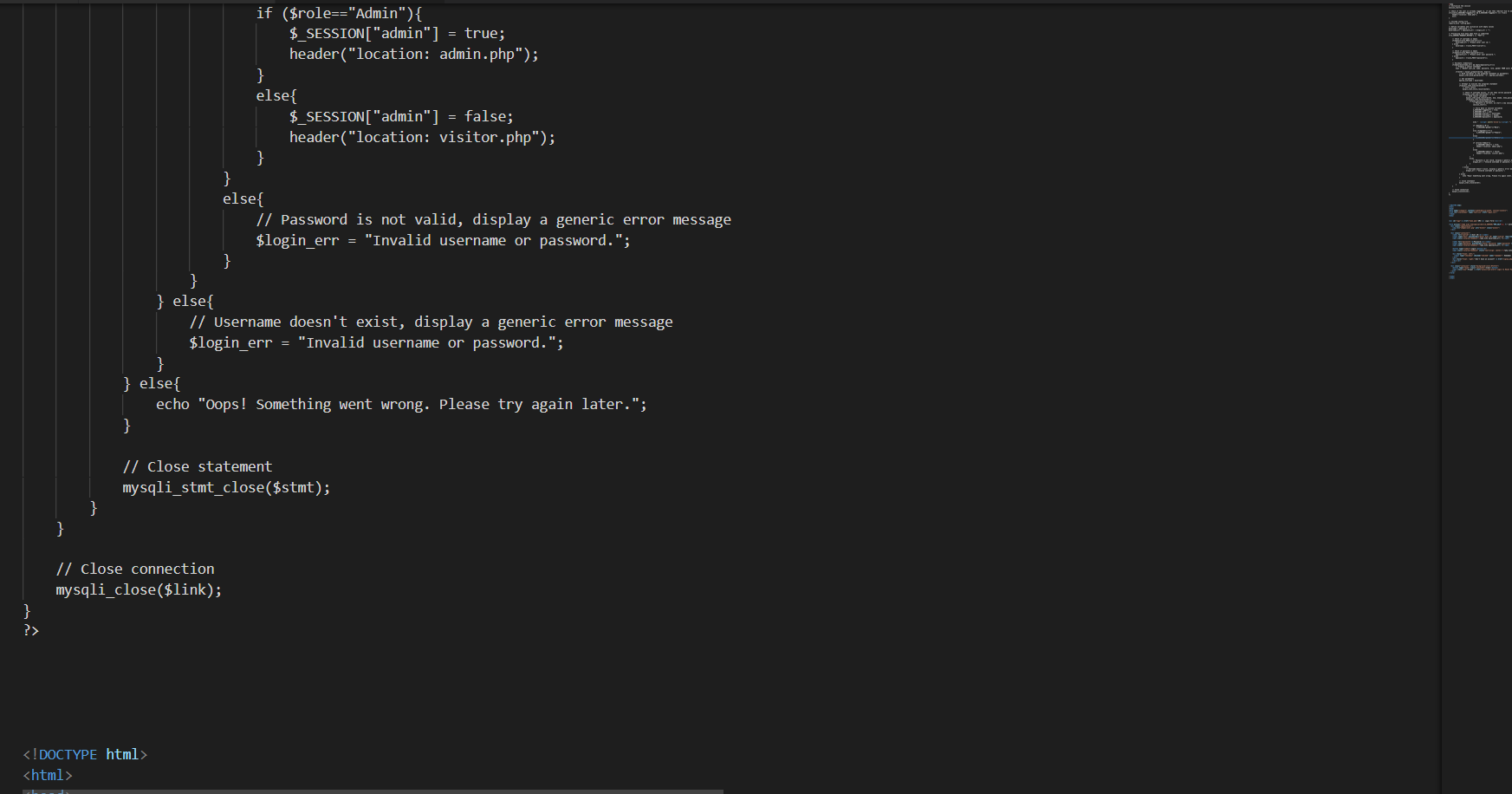
****

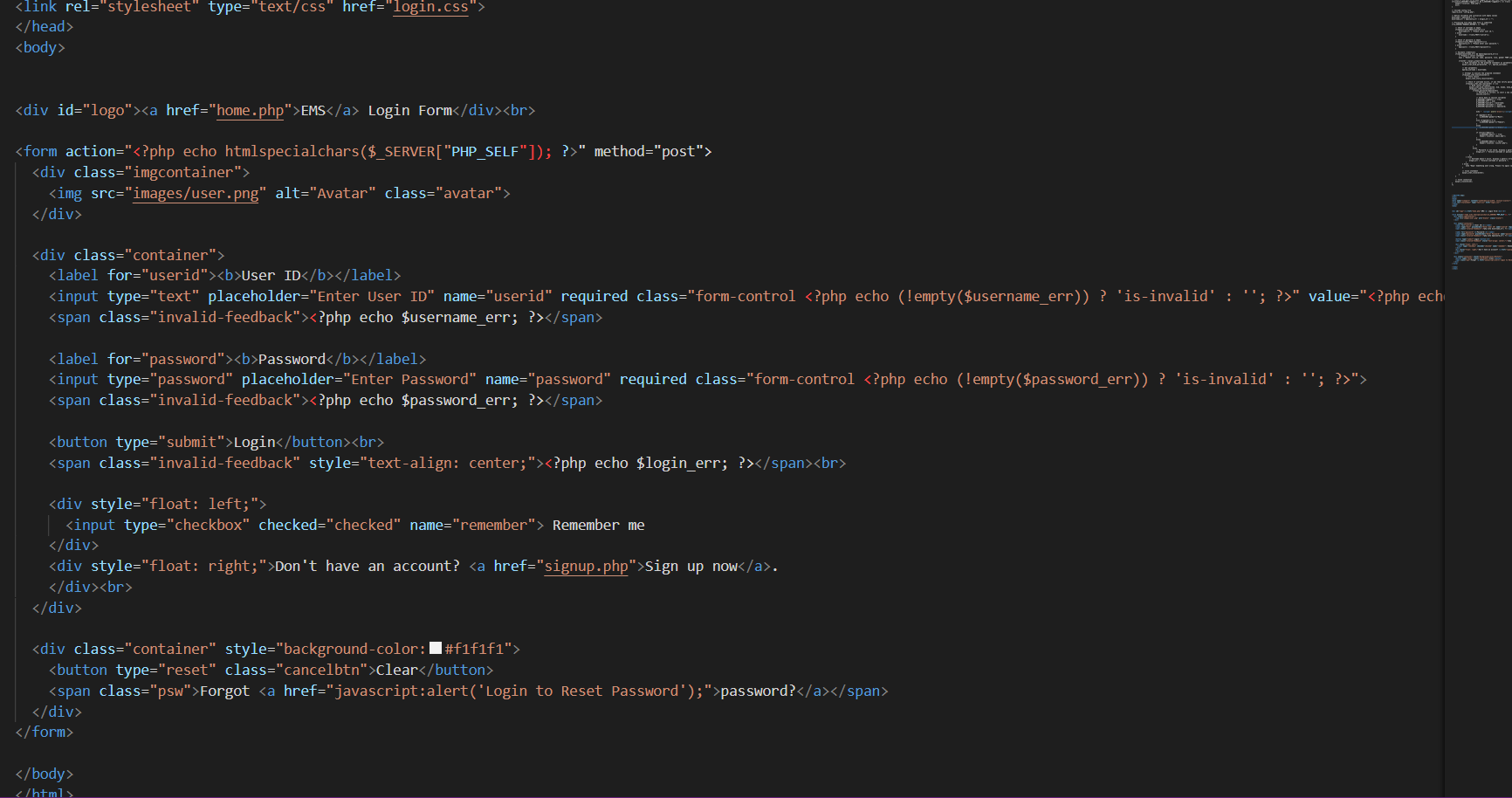
****

**Login:**

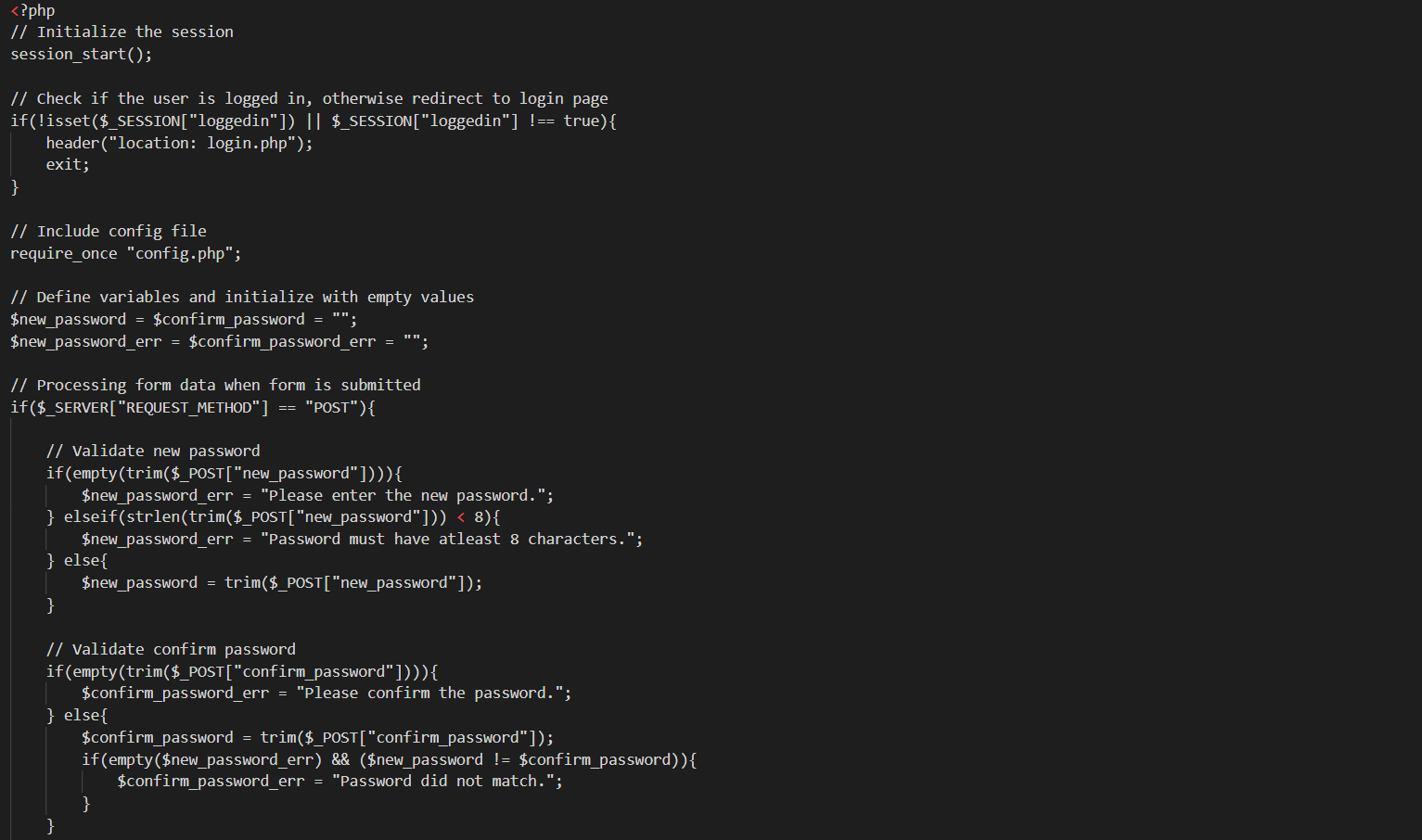
****

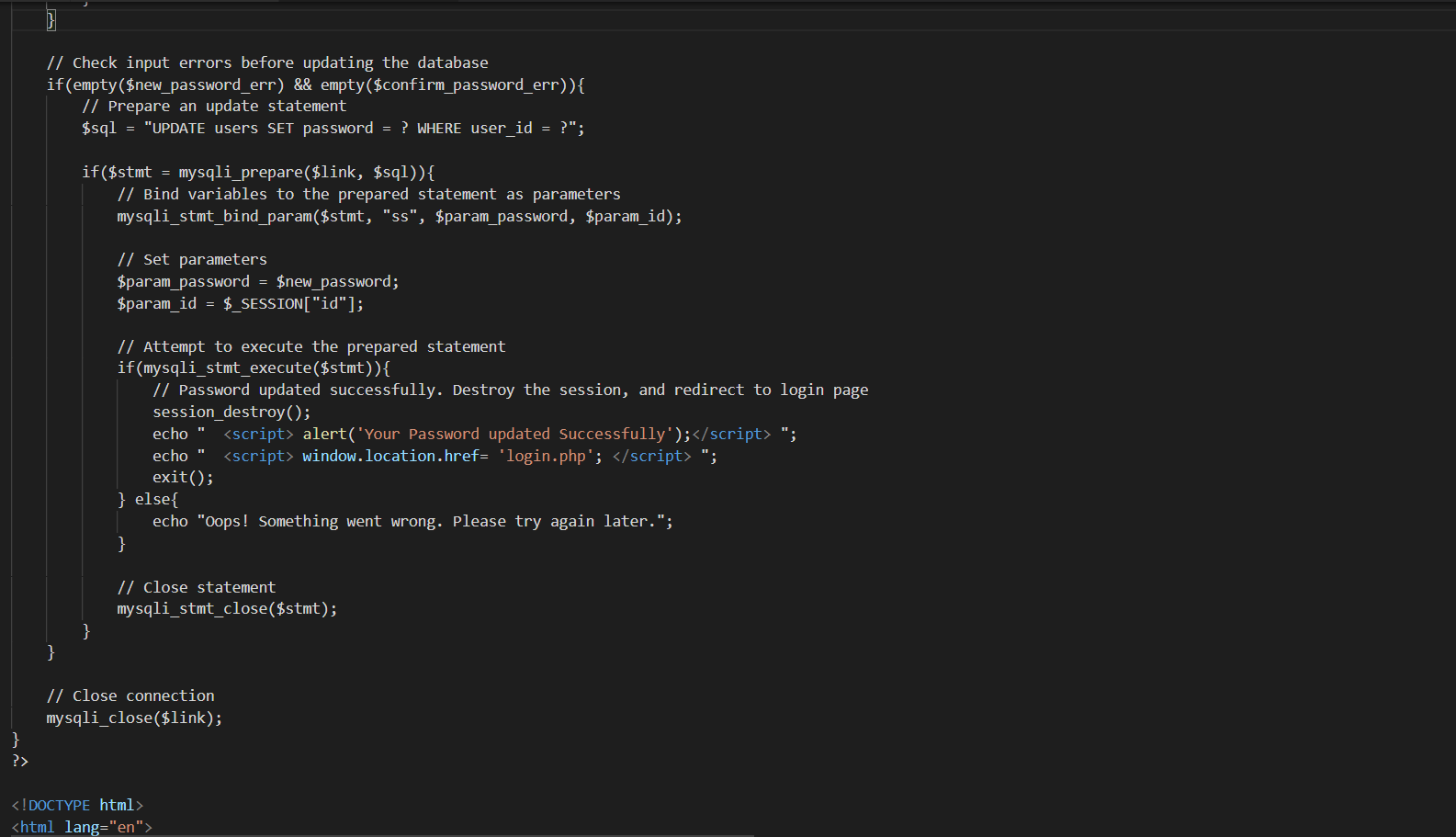
****

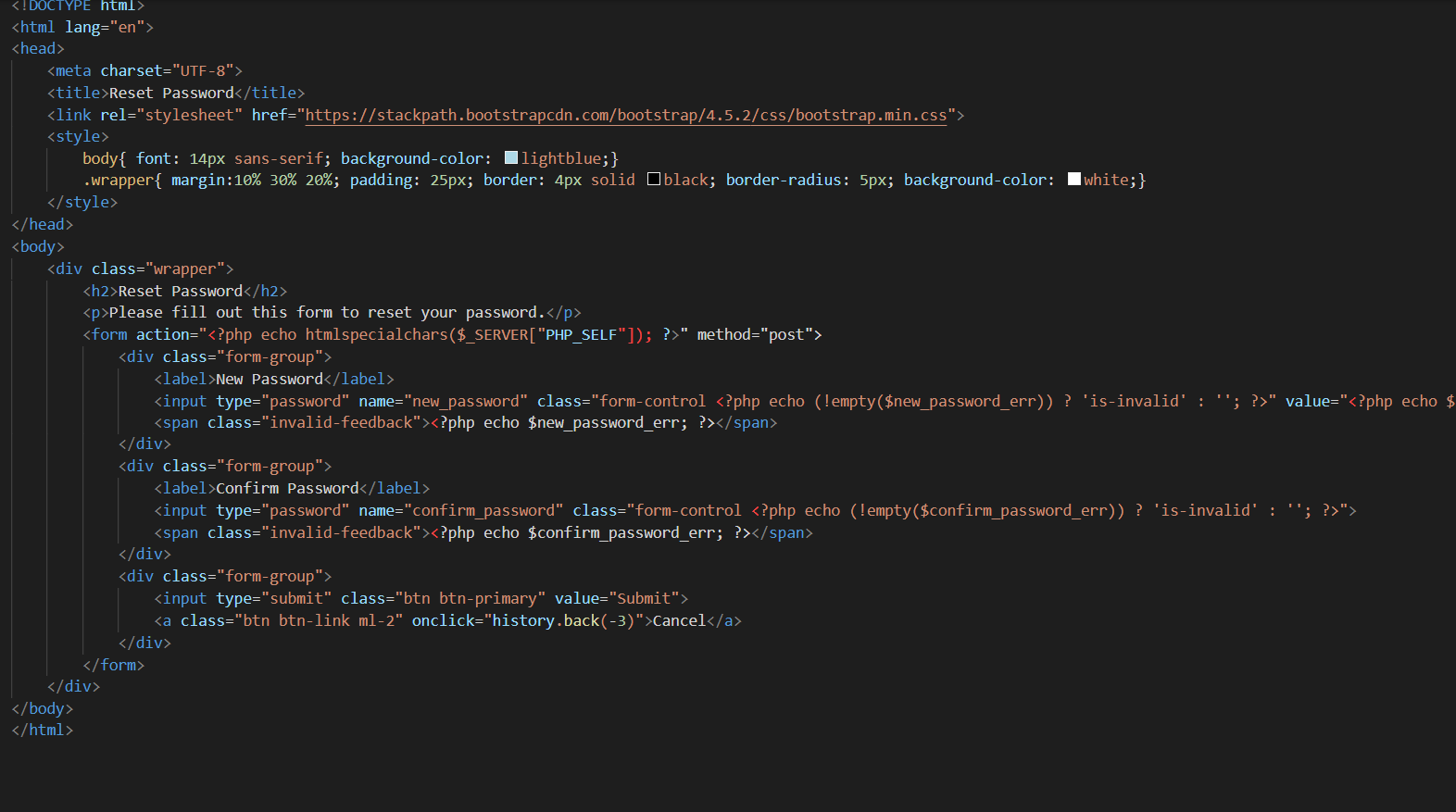
****

****

**Reset Password:**

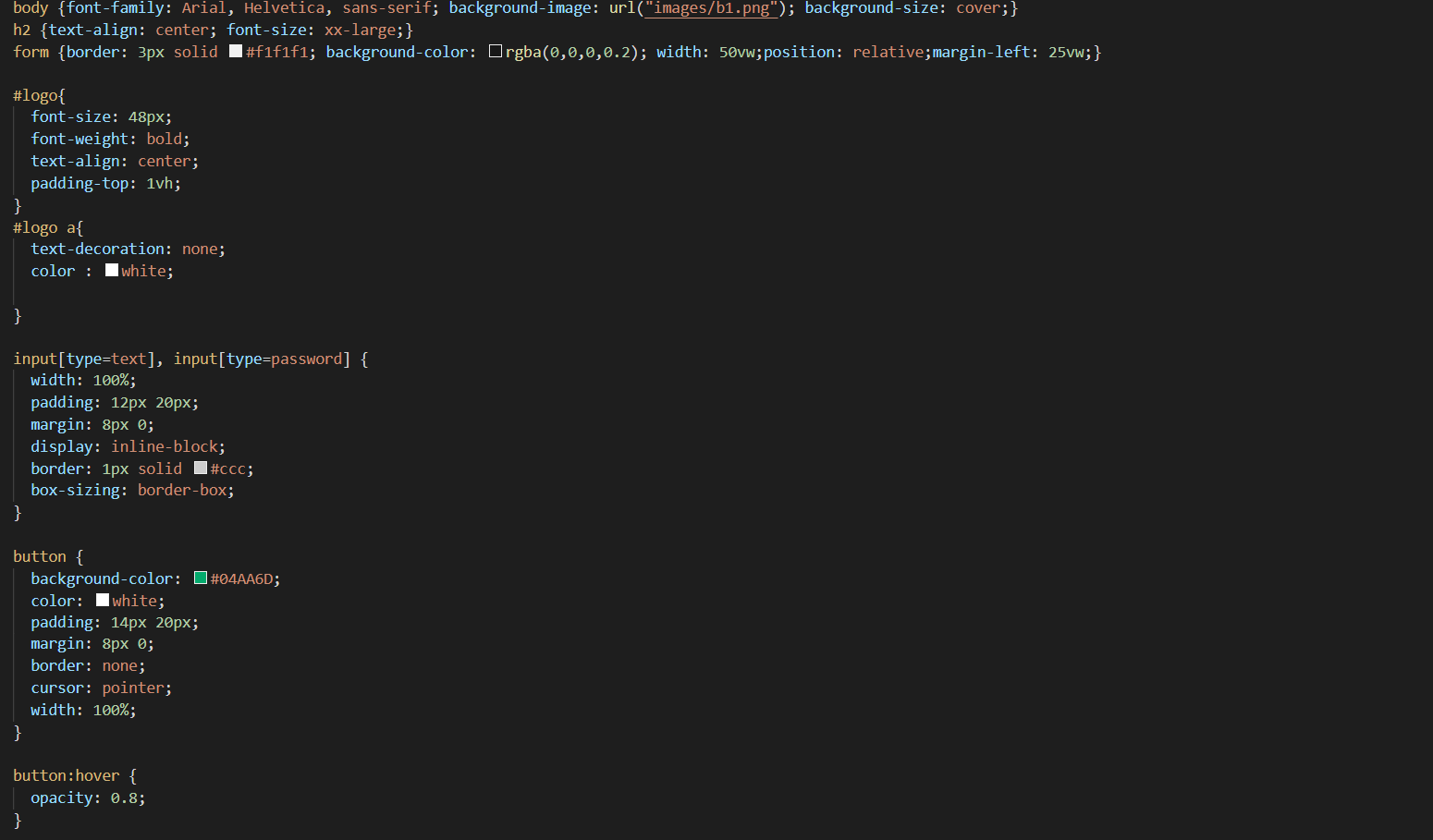
****

****

****

**Some of the CSS Part of the Project:**

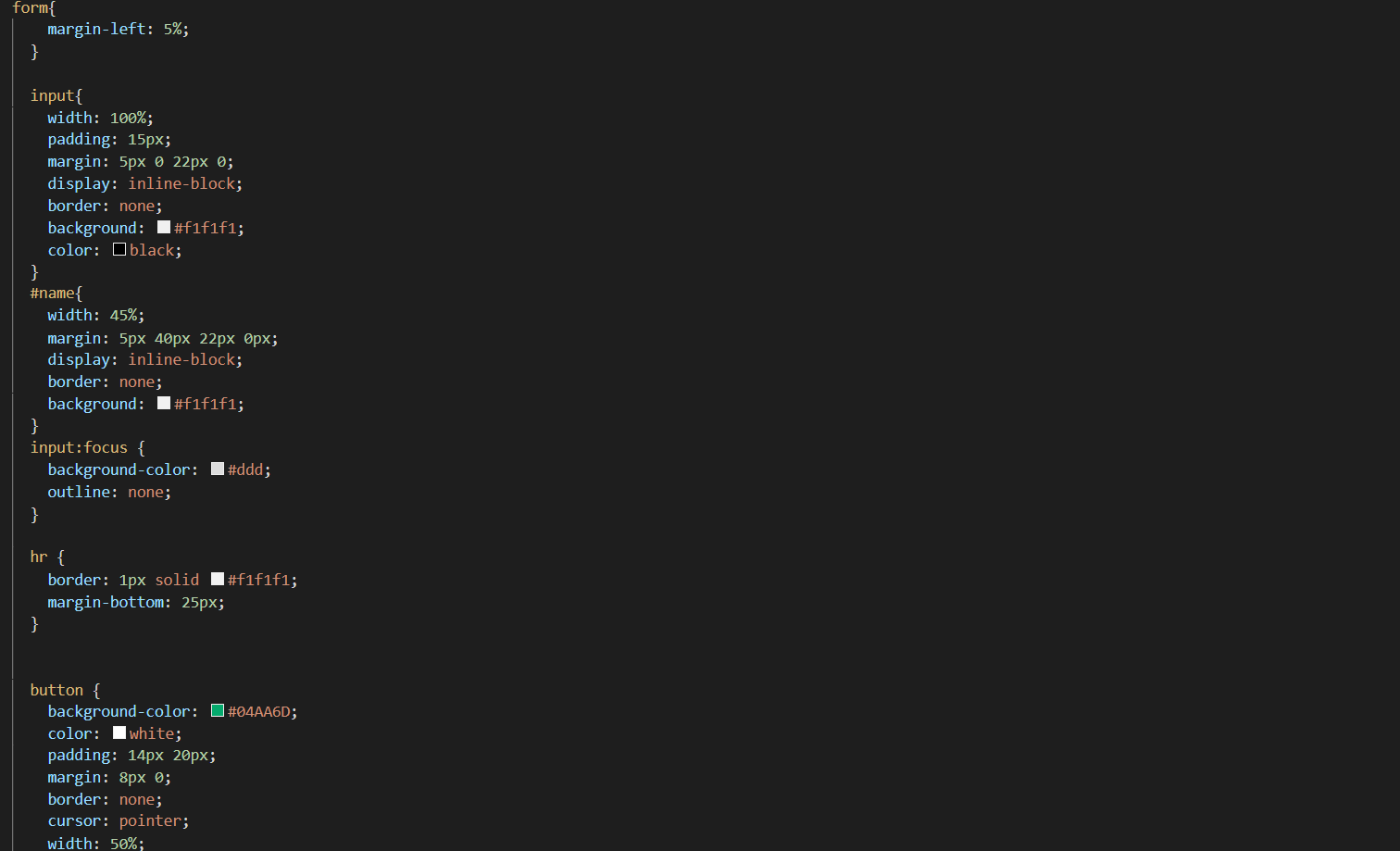
**Login:**

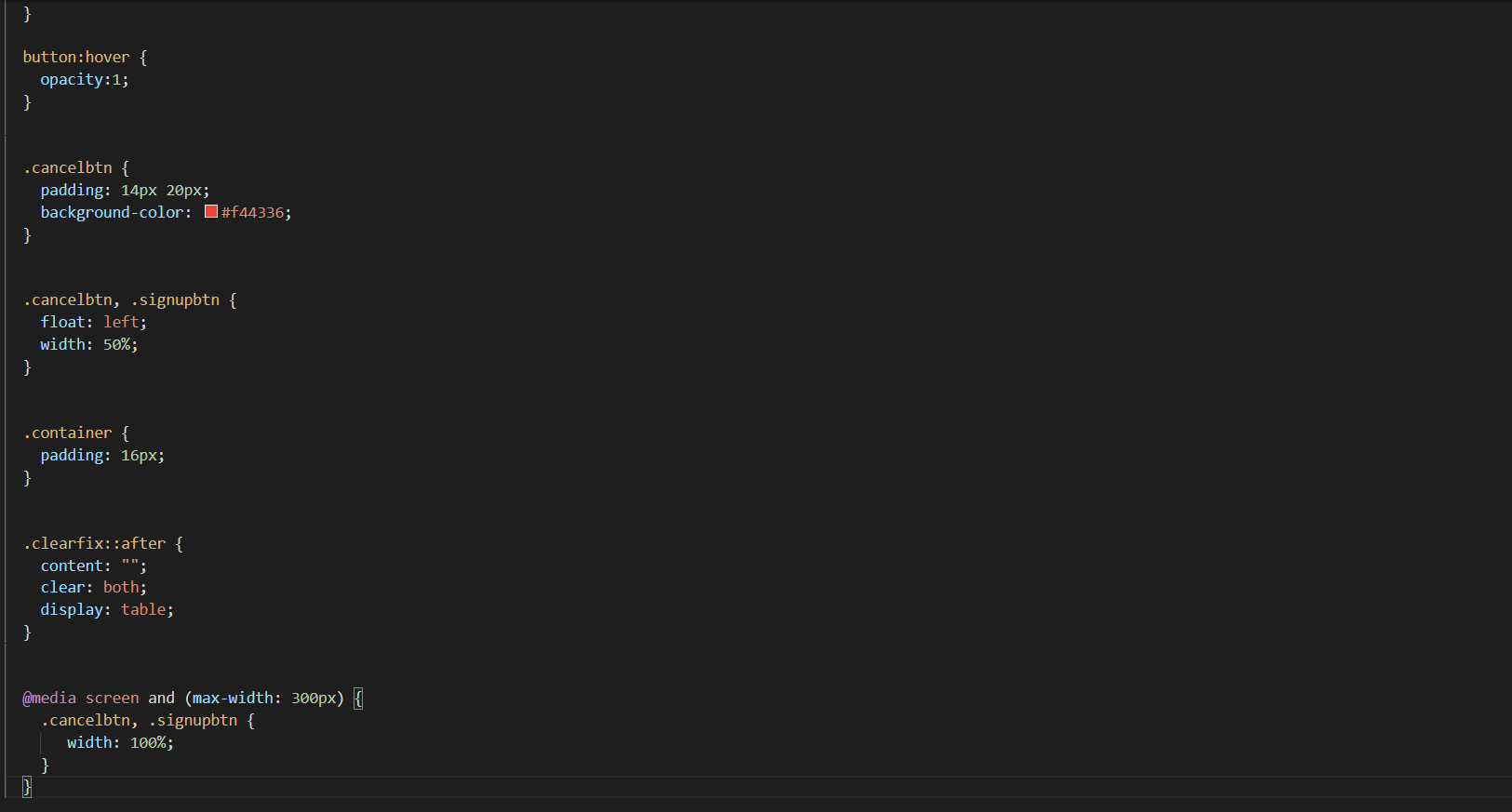
****

****

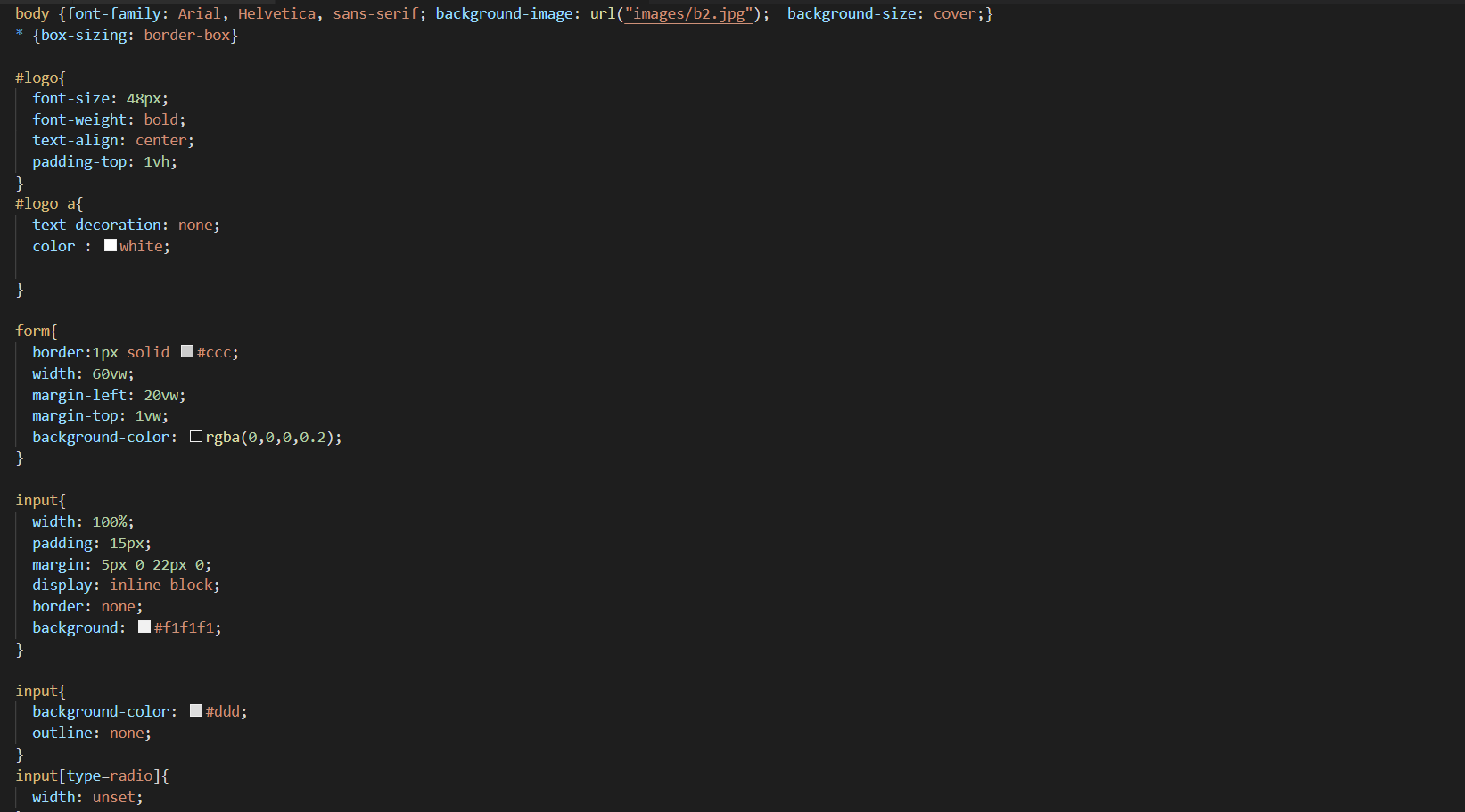
****

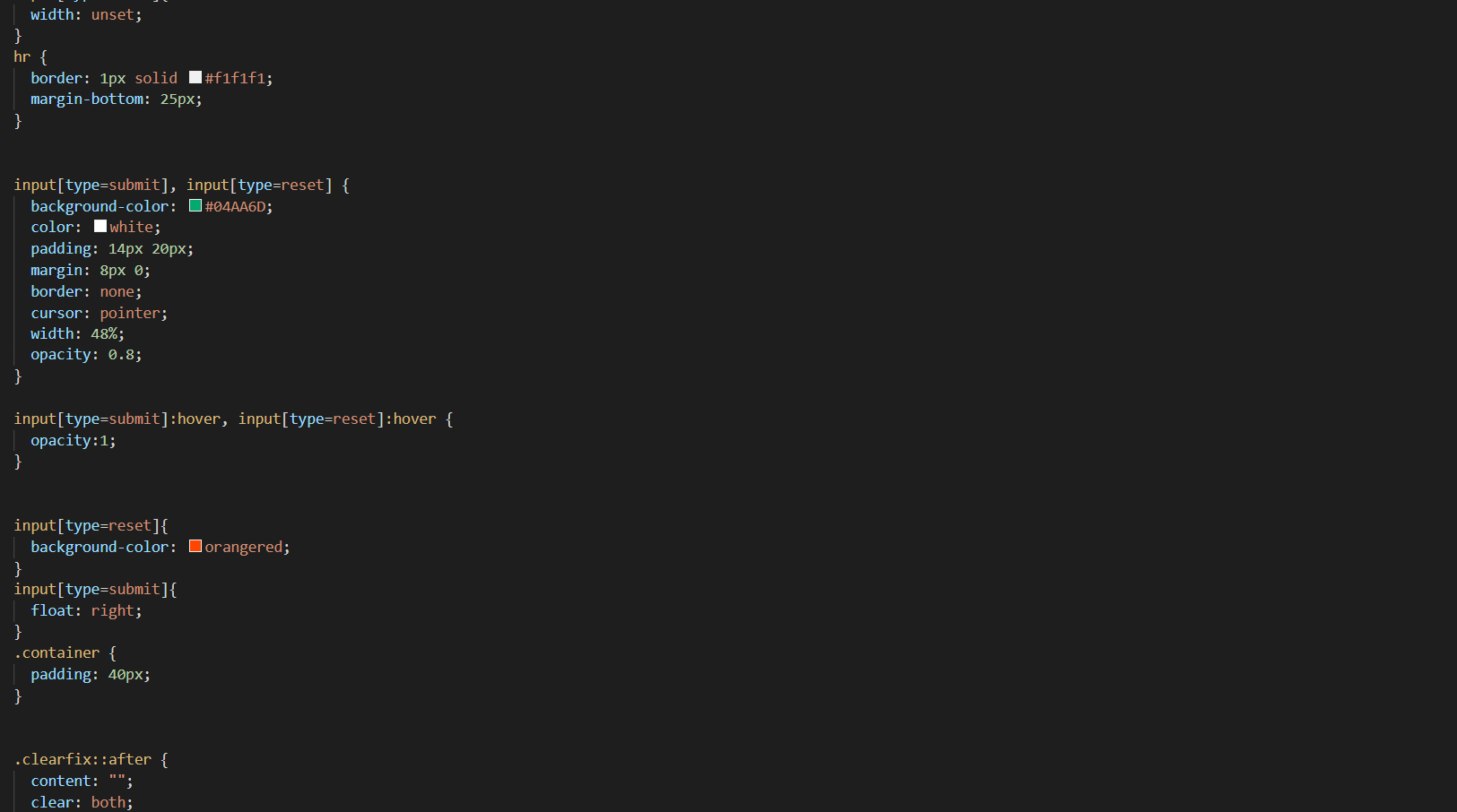
**Profile:**

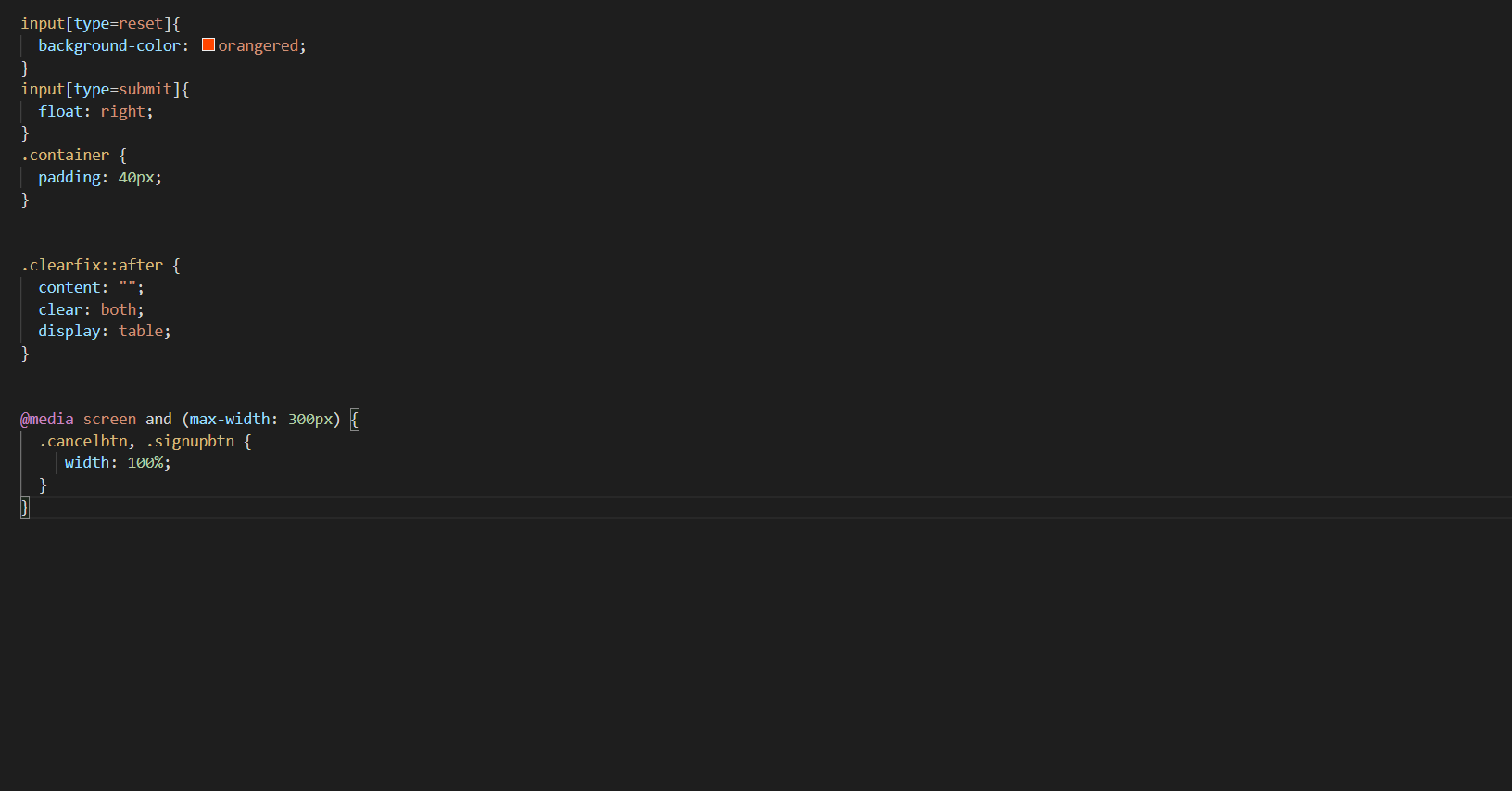
****

****

**Sign Up page:**

****

****

****