Chapter 1

INTRODUCTION

With the help of Covid-19 Management system various authorities could keep an eye on the patient's data. This project is aimed at studying and simulating the voluminous quantities of data that are stored in case of Covid-19 Pandemic. There will be portals for different authorities-hospital portal, government portal and public portal. Government portal could keep an eye on other portals and analyse the finalised data.

Hospital portal will help the hospital authorities to get the Corona patient's data and number of beds vacant in the hospital and various similar situations within which Doctors portal would help them to get the data for the doctors as required.

Public portal as the name suggests is for public and any citizen can introspect through the website data to know the status of Corona patients' recovery and death etc.

The main aim of a DBMS is to supply a way to store up and retrieve database information that is both convenient and efficient. By data, we mean known facts that can be recorded and that have embedded meaning. Normally people use software such as DBASE IV or V, Microsoft ACCESS, or EXCEL to store data in the form of database.

Database systems are meant to handle large collection of information. Management of data involves both defining structures for storage of information and providing mechanisms that can do the manipulation of those stored information. Moreover, the database system must ensure the safety of the information stored, despite system crashes or attempts at unauthorized access.

The project, "Covid-19 Patient Management System-AarogyaBharat" is also a step towards offering more or less the similar features. This system enables to manage and record the activities of all Patient handled by the Doctor and Hospital.

Chapter 2

SYSTEM REQUIREMENTS

2.1 Hardware Requirements

The hardware requirements for this project are as follows:-

- **Processor:** Intel i3 or more on Windows 7 64-bit / Windows 8 64-bit / Windows 8.1 64 bit/Windows 10 64-bit/MacOS

- **RAM:** 4GB RAM or more

- **Memory:** 256GB HDD

2.2 Software Requirements

The software requirements for this project are as follows:-

- **Operating System:** Windows 7 64-bit/Windows 10/MacOS

- **Front-end:** HTML, CSS, Bootstrap

- **Back-end:** MySQL

- **Framework:** Flask, SQLAlchemy, SQLAlchemy.orm

2.3 Functional Requirements

The functional requirements of this project are as follows:-

- **User Registration:** The system must allow people to register themselves in the registration portal

- **Admin Login:** The system must allow admin to login with their username and password in order to update Covid-19 cases and manipulate hospital data.

- **Hospital Login:** The hospital can login and see patients and doctors data under them

- **Doctor Login:** Doctors can login to see the patients under them.

2.4 Non-Functional Requirements

Some of the non-functional requirements of our proposed software are

- **Health services at ease**: The software must be quick enough to book new patient's registration assign hospitals/doctors to their needs.

- **Security:** The software must be protected against malicious users attempting to attack it using nefarious techniques like timing attacks, SQL injection, etc.

- **Reliability:** The software must be reliable enough to be expected to run 24x7 with minimal supervision

- **Data protection:** The software must ensure that the data it stores is adequately secured and cannot be tampered with

- **Data consistency:** The software must try to maintain only the bare minimum of redundant data. Wherever such redundancies are present, the software must ensure that the data is consistent.
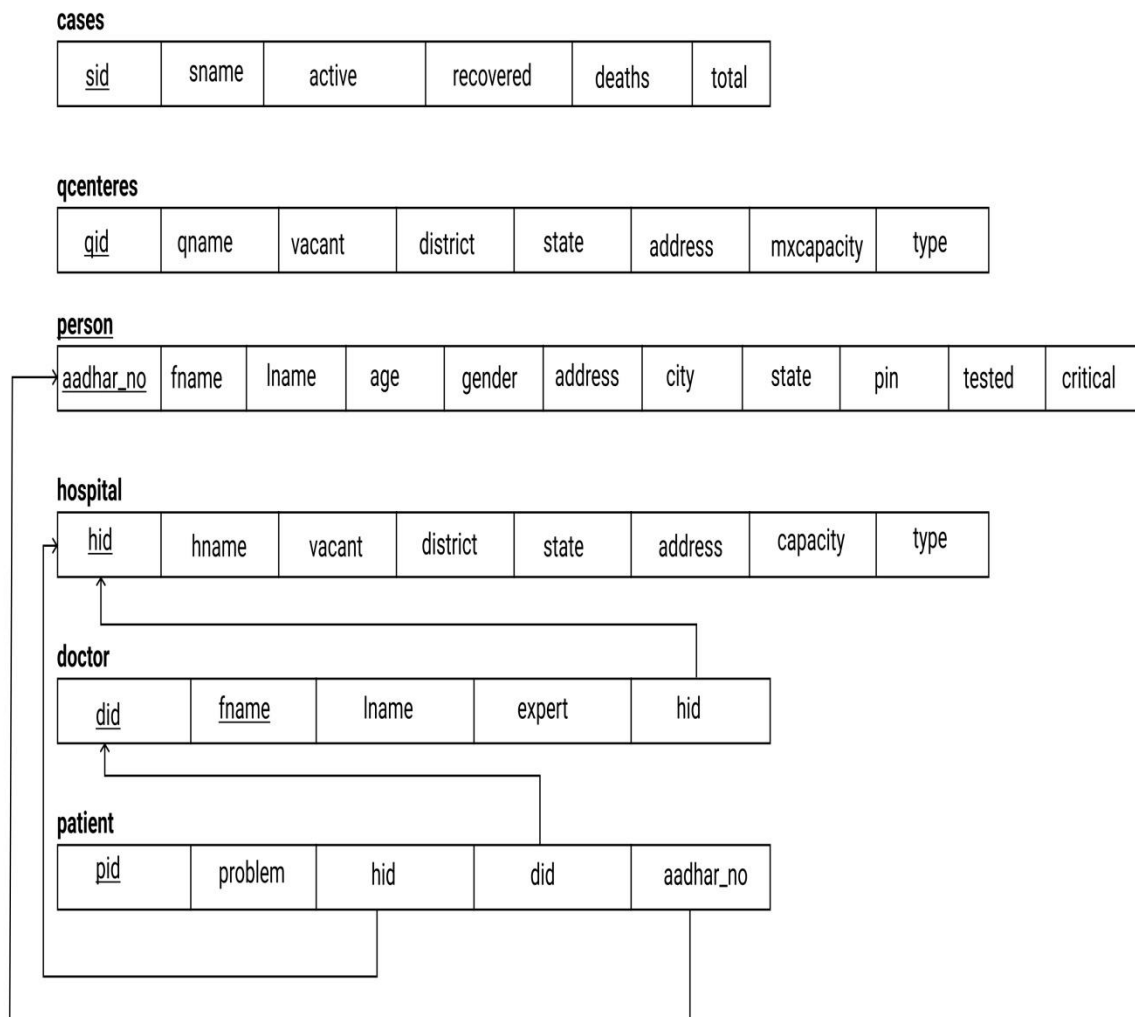
Chapter 3

SYSTEM DESIGN

3.1 DATABASE DESIGN

The data in the system has to be stored and retrieved from database. Designing the database is part of system design. Data elements and data structures to be stored have been identified at analysis stage. They are structured and put together to design the data storage and retrieval system. A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make database access easy, quick, inexpensive and flexible for the user. Relationships are established between the data items and unnecessary data items are removed. Normalization is done to get an internal consistency of data and to have minimum redundancy and maximum stability. This ensures minimizing data storage required, minimizing chances of data inconsistencies and optimizing for updates. The MS Access database has been chosen for developing the relevant databases.

3.2 SCHEMA DIAGRAM

The term "schema" refers to the organization of data as a blueprint of how the database is constructed (divided into database tables in the case of relational databases). The formal definition of a database schema is a set of formulas (sentences) called integrity constraints imposed on a database. A relational schema shows references among fields in the database. When a primary key is referenced in another table in the database, it is called a foreign key. This is denoted by an arrow with the head pointing at the referenced key attribute. A schema diagram helps organize values in the database. It also gives an idea of what order the tables should be created in. The following diagram shows the schema diagram for the database.

A schema diagram is an illustrative display of most aspects of a database schema. A schema construct is a component of the schema, or an object within the schema. The schema diagram of our database system is illustrated below, in Figure 3.2.

cases

sid	sname	active	recovered	deaths	total

qcenteres

qid	qname	vacant	district	state	address	mxcapacity	type

person

aadhar_no	fname	lname	age	gender	address	city	state	pin	tested	critical

hospital

hid	hname	vacant	district	state	address	capacity	type

doctor

did	fname	lname	expert	hid

patient

pid	problem	hid	did	aadhar_no

Figure 3.1: Schema Diagram of the Covid-19 Patient Management System

3.3 ENTITY-RELATIONSHIP DIAGRAM

An entity–relationship model is usually the result of systematic analysis to define and describe what is important to processes in an area of a business. An E-R model does not define the business processes; it only presents a business data schema in graphical form. It is usually drawn in a graphical form as boxes (entities) that are connected by lines (relationships) which express the associations and dependencies between entities.

An ER model can also be expressed in a verbal form, for example: one building may be

divided into zero or more apartments, but one apartment can only be located in one building. Entities may be characterized not only by relationships, but also by additional properties (attributes), which include identifiers called "primary keys" Diagrams created to represent attributes as well as entities and relationships may be called entity-attribute relationship diagram.

In a relational database a relationship between entities is implemented by storing the primary key of one entity as a pointer or "foreign key" in the table of another entity. There is a tradition for ER/data models to be built at two or three levels of abstraction. Note that the conceptual-logical-physical hierarchy below is used in other kinds of specification, and is different from the three-schema approach to software engineering.

The four main cardinal relationships are:

- **1:N**

  o Many patients can book one hospital and one hospital can have multiple patients but one patient can be in one hospital.

- **1:1**

  o One person can become one patient.

- **M:N**

  o One doctors can handle multiple patients and one patient can be under multiple doctors.

  o One doctor can work under multiple hospitals and one hospital can have multiple doctors.

The ER diagram for the project is shown in Figure 3.1 below:
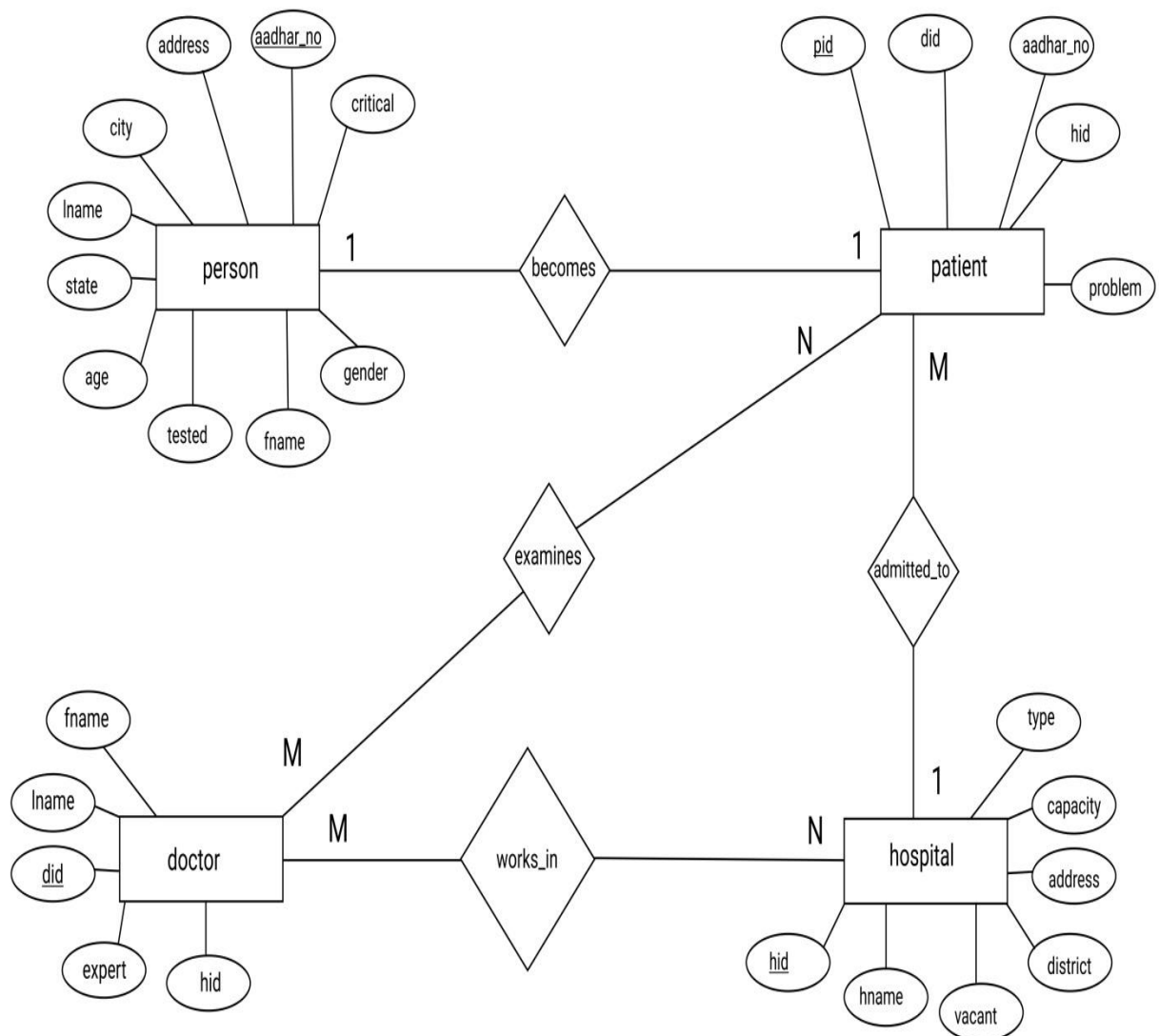
Figure 3.2: ER Diagram of the Covid-19 Patient Management System

Chapter 4

IMPLEMENTATION

4.1 Relations

4.1.1 Cases

This table shows the cases of Covid-19 in different states of India. When user enters in the website the contents of this table is shown in the home page. The full list of the relation's attributes is given in Figure 4.1 below.

```
mysql> desc cases;
+-----------+---------------------+------+-----+---------+----------------+
| Field     | Type                | Null | Key | Default | Extra          |
+-----------+---------------------+------+-----+---------+----------------+
| sid       | bigint(20) unsigned | NO   | PRI | NULL    | auto_increment |
| sname     | varchar(25)         | NO   |     | NULL    |                |
| active    | bigint(20)          | NO   |     | NULL    |                |
| recovered | bigint(20)          | NO   |     | NULL    |                |
| deaths    | bigint(20)          | NO   |     | NULL    |                |
| total     | bigint(20)          | NO   |     | NULL    |                |
+-----------+---------------------+------+-----+---------+----------------+
6 rows in set (0.02 sec)
```
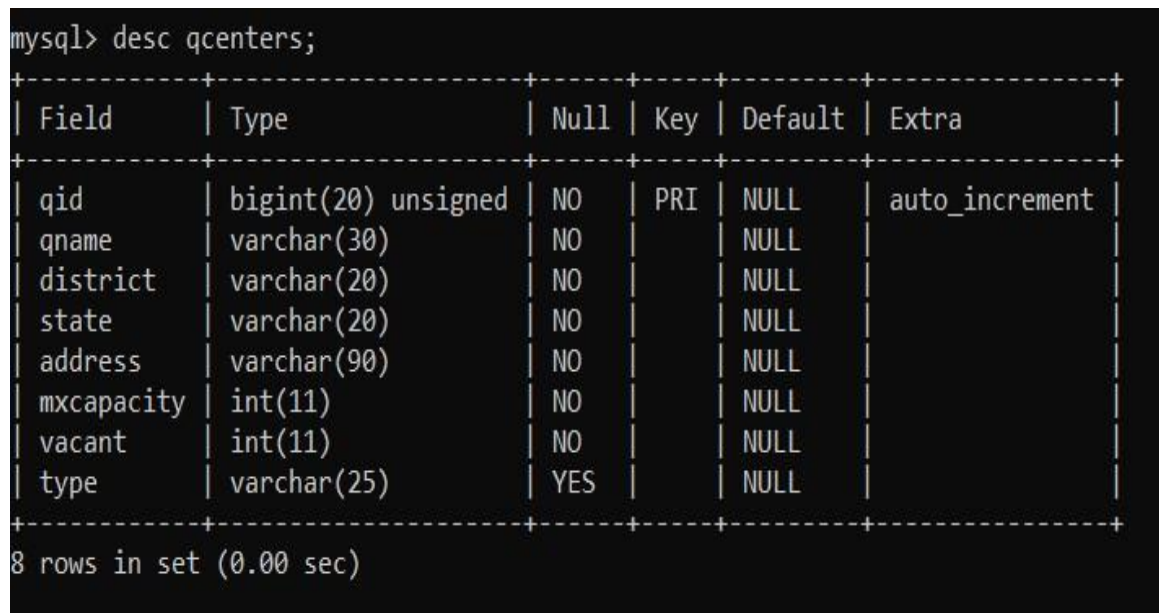
Figure 4.1: Attributes of the relation 'cases'

create table cases(sid serial not

null primary key, sname

varchar(25) not null, active bigint

not null, recovered bigint not null,

deaths bigint not null, total bigint

not null);

4.1.2 Qcenters

The quarantine centre table gives the data about the quarantine centres near the user locality. If the user registers oneself as tested positive for Covid-19 but condition is not critical then they are redirected to the Quarantine centres detail page. The attributes of this relation are shown in Figure 4.2 below:

```
mysql> desc qcenters;
+-----------+---------------------+------+-----+---------+----------------+
| Field     | Type                | Null | Key | Default | Extra          |
+-----------+---------------------+------+-----+---------+----------------+
| qid       | bigint(20) unsigned | NO   | PRI | NULL    | auto_increment |
| qname     | varchar(30)         | NO   |     | NULL    |                |
| district  | varchar(20)         | NO   |     | NULL    |                |
| state     | varchar(20)         | NO   |     | NULL    |                |
| address   | varchar(90)         | NO   |     | NULL    |                |
| mxcapacity| int(11)             | NO   |     | NULL    |                |
| vacant    | int(11)             | NO   |     | NULL    |                |
| type      | varchar(25)         | YES  |     | NULL    |                |
+-----------+---------------------+------+-----+---------+----------------+
8 rows in set (0.00 sec)
```

Figure 4.2: Attributes of the relation 'qcenters'

create table qcenters(qid serial not null
primary key,qname varchar(30) not
null,district varchar(20) not null,state
varchar(20) not null,address varchar(90) not
null,mxcapacity int not null,vacant int not
null,type varchar(25));

4.1.3 Hospitals

The hospital table contains the information of hospital in the area. The exact attributes of the relation are given in Figure 4.3

Figure 4.3: Attributes of the relation 'hospital'

create table hospital(hid serial not

null primary key,hname varchar(50)

not null,district varchar(20) not

null,state varchar(20) not

null,address varchar(50) not

null,capacity int not null,vacant int

4.1.4 Person

When a person registers in the website using the registration form then person table captures the information as shown in the figure 4.4

Figure 4.4: Attributes of the relation 'person'

create table person(aadhar_no bigint
primary key, fname varchar(50) not null,
lname varchar(50), age int not null, gender
varchar(10), address varchar(100), city
varchar(50), state varchar(50), pincode int
not null, tested char(1) not null, critical
char(1) not null);

4.1.5 Patient

The 'patient' table gives the data about the registered person as a patient and in which hospital he/she is admitted to. The attribute of this table are given in Figure 4.5 below

```
mysql> desc patient;
+---------+-------------------+------+-----+---------+----------------+
| Field   | Type              | Null | Key | Default | Extra          |
+---------+-------------------+------+-----+---------+----------------+
| pid     | bigint(20) unsigned | NO   | PRI | NULL    | auto_increment |
| hid     | bigint(20) unsigned | YES  | MUL | NULL    |                |
| problem | varchar(20)       | NO   |     | NULL    |                |
| adhno   | bigint(20)        | YES  | MUL | NULL    |                |
+---------+-------------------+------+-----+---------+----------------+
4 rows in set (0.00 sec)
```

Figure 4.5: Attributes of the relation 'patient'

create table patient (pid serial not null,hid
bigint unsigned ,problem varchar(50) not
null,adhno bigint ,foreign key hid
references hospitals(hid) on delete cascade,
foreign key adhno references
person(aadhar_no) on delete cascade);

4.1.6 Doctor

This table contains the information about doctors and the hospitals he's working in.

```
mysql> desc doctor;
+--------+---------------------+------+-----+---------+----------------+
| Field  | Type                | Null | Key | Default | Extra          |
+--------+---------------------+------+-----+---------+----------------+
| did    | bigint(20) unsigned | NO   | PRI | NULL    | auto_increment |
| fname  | varchar(15)         | NO   |     | NULL    |                |
| lname  | varchar(15)         | NO   |     | NULL    |                |
| expert | varchar(15)         | NO   |     | NULL    |                |
| hid    | bigint(20) unsigned | YES  | MUL | NULL    |                |
+--------+---------------------+------+-----+---------+----------------+
5 rows in set (0.02 sec)
```

Figure 4.5: Attributes of the relation 'doctor'

create table doctor(did serial not

null primary key,fname

varchar(15) not null,lname

varchar(15) not null,expert

varchar(15) not null,hid bigint

unsigned, foreign key hid

references hospitals(hid) on

delete cascade);

Chapter 5

SNAPSHOTS

Fig 5.1 below shows the starting page of the project. From here, one can choose to access the different portal, like hospital login, admin login, user registration.
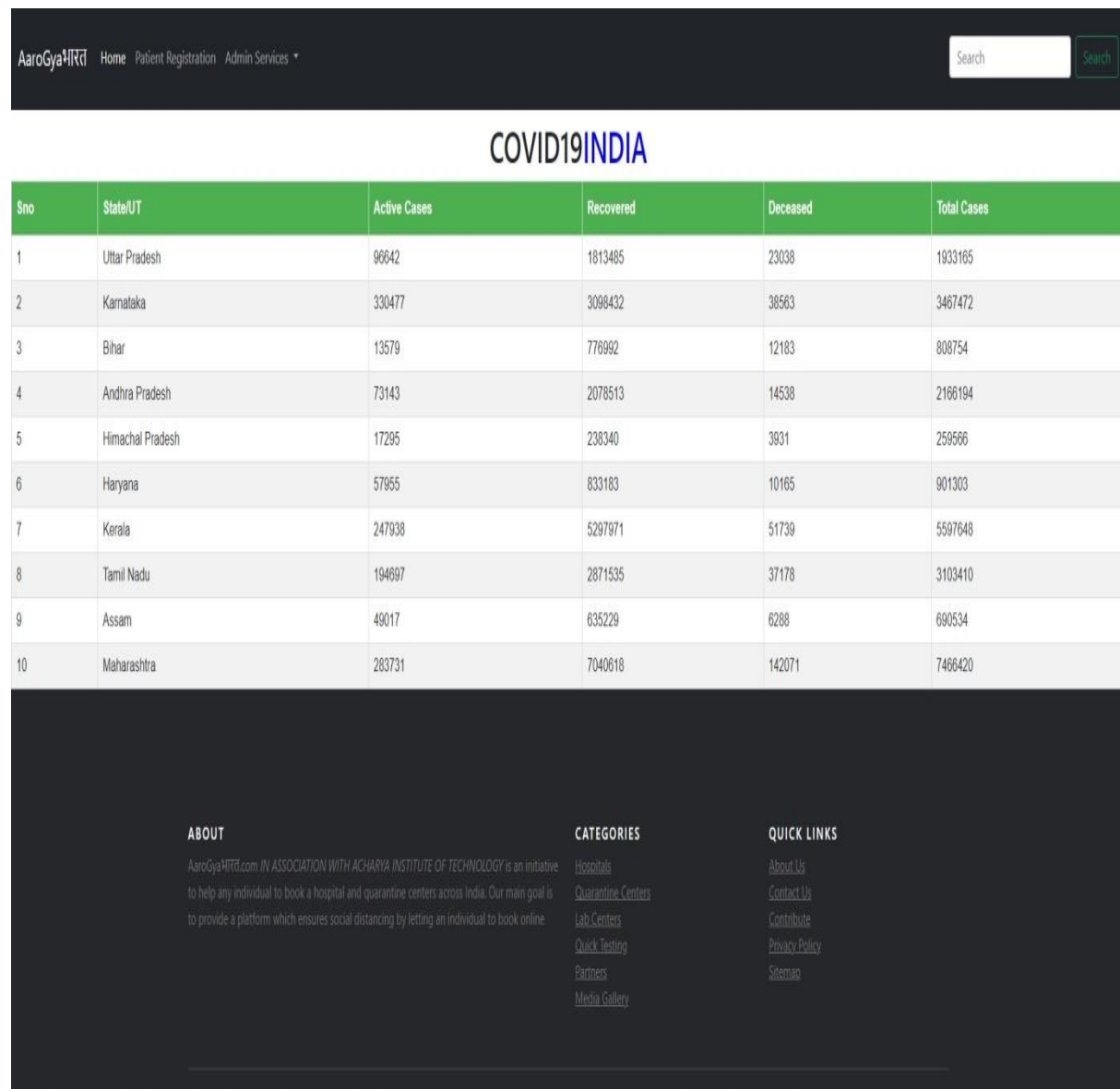
Sno	State/UT	Active Cases	Recovered	Deceased	Total Cases
1	Uttar Pradesh	96642	1813485	23038	1933165
2	Karnataka	330477	3098432	38563	3467472
3	Bihar	13579	776992	12183	808754
4	Andhra Pradesh	73143	2078513	14538	2166194
5	Himachal Pradesh	17295	238340	3931	259566
6	Haryana	57955	833183	10165	901303
7	Kerala	247938	5297971	51739	5597648
8	Tamil Nadu	194697	2871535	37178	3103410
9	Assam	49017	635229	6288	690534
10	Maharashtra	283731	7040618	142071	7466420

ABOUT

AaroGya भारत.com IN ASSOCIATION WITH ACHARYA INSTITUTE OF TECHNOLOGY is an initiative to help any individual to book a hospital and quarantine centers across India. Our main goal is to provide a platform which ensures social distancing by letting an individual to book online

CATEGORIES

Hospitals
Quarantine Centers
Lab Centers
Quick Testing
Partners
Media Gallery
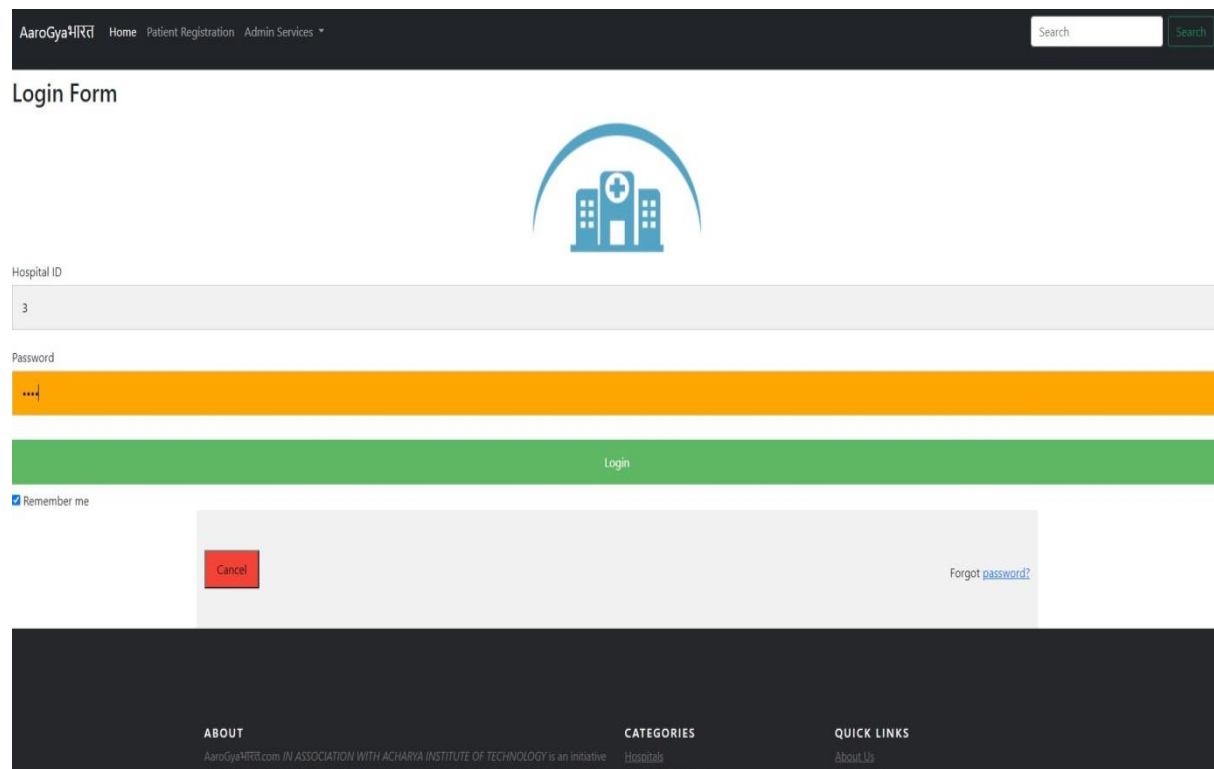
QUICK LINKS

About Us
Contact Us
Contribute
Privacy Policy
Sitemap

Fig. 5.1: Home Page

Fig. 5.2.1- Registration page

In this figure it's shown that the person is covid positive and condition is critical. Hence he will be directed to the Hospital booking page, which is shown in the next figure. (Fig. 5.2.1) The data from this form is stored in the person table in database.
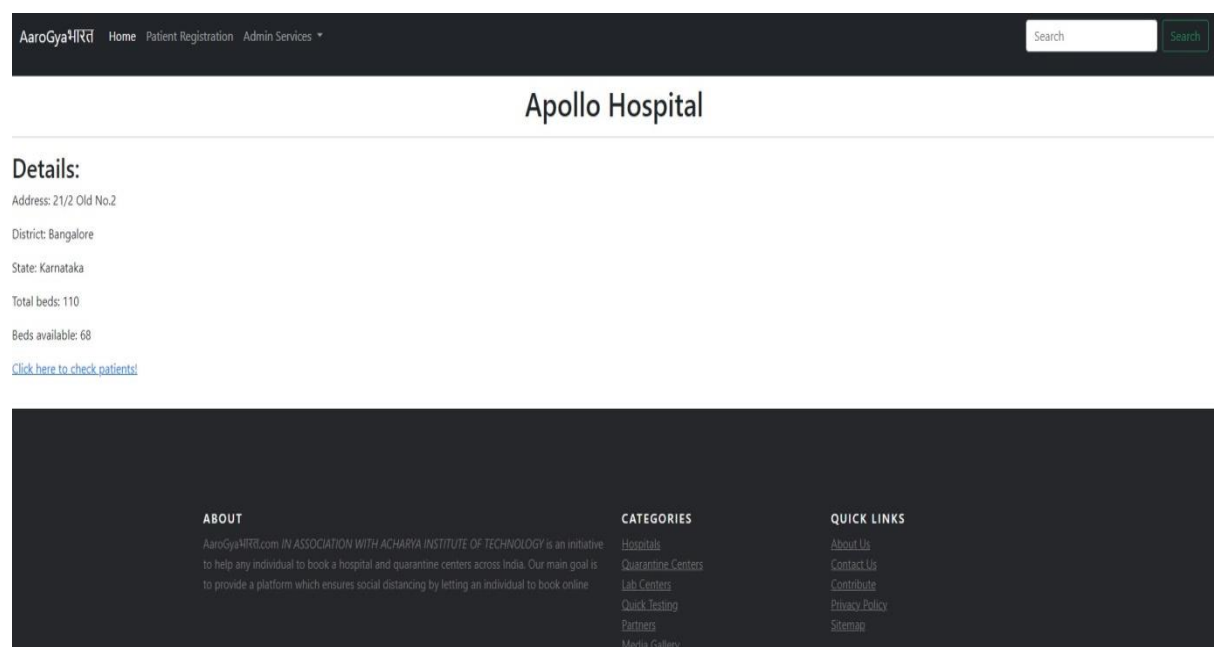
Fig. 5.2.2

In this figure, since the person mentioned his area as Bangalore hence the hospital in this area is shown and the booking can be done specifying his problem.

Also booking can be done using Aadhaar number which is the primary key in the patient table.

The data from here will be stored in the patient table.

When the user clicks on the book button a web page is displayed showing the confirmation of booking which is shown in the next figure (Fig-5.2.3)

Fig-5.2.3

Fig. 5.3- Registration form

In this figure it's shown that the person is covid positive and condition is not critical. Hence he will be directed to the Quarantine centre page, which is shown in the next figure (Fig.5.3.1)

The data from this form is stored in the person table in database.

Fig. - 5.3.1

In this figure, since the person mentioned his area as Bangalore hence the quarantine centre in this area is shown and the booking can be done by physically going there.

The data from here will be stored in the quarantine centres table.

Fig.- 5.4: Hospital Login Page

Fig.- 5.4.1: Hospital details

After logging in hospital, this page shows up showing logging successful.

Fig.-5.5 Admin Login Page

Fig.-5.5.1: Case Updating Page
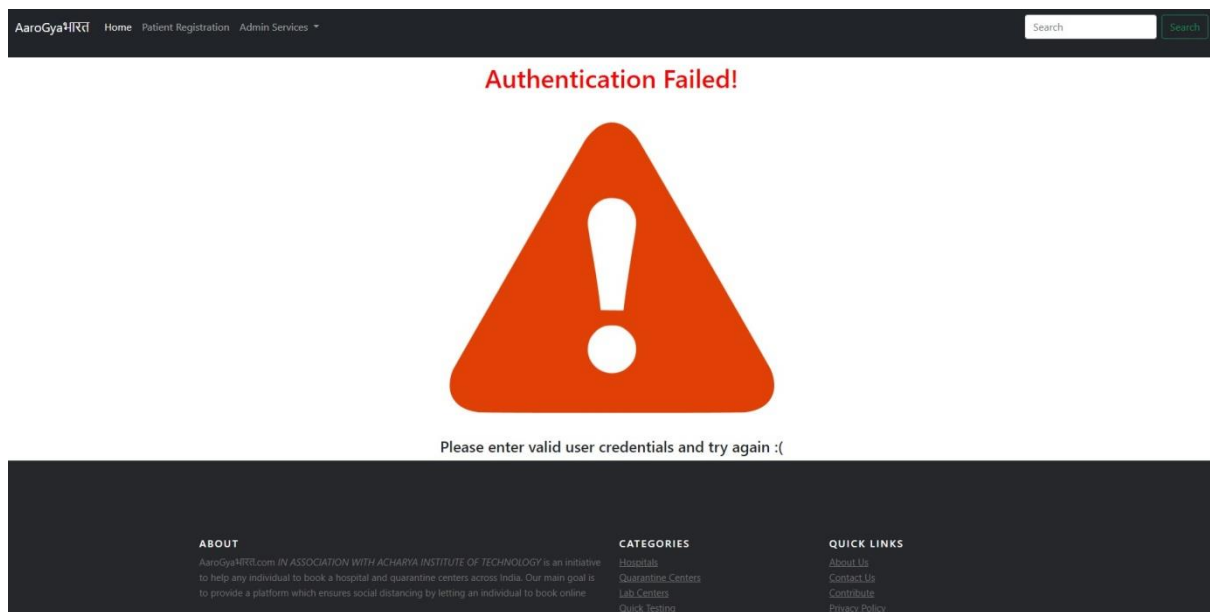
Fig.-5.6: Error page

If the entered credentials are wrong in the login page, then this page appears stating

"Authentication failed".

Chapter 6

CONCLUSION

The novel coronavirus spread so rapidly that it has changed the rhythm of the globe. Whether from the perspective of a single country or multilateral levels, the solidity of international relations has been put under test.It created opportunities to develop health services all across the globe and by health services I mean not only for Covid-19 but a health service all in all.After we overcome the pandemic, which will surely happen, we must carry out a comprehensive evaluation of the world's ability to maintain stability when faced with similar challenges in the future.We must also craft measures to cope with these challenges together. Such one small measure is taken up by our team. So, we have tried our best to design a web application such that one can register and get the best possible and immediate treatment of not only corona virus but all possible diseases (as registering for different kinds of diseases is possible in our portal).And at the same time, government and concerned authorities like hospitals will be able to track and monitor all the data flow.

Chapter 7

BIBLIOGRAPHY

[1] Login Form from https://www.w3schools.com/

[2] Header Footer for the website from https://getbootstrap.com/

[3] Creating databases from https://www.mysql.com/

[4] Rectifying errors and project enhancement from https://www.youtube.com/

[5] Learnt Flask for backend and SQLAlchemy to link database and python.

[6] RamezElmasri and Shamkant B. Navathe,"Database systems Models,Languages, Design and Application Programming", 6 th Edition, Pearson,2017.

[7] Ramakrishnan and Gehrke, "Database management systems",3rd Edition McGraw Hill 2014.

[8] Abraham Silberchatz, Henry korth and S.Sudarshan, "Database System Concepts", McGraw-Hill Education, 16 Jun 2010.

[9] https://flask.palletsprojects.com/en/2.0.x/

[10] https://www.sqlalchemy.org/

[11] Prof. Swathi U, from whom we learnt and could develop this project.