# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

**"Jnana Sangama", Belagavi-590018**

**A**

**DBMS Laboratory Mini Project Report**

**On**

# "COVID RECORD MANAGEMENT SYSTEM"

**SUBMITTED IN PARTIAL FULFILLMENT FOR 5TH SEMESTER**

**BACHELOR OF ENGINEERING**

**IN**

**INFORMATION SCIENCE AND ENGINEERING**

**SUBMITTED BY**


**ADITHI SATISH (1JB19IS003)**


**Under the Guidance of:**


**Mr. CHETAN R**
**Assistant Professor,**
**Dept. of ISE, SJBIT**



**DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING**

**SJB INSTITUTE OF TECHNOLOGY**

**BGS HEALTH AND EDUCATION CITY,**
**KENGERI, BENGALURU-560060, KARNATAKA, INDIA.**


**2021 - 2022**

|| Jai Sri Gurudev ||
Sri Adichunchanagiri Shikshana Trust ®

# SJB INSTITUTE OF TECHNOLOGY

BGS Health & Education City, Kengeri, Bengaluru – 560 060

## Department of Information Science & Engineering

# CERTIFICATE

Certified that the Mini project work entitled "**COVID RECORD MANAGEMENT SYSTEM**" carried out by Ms.**ADITHI SATISH** bearing USN **1JB19IS003** is a bonafide student of **SJB Institute of Technology** in partial fulfilment for 5th Semester DBMS Mini Project with Laboratory in **INFORMATION SCIENCE AND ENGINEERING** of the **Visvesvaraya Technological University**, **Belagavi** during the academic year **2021-22.** It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of mini project work prescribed for the said degree.

**Mr.CHETAN R**
**Assistant Professor**
**Dept. of ISE, SJBIT**

**Dr. MOHAN H.S**
**Professor & Head**
**Dept. of ISE, SJBIT**

1. **Internal Examiner** _____

2. **External Examiner** _____

# ACKNOWLEDGEMENT


I would like to express my profound grateful to His Divine Soul **Padmabhushan Sri Sri Sri Dr. Balagangadharanatha Maha Swamiji** and His Holiness **Jagadguru Sri Sri Sri Dr. Nirmalanandanatha Maha Swamiji** for providing me an opportunity to complete my academics in this esteemed institution.

I would also like to express my profound thanks to **Revered Sri Sri Dr. Prakashnath Swamiji**, Managing Director, SJB Institute of Technology, for his continuous support in providing amenities to carry out this project in this admired institution.

I express my gratitude to **Dr. K V Mahendra Prashanth**, **Principal**, SJB Institute of Technology, for providing me an excellent facilities and academic ambience; which has helped me in satisfactory completion of project work.

I extend my sincere thanks to **Dr. Mohan H S**, Professor & Head, Department of Information Science and Engineering; for providing me an invaluable support throughout the period of my project work.

I wish to express my heartfelt gratitude to my **guide, Mr. Chetan R**, **Assistant Professor,** Department of Information Science and Engineering for his valuable guidance, suggestions and cheerful encouragement during the entire period of my project work.

I express my truthful thanks to Mini Project Co-ordinator**, Mr. Chetan R**, **Assistant Professor,** Department of Information Science and Engineering for his valuable support.

Finally, I take this opportunity to extend my earnest gratitude and respect to my parents, Teaching &Non-teaching staffs of the department, the library staff and all my friends, who have directly or indirectly supported me during the period of my project work.

Regards,

ADITHI SATISH(1JB19IS003)

# ABSTRACT

Online Covid Record Management System is a system that gives you facility of booking any type of beds in an hospital for covid affected patients. This system is made so that patients will not have any difficulty in booking his/her slots. The project "Covid Record Management System' is developed to replace the currently existing system, which helps in keeping records of the patients, details of available bed as well as the bed slot allotted for patients.We are also providing the information about the hospital users who can access the hospital data and thereby edit,update or delete it.

In the present era where time proves to be the most important asset for an individual byreplacing the current register system to fully computerize, it not only saves the precious asset that is time, but also accuracy, reliability and uniformity can be maintained. This project is useful for the hospital as it helps them to search the data faster than existing system, to get patient record easily .

# TABLE OF CONTENTS

## LIST OF FIGURES PAGE NO

## LIST OF TABLES PAGE NO

# Chapter 1

# INTRODUCTION

## 1.1 OBJECTIVES

The proposed system is a web based application and maintains a centralized repository of all related information. The system allows one to easily access the relevant patient information and make necessary hospital bed bookings . Users can check the hospitals in which the beds are available and make bookings online for covid affected patients.

The objective of the Covid Record Management System project is to develop a system that automates the bed availability for the covid patients and the purpose is to design a system using which one can perform all operations related to bed slot booking.

This web application provides functionalities that allows a patient to book a bed using his/her SRFID. It maintains the details of all the patient users .

## 1.2 DBMS

A database management system (DBMS) is system software for creating and managing databases. The DBMS provides users and programmers with a systematic way to create, retrieve, update and manage data.
A DBMS makes it possible for end users to create, read, update and delete data in a database. The DBMS essentially serves as an interface between the database and end users or application programs, ensuring that data is consistently organized and remains easily accessible.

A DBMS provides concurrency, security, data integrity, consistency, controls redundancy and data independence. In this project the Relational DBMS (RDBMS)

used is MySQL. It is an open source software which uses SQL (Structured Query Language) which is a standard language for storing, manipulating and retrieving data in databases.

## 1.3 PYTHON

Python is one of the most versatile programming languages. It emphasizes code readability with extensive use of white space. It comes with the support of a vast collection of libraries which serve for various purposes, making our programming experience smoother and enjoyable.
Python programs are used for:

- Connecting with databases and performing backend development.
- Making web applications.
- Writing effective system scripts.
- And especially in data science and artificial intelligence.

In Python, web browser module provides a high-level interface which allows displaying.

# Chapter 2

# REQUIREMENTS SPECIFICATION

A Computerized way of handling information about property and user's details is efficient, organized and time saving, compared to a manual way of doing so this is done through a database driven web application whose requirements are mentioned in this section.

## 2.1 Specific Requirements

The specific requirements of Covid Record Management System are stated as follows:

### 2.1.1 Software Requirements

**Softwares used:**

- Operating System –Windows OS
- Front End – Visual Studio Code
- Back End – Xampp

**Technologies used:**

- Front End – HTML,CSS,JavaScript
- Contoller – Python django framework
- Back End – SQL,python flask

### 2.1.2 Software Requirements

**Hardware Components used:**

- CPU –Intel Core i3
- RAM – 6GB
- Peripherals – Standard PS/2 or USB Keyboard,Standard PS/2 or USB Wheel/Optical Mouse

## 2.2 Technology Used

- **Hypertext Mark-up Language (HTML)-** is the standard mark-up language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web. Web

browsers receive HTML documents from a web server or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

- **Cascading Style Sheets (CSS)-** It is a style sheet language used for describing the presentation of a document written in a mark-up language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media.

- **Django framework-** Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development. It is a Python-based free and open-source web framework that follows the model–template–views architectural pattern.

- **Python Flask -**Flask is a micro-framework developed in Python that provides only the essential components - things like routing, request handling, sessions, and so on. It provides you with libraries, tools, and modules to develop web applications like a blog, wiki, or even a commercial website. Flask is used for the backend, but it makes use of a templating language called Jinja2 which is used to create HTML, XML or other markup formats that are returned to the user via an HTTP request.

- **JavaScript -** JavaScript, often abbreviated as JS, is a high-level, dynamic, weakly typed, prototype-based, multi-paradigm, and interpretedprogramming language. Alongside HTML and CSS, JavaScript is one of the three core technologies of World Wide Webcontent production. It is used to make webpages interactive and provide online programs, including video games. The majority of websites employ it, and all modern web browsers support it without the need for plug-ins by means of a built-in JavaScript engine.

- **Structured Query Language (SQL)-**It is a domain-specific language used in programming and designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS). In comparison to older read/write APIs like ISAM or VSAM, SQL, offers two main advantages: first, it introduced the concept of accessing many records with one single command; and second, it eliminates the need to specify how to reach a record, eg. with or without an index.

- **XAMPP -** It is a free and open sourcecross-platformweb serversolution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDBdatabase, and interpreters for scripts written in the PHP and Perlprogramming languages. XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes. Everything needed to set up a web server - server application (Apache), database (MariaDB), and scripting language (PHP) is included in an - extractable file. XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows.

## 2.3 Functionality:

### Non-functional requirements:

Non-functional requirements place constraints on how the system will do so, this requirement **elaborates a performance characteristic** of the system. Some of them are:

**Performance**: The covid bed Management Database System must be able to perform 24X7 providing exceptional and valid results every time. The response must be very quick any time any type of user tries to access.

**Storage**: This system must be able to store all the information and the valid key in the database so that it is easy to use and more robust. It must be available for a long time.

**Usability**: The Database system will be used by various users and it must be capable of sharing and display the data in the most elegant way so that all the user can be able to use it and retrieve data from it.

**Security**: The system must be secure so that no unintended change in the database is done that reflects the loss of data or displaying wrong results. It must be safe from all hackers who try to get into the database.

**Availability**: The system must be ready to use at any time by any form of user. It should display the correct form of results.

## Functional requirements:

A **Functional Requirement** (FR) is a description of the service that this software offers. It describes a software system or its component. A function is nothing but inputs to the software system, its behavior, and outputs. It can be hospital information manipulation, patient user interaction, or any other specific functionality which defines what function a system is likely to perform like bed slot booking ,updating and deleting.

# Chapter 3

# SYSTEM DESIGN

The purpose people to gain from his/her project. As the developer works on the project, the test for of the design phase is to develop a clear understanding of what the developer wants every design decision should be efficient.

A purpose statement affects the design process by explaining what the developer wants the project to do, rather than describing the project itself. The Design Document will verify that the current design meets all of the explicit requirements contained in the system model as well as the implicit requirements desired by the customer.

## Structure of Design Document

System architecture section has:

**System Architecture Design -**The detailed diagram of the system server and client. **Data Design-**The data design includes an ER as well as Database design.

## Data Design:

## 3.1 Entity Relationship Diagram:

This relationship diagram shows how the tables in the database are connected to each other and how the control flows from one table to another when some action is triggered by the user. It also shows the constraints on the database such as primary key constraints, foreign key constraints and procedures and triggers. Entity Relationship Diagram is also called ER Diagram. When documenting a system or process, looking at the system in multiple ways increases the understanding of that system.

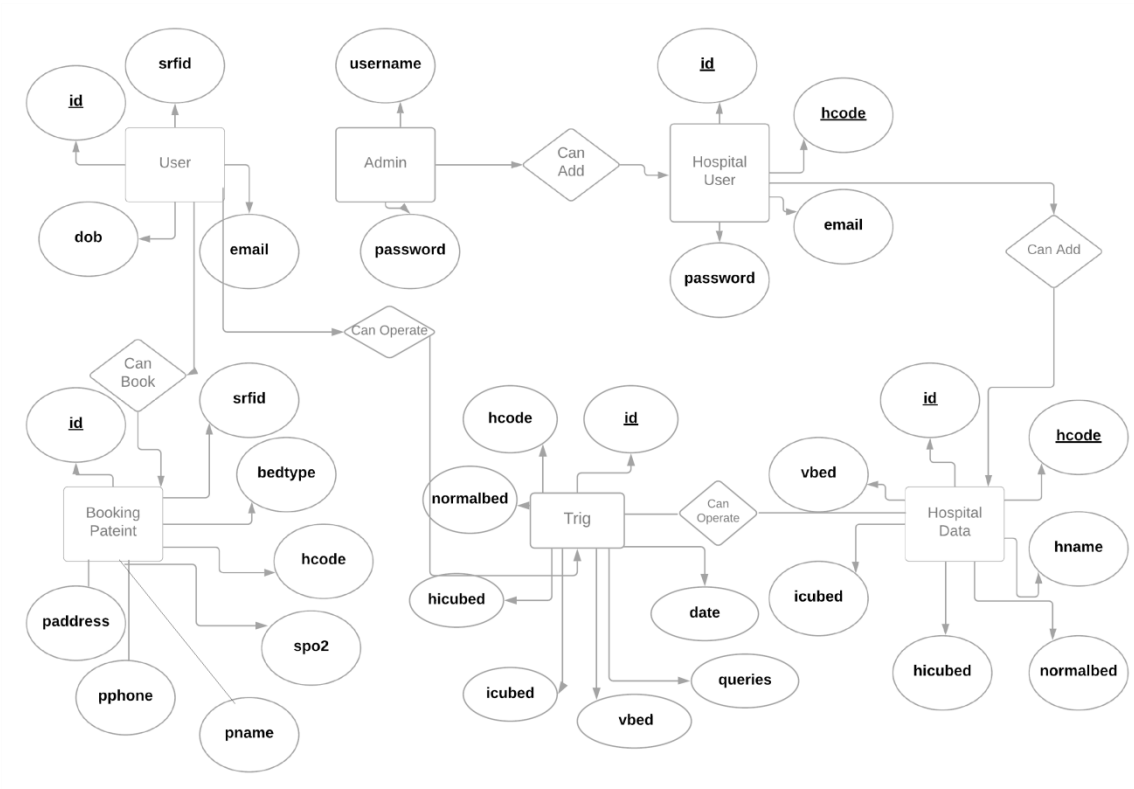



**Fig 3.1 ER diagram for Covid Record Management System**

ER diagrams are commonly used in conjunction with a data flow diagram to display the contents of a data store. They help us to visualize how data is connected in a general way, and are particularly useful for constructing a relational database. The database is normalized up to "third" normal form. That is the tables in the database will not have any multi valued fields(attributes) and there will be one primary key in each table that uniquely identifies the each tuple in the table.

## 3.2 Schema diagram:

The Schema Diagram gives us the information about the attributes in the table of the database and how the given tables are related to each other.

There are three types System Design.

## Architectural design

The architectural design of a system emphasizes the design of the system architecture that describes the structure, behavior and more views of that system and analysis.

## Logical design

The logical design of a system pertains to an abstract representation of the data flows, inputs and outputs of the system. This is often conducted via modelling, using an over abstract (and sometimes graphical) model of the actual system. In the context of systems, designs are included. Logical design includes entity-relationship diagrams (ER diagrams).

## Physical design

The physical design relates to the actual input and output processes of the system. This is explained in terms of how data is input into a system, how it is verified/authenticated, how it is processed, and how it is displayed. In physical design, the following requirements about the system are decided.

1. Input requirement,
2. Output requirements,
3. Storage requirements,
4. Processing requirements,
5. System control and backup or recovery.

Put another way, the physical portion of system design can generally be broken down into three sub-tasks:

1. User Interface Design
2. Data Design
3. Process Design

User Interface Design is concerned with how users add information to the system and with how the system presents information back to them. Data Design is concerned with how the data is represented and stored within the system. Finally, Process Design is concerned with

how data moves through the system, and with how and where it is validated, secured and/or transformed as it flows into, through and out of the system. At the end of the system design phase, documentation describing the three sub-tasks is produced and made available for use in the next phase.

Physical design, in this context, does not refer to the tangible physical design of an information system. To use an analogy, a personal computer's physical design involves input via a keyboard, processing within the CPU, and output via a monitor, printer, etc. It would not concern the actual layout of the tangible hardware, which for a PC would be a monitor, CPU, motherboard, hard drive, modems, video/graphics cards, USB slots, etc. It involves a detailed design of a user and a product database structure processor and control processor. The H/S personal specification is developed for the proposed system.

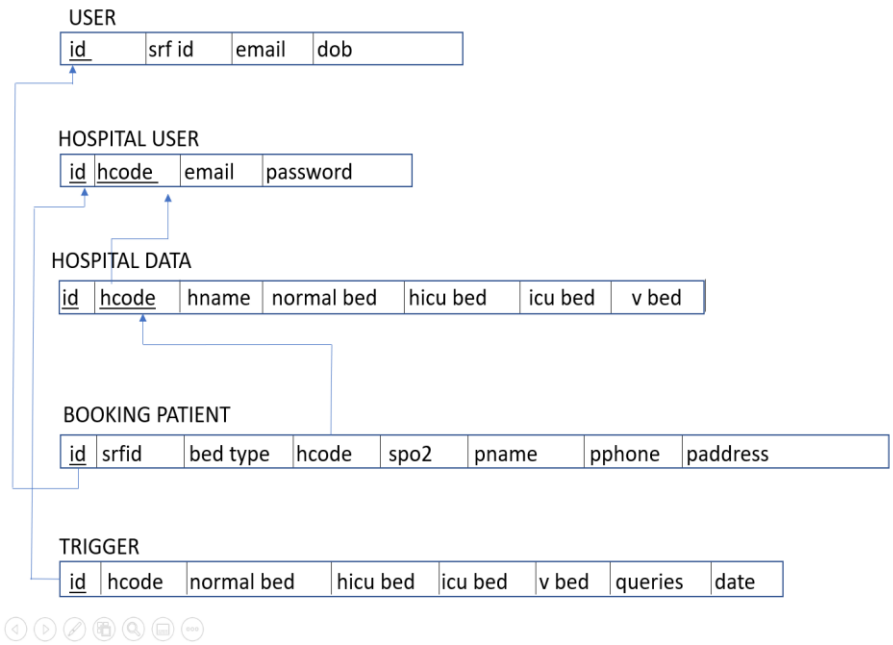



**Fig 3.2 Schema Diagram for Covid record Management System**

## Creating a database:

XAMPP stack of software is an open-source localhost server providing a number of functionalities through the package of software it contains. The software, which is part of XAMPP is started/stopped using the XAMPP_Control_Panel. It is used for testing the projects and modifications offline before launching it on the global web. One such very important functionality provided by XAMPP is the creation of the MySQL database. This is done by using phpMyAdmin**.**

phpMyAdmin is a costless and open source software that provides the functionality of operating and managing MySQL over the internet. It provides an ease to the user to control and supervise the database with the help of a graphic user interface known as phpMyAdmin.This GUI is written in PHP programming language. Over time it has gained a lot of trust and demand for the purpose of finding a web-based MySQL administration solution. The user can operate upon MySQL via phpMyAdmin user interface while still directly executing SQL queries. The GUI allows the host to carry a number of manipulation operations on the database, such as editing, creating, dropping, amending, alteration of fields, tables, indexes, etc. It can also be used to manage access control over the data by giving privileges and permissions. phpMyAdmin has thus a vital role to play in handling and creating a database

.

Now that we have run and tested phpMyAdmin, the next step is running MySQL and creating a database and table which will hold information to be used by our database. In order to start MySQL, navigate to the xampp directory and run the mysql_start.bat batch file. The XAMPP package contains an application called phpMyAdmin which allows developers to administer and maintain MySQL databases. We will be using phpMyAdmin to create a database and table, and enter test data. Before testing phpMyAdmin, make sure that both Apache and MySQL are running by opening their respective batch files: apache_start.bat and mysql_start.bat.

The below picture shows how exactly the xampppphpMyAdmin page looks like. All the SQL commands can be executed

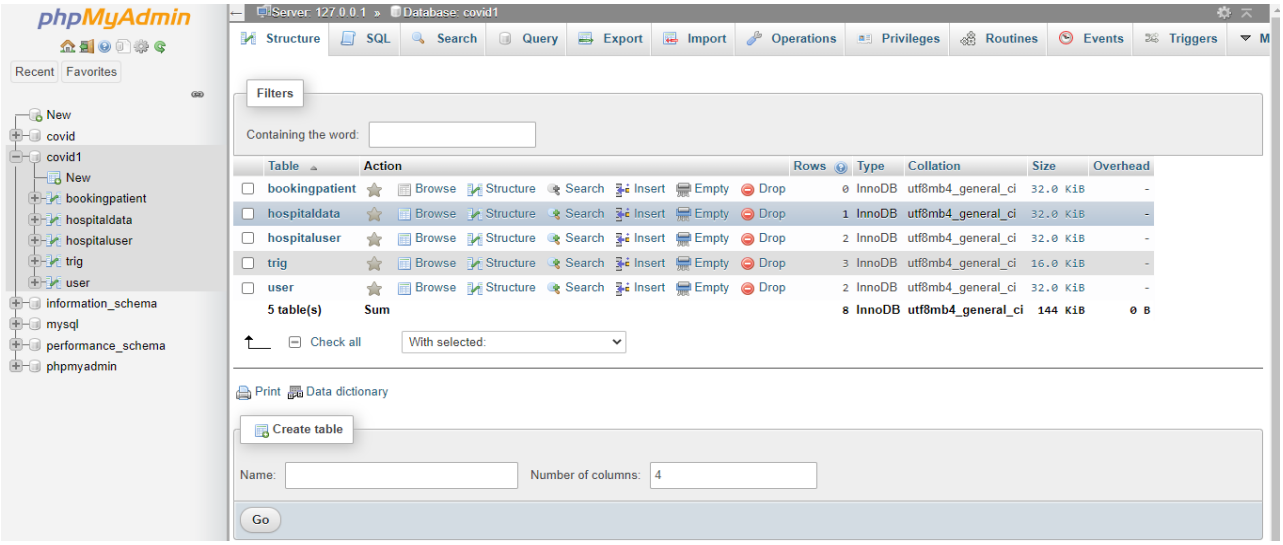



**Fig 3.3 Project Database Description**

# Chapter 4

# IMPLEMENTATION

## 4.1 Component Modules

### Module 1: Sign In

The Sign In has three options,which will redirect to the required pages.

They are:

1. User login page or home page
2. Hospital login page
3. Admin login page

### Module 2: Home page Window

### index.html

- Is the home page for users

### details.html

- Contains patient details

### hospitaldata.html

- Contains information about bed availability after retrieving data from hospital data

### trig.html

- Contains all the operations like updation ,deletion and insertion of hospital data

### booking.html

- Books the bed in the hospital in accordance with the availability of beds

### Sign In:

Contains all login pages of the hospital like:

- User Log In
- Admin Log In
- Hospital Log In

## Module 3: User home page Window

### usersignup.html

- Signup page gets the information of the new user
- It helps in accessing login page with the same data

### userlogin.html

- Logs into the page and the user can book the slot

### booking.html

- Here the user can view the availability of beds in the hospital and can book the slot

## Module 4: Admin home page Window

### admin.html

- Logs in the specific admin

### addhosuser.html

- Here the admin can add the hospital user
- Successful addition of the user says data sent and inserted
- Hospital Users id and password is sent to their Email address

## Module5:Hospital homepage Window

### hospitallogin.html

- Only the specific Hospital users are allowed to access this page

### Addhospitaldata.html

- Contains all the information regarding the availability of the bed in the hospital
- Hospital Users are allowed to update and delete the data

## 4.2 Connection to database:

The database is connected to the front end html using python flask,the code for database connection is shown below:

```
from flask_sqlalchemy import SQLAlchemy

#mydatabase connection
local_server=True
app=Flask(__name__)
app.secret_key="dbmsss"


app.config['SQLALCHEMY_TRACK_MODIFICATIONS']=False
#app.config['SQLALCHEMY_DATABASE_URI']='mysql://username:password@localhost/
databasename'
app.config['SQLALCHEMY_DATABASE_URI']='mysql://root:@localhost/covid1'
db=SQLAlchemy(app)
```

## 4.3 Table Creation

### Table 1:USER

The User table consists of the details of the user they are entering while signing up.

### User:

| **id** | **srfid** | **email** | **dob** |
|---|---|---|---|

- id refers to the unique id of the user.
- srfid refers to the srfid of the patient.
- email refers to the email id of the user.
- dob refers to the date of birth of the user(which is also the unique password for Log In).

The table **user** code is shown below:

```
class User(UserMixin,db.Model):
    id=db.Column(db.Integer,primary_key=True)
    srfid=db.Column(db.String(20),unique=True)
    email=db.Column(db.String(100))
    dob=db.Column(db.String(1000))
```

- In this table "id" is the primary key.

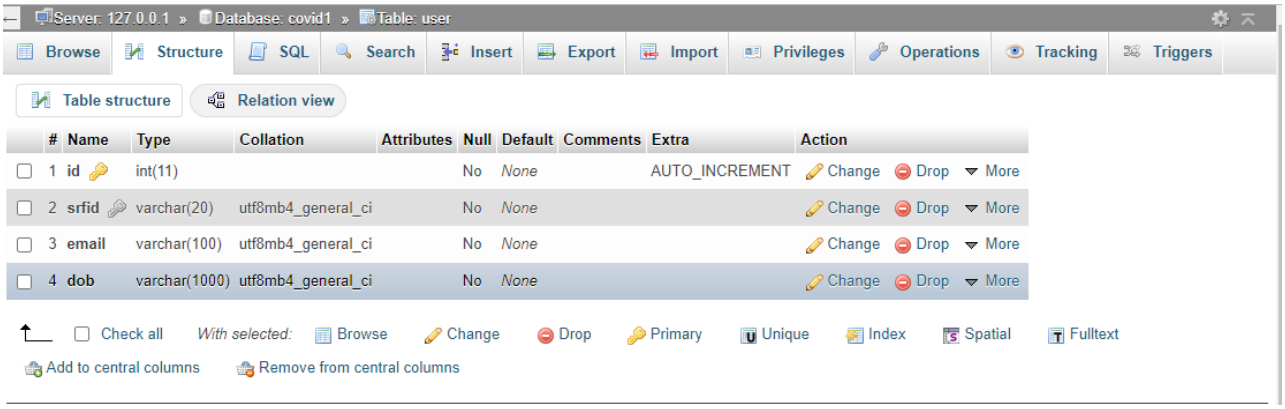



**Fig 4.1 User Table**

## Table 2:HOSPITALUSER

The HospitalUser table consists of the details of the hospital users added by the admin.

## Hospitaluser:

| **Id** | **hcode** | **email** | **password** |
|---|---|---|---|

- id refers to the unique id of the user.
- hcode refers to the hospital code of the particular hospital.

- email refers to the email id of the hospital user.
- password refers to the password of the user.

The table **hospitaluser** code is shown below:

```
class Hospitaluser(UserMixin,db.Model):
    id=db.Column(db.Integer,primary_key=True)
    hcode=db.Column(db.String(20),unique=True)
    email=db.Column(db.String(100))
    password=db.Column(db.String(1000))
```

- In this table "id" is the primary key and "hcode" is the unique key.

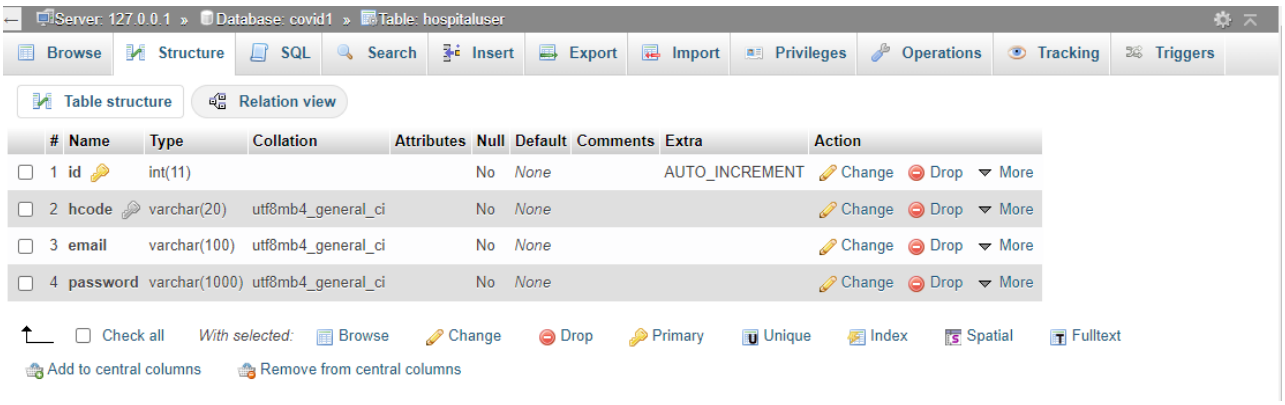



**Fig 4.2 HospitalUser Table**

## Table 3:HOSPITALDATA

The HospitalData table consists of all hospital related details like the number of ICU,HICU,VENTILATOR and NORMAL beds available.

## Hospitaldata:

| id | hcode | hname | normalbed | hicubed | icubed | vbed |
|---|---|---|---|---|---|---|

- id refers to the unique id of hospitalusers.
- hcode refers to the hospital code of the hospital.
- hname refers to the hospital name.
- normalbed refers to the number of normal beds available in the hospital.

- hicubed refers to the number of HICU beds available in the hospital.
- icubed refers to the number of ICU beds available in the hospital.
- vbed refers to the number of ventilator beds available in the hospital.

The table **hospitaldata** code is shown below:

```
class Hospitaldata(db.Model):
    id=db.Column(db.Integer,primary_key=True)
    hcode=db.Column(db.String(200),unique=True)
    hname=db.Column(db.String(200))
    normalbed=db.Column(db.Integer)
    hicubed=db.Column(db.Integer)
    icubed=db.Column(db.Integer)
    vbed=db.Column(db.Integer)
```

- In this table "id" is the primary key.

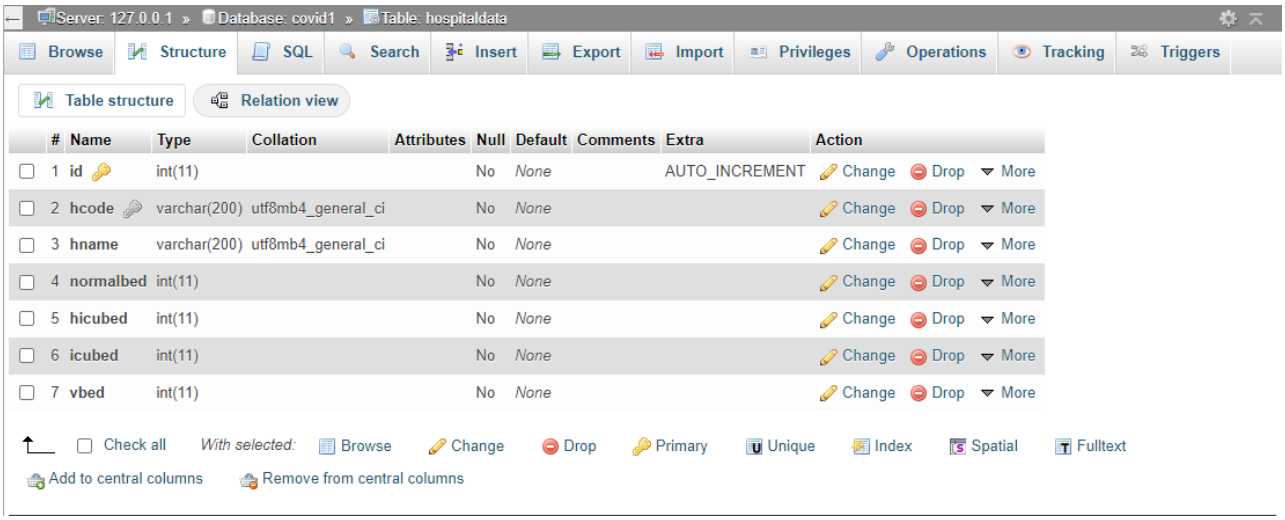



**Fig 4.3 HospitalData Table**

## Table 4:BOOKINGPATIENT

The BookingPatient table consists of all the details of the patients .

## Bookingpatiet:

| Id | srfid | bedtype | hcode | Spo2 | pname | pphone | paddress |
|---|---|---|---|---|---|---|---|

- id refers to the unique id of the patient user.
- srfid refers to the srfid of the patient.
- bedtype refers to the type of hospital bed required.
- hcode refers to the hospital code of the hospital.
- spo2 refers to the oxygen level of the patient.
- pname refers to the patient name.
- pphone refers to the patient phone number.
- Paddress refers to the patient address.

The table **BookingPatient** code is shown below:

```
class Bookingpatient(db.Model):
    id=db.Column(db.Integer,primary_key=True)
    srfid=db.Column(db.String(20),unique=True)
    bedtype=db.Column(db.String(100))
    hcode=db.Column(db.String(20))
    spo2=db.Column(db.Integer)
    pname=db.Column(db.String(100))
    pphone=db.Column(db.String(100))
    paddress=db.Column(db.String(100))
```

- In this table “id” is the primary key.

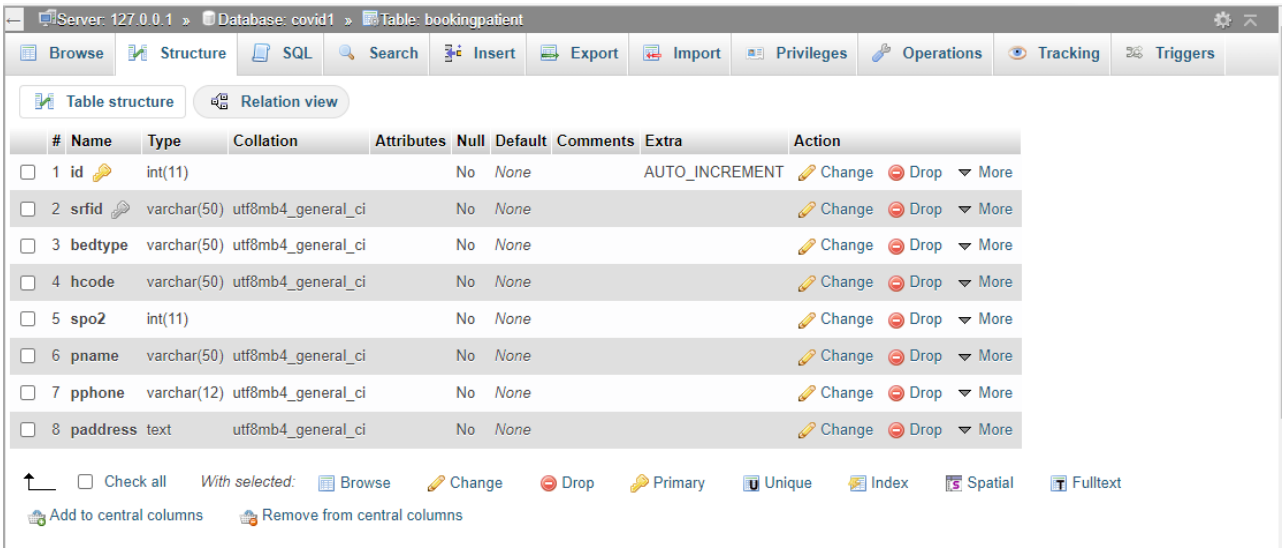



**Fig 4.4 BookingPatient Table**

## Table 5:TRIG

The Trig table consist of all the details of the bed availability after updation,insertion and deletion.

## Trig:

| **Id** | **hcode** | **normalbed** | **hicubed** | **icubed** | **vbed** | **queries** | **date** |
|---|---|---|---|---|---|---|---|

- id refers to the unique id of the hospital users.
- hcode refers to the hospital code of the hospital.
- normalbed refers to the number of normal beds available in the hospital.
- hicubed refers to the number of HICU beds available in the hospital.
- icubed refers to the number of ICU beds available in the hospital.
- vbed refers to the number of ventilator beds available in the hospital.
- queries refers to the operation like insertion,updation and deletion.
- date refers to the date when the operation took place.

The table **trig** code is shown below:

```
class Trig(db.Model):
    id=db.Column(db.Integer,primary_key=True)
    hcode=db.Column(db.String(20))
    normalbed=db.Column(db.Integer)
    hicubed=db.Column(db.Integer)
    icubed=db.Column(db.Integer)
    vbed=db.Column(db.Integer)
    queries=db.Column(db.String(50))
    date=db.Column(db.String(50))
```

- In this table “id” is the primary key.

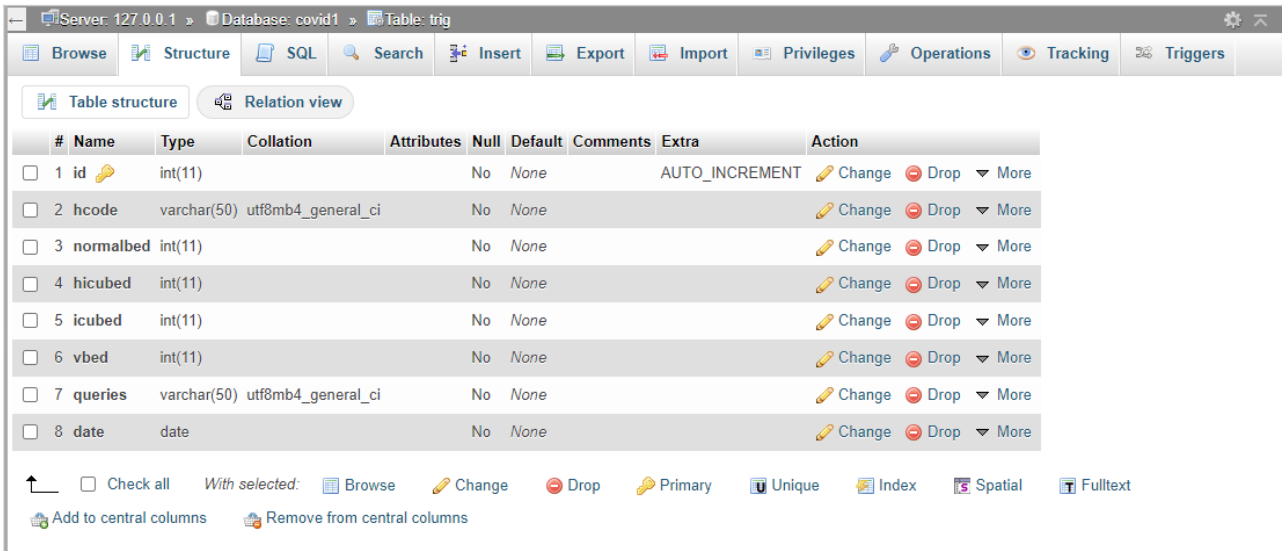



**Fig 4.5 Trig Table**

# TRIGGERS:

**Triggers** are stored programs, which are automatically executed or fired when some event occurs. **Triggers** are written to be executed in response to any of the following events. A database manipulation (DML) statement (DELETE, INSERT, or UPDATE). A database definition (DDL) statement (CREATE, ALTER, or DROP).

The trigger named **TRIG** used in this project enters the date at which a user registered the account. Using this trigger the user who is entering his details does not have to enter the date of registeration, the date will be generated automatically to that date.

- create trigger [trigger_name]: Creates or replaces an existing trigger with the trigger_name.
- [before | after]: This specifies when the trigger will be executed.
- {insert | update | delete}: This specifies the DML operation.
- on [table_name]: This specifies the name of the table associated with the trigger.
- [for each row]: This specifies a row-level trigger, i.e., the trigger will be executed for each row being affected.
- [trigger_body]: This provides the operation to be performed as trigger is fired

The code of trigger is give below:

## Code:

## FOR INSERTION:

```
INSERT INTO trig
VALUES(null,NEW.hcode,NEW.normalbed,NEW.hicubed,NEW.icubed,NEW.vbed,
'INSERTED',NOW())
```

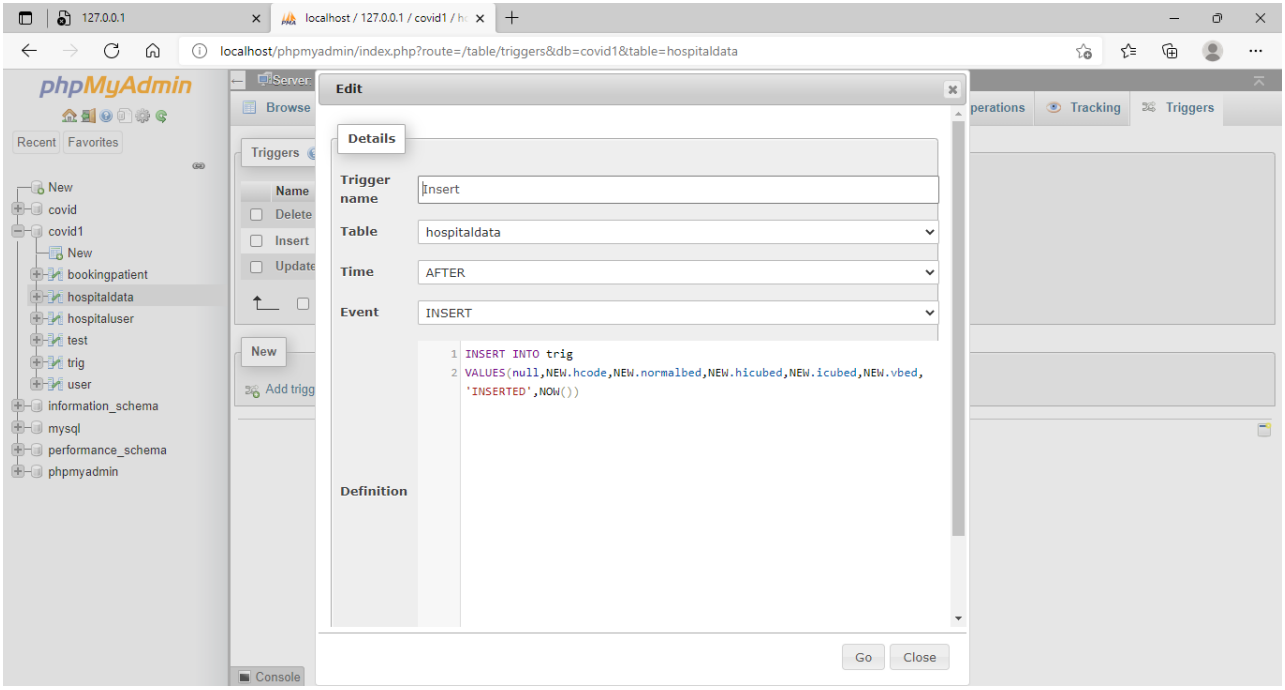



**Fig 4.6 Insertion trigger**

# FOR UPDATION:

```
INSERT INTO trig
VALUES(null,NEW.hcode,NEW.normalbed,NEW.hicubed,NEW.icubed,NEW.vbed,
'UPDATED',NOW())
```

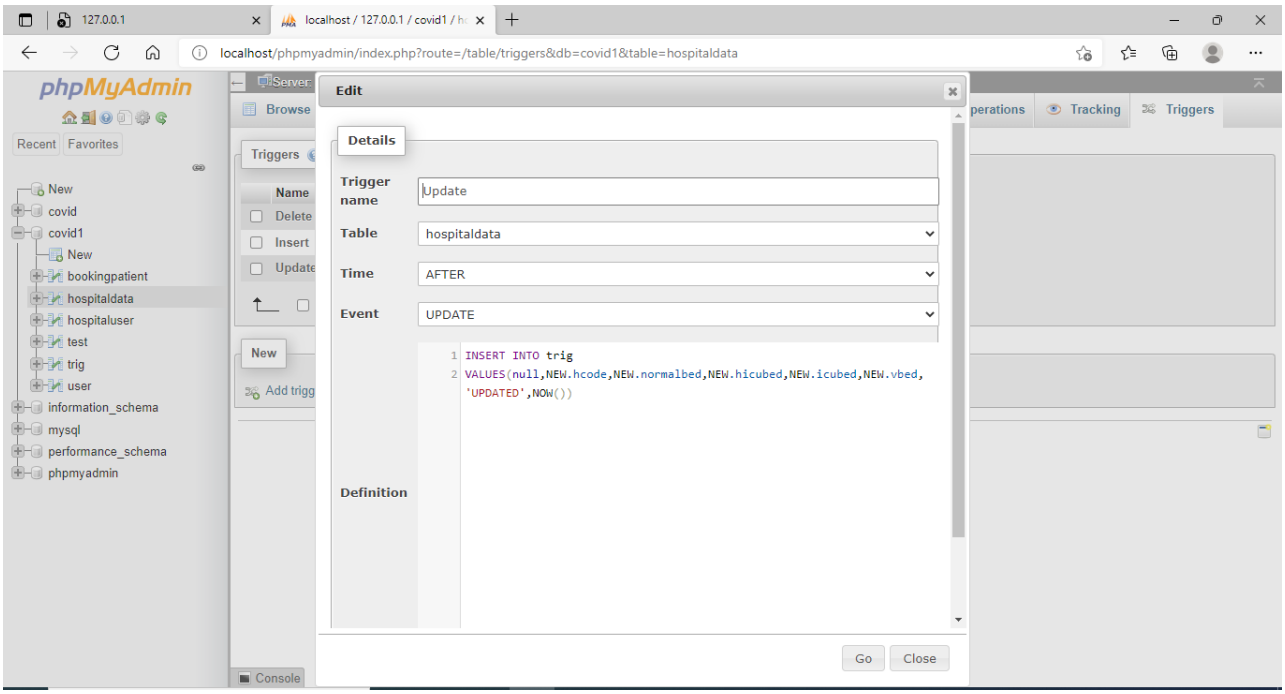



**Fig 4.7 Updation trigger**

# FOR DELETION:

```
INSERT INTO trig
VALUES(null,OLD.hcode,OLD.normalbed,OLD.hicubed,OLD.icubed,OLD.vbed,
'DELETED',NOW())
```

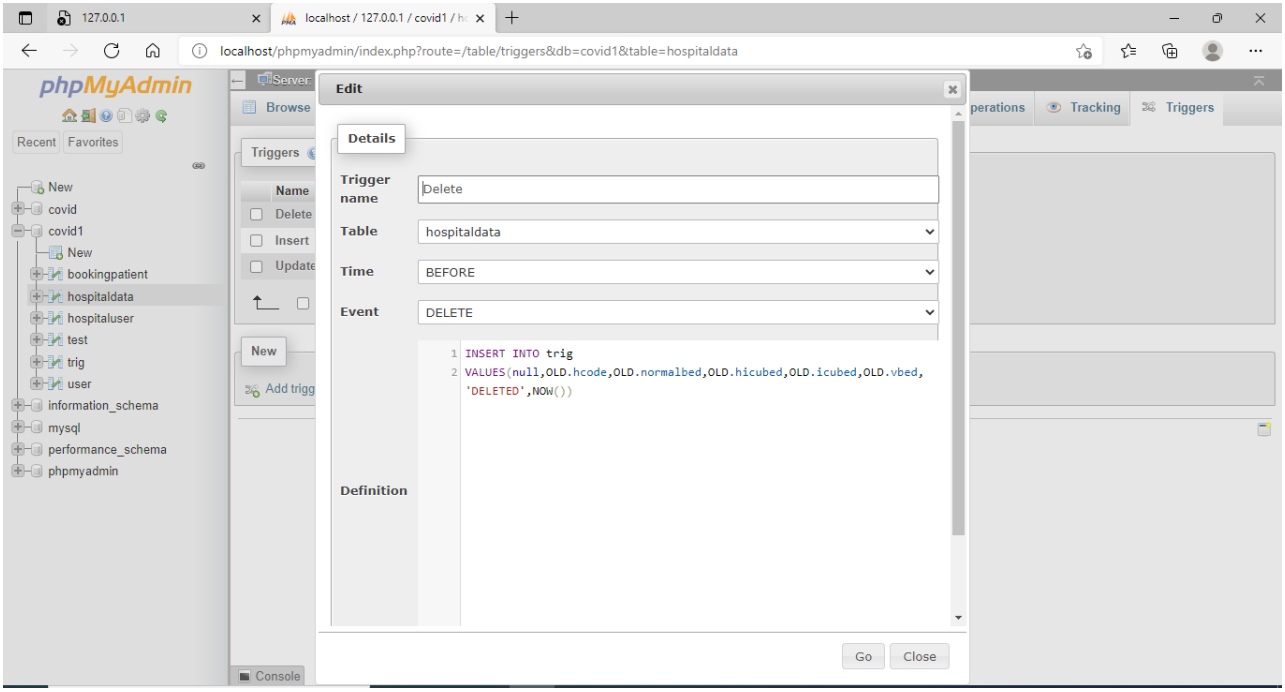



**Fig 4.8 Deletion trigger**

# Chapter 5

# SYSTEM TESTING

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic.

Two Types of Testing are:

1.Unit testing
2.Integration testing

| **Functionality** | **Action** | **Expected Result** | **Actual Result** | **Test Result** |
|---|---|---|---|---|
| Creating an account for patient user and storing details to user table | Signup is clicked | Should register a patient user successfully and enter user login page | Data stored successfully in the table and user can login now | PASS |
| Accepting user input for user login | Login is clicked | Should enter User Home Page | Entered into User Home Page | PASS |
| Checking the Admin Login Credentials | Admin Login is clicked | Should enter Admin Home Page | Entered into Admin Home Page | PASS |
| Admin adding the hospital users and storing details in hospitaluser table | Add is clicked in Admin Page | Should register a hospital user successfully and send an email to their email address | Data stored successfully in the table and email sent | PASS |

| | | | | |
|---|---|---|---|---|
| | | | | |
| Admin Logout | Logout is clicked | Should successfully logout and display Admin Login page | Successfully redirected to Admin login page | PASS |
| Accepting hospital user input for hospital login | Login is clicked | Should enter Hospital Home Page | Entered into Hospital Login Home page | PASS |
| Reconfirming the hospital user login credentials | Add Hospital is clicked | Should ask for reconfirmation and enter the add hospital information page | Successfully reconfirmed and entered into add hospital information page | PASS |
| Adding Hospital information and storing the details in Hospitaldata table | Add is Clicked | Should add the information successfully and display add hospital information page | Data stored successfully in the hospitaldata table | PASS |
| Editing Hospital Information | Edit symbol is clicked | Should enter Hospital edit page | Successfully entered into hospital edit page | PASS |
| Updating Hospital Information and storing the updated value in hospitaldata table | Update is clicked | Should update the information successfully and display add hospital information page | Data updated successfully in the hospitaldata table | PASS |

| | | | | |
|---|---|---|---|---|
| Deleting Hospital Information | Delete symbol is clicked | Should delete the information successfully | Data successfully deleted | PASS |
| Hospital user Logout | Logout is clicked | Should successfully logout | Successfully logged out | PASS |
| Booking bed slot for patient | Book Slot is clicked | Should redirect to Slot booking page | Successfully redirected to slot booking page | PASS |
| Adding patient details and storing it in bookingpatient table | Book Slot is clicked | Should book the slot successfully and display slot booking page | Data successfully stored in bookingpatient table | PASS |
| Viewing patient details | Patient details is clicked | Should display the patient details | Patient details displayed | PASS |

| | | | | |
|---|---|---|---|---|
| User Logout | Logout is clicked | Should successfully logout and redirect to user login page | Successfully redirected to user login page | PASS |

# Chapter 6

# SNAPSHOTS

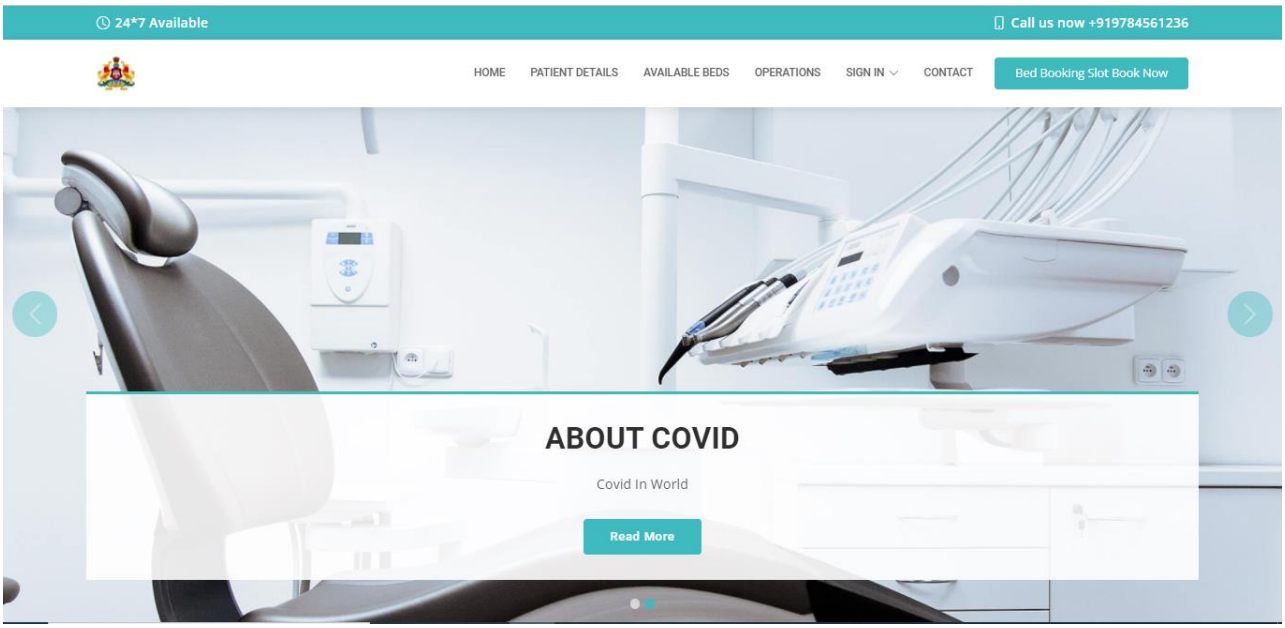



**Fig 6.1 Home Page**

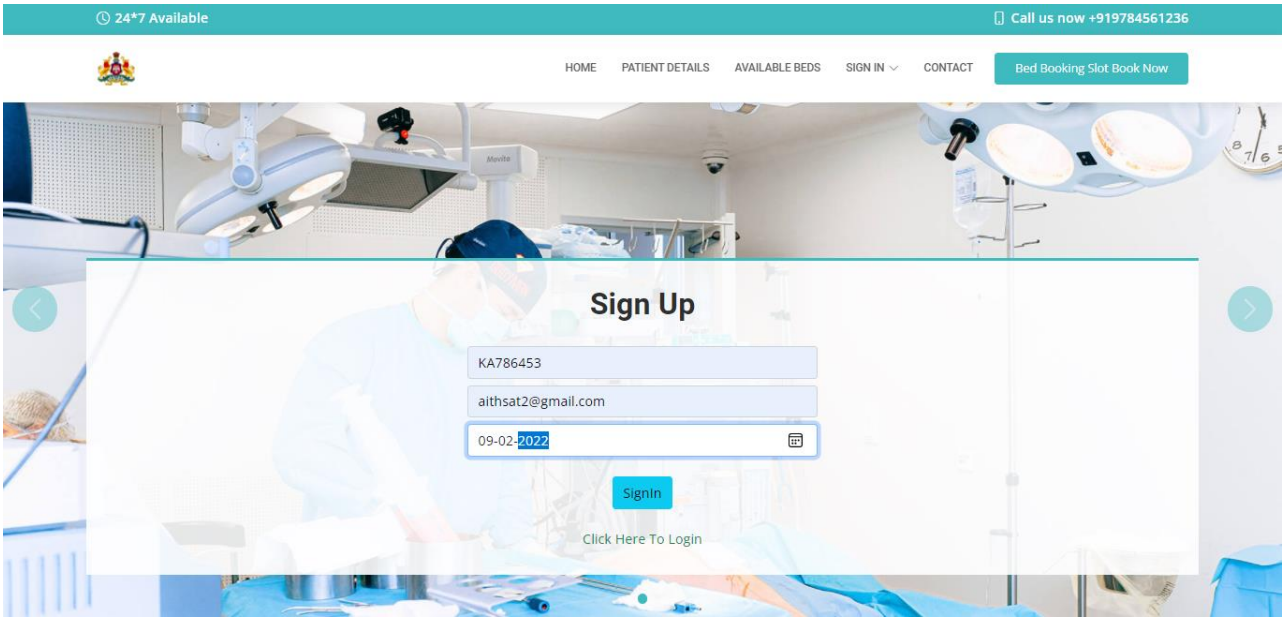



**Fig 6.2 User Sign Up page**

Signup page creates an account for patient user.

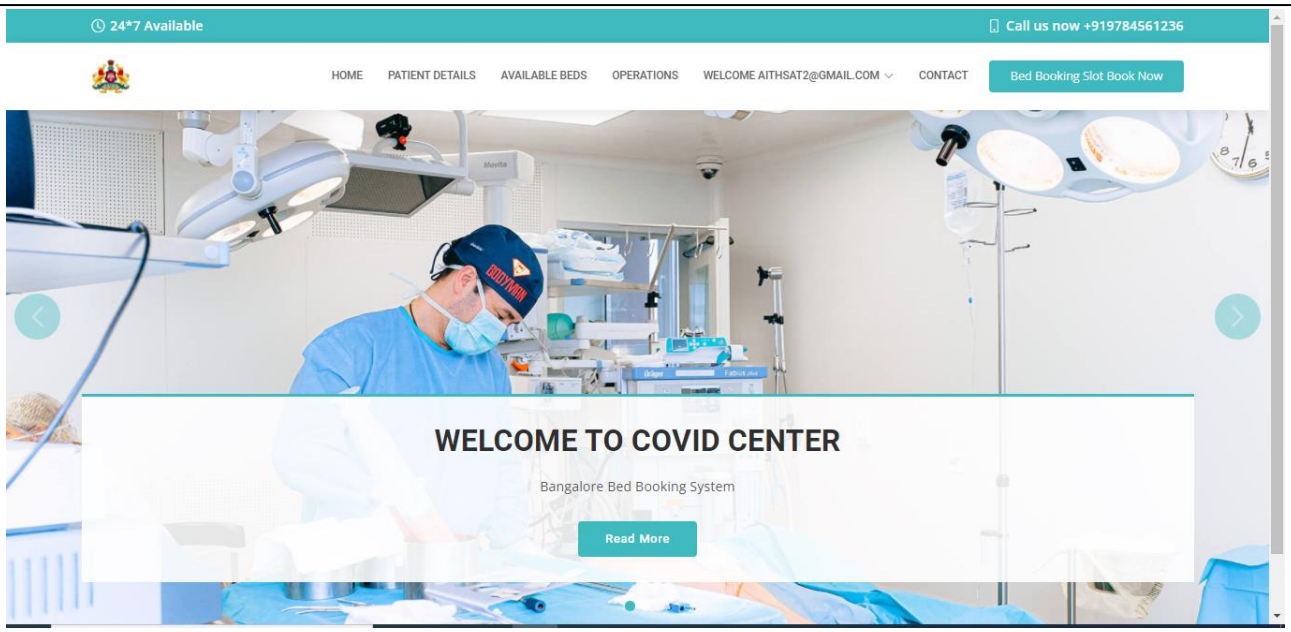



**Fig 6.3 Patient User Login Page**

This page is used by the patient user for booking the bed slots in the hospital.

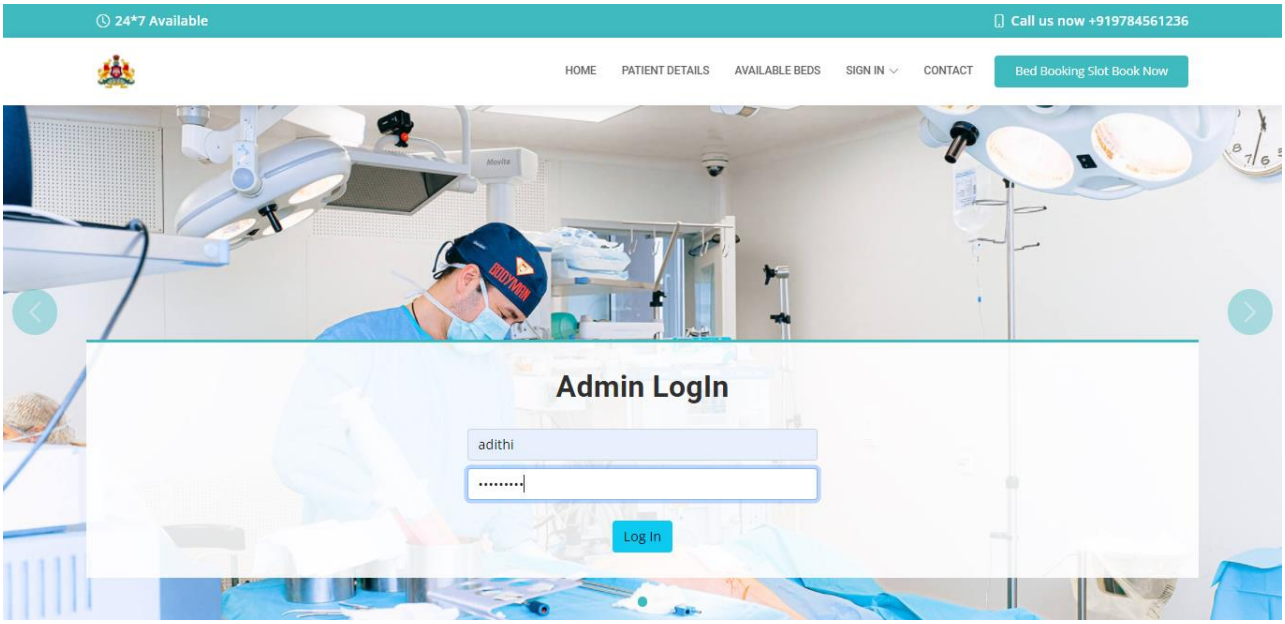



**Fig 6.4 Admin login Page**

This page is used by the admin to add the hospital users.

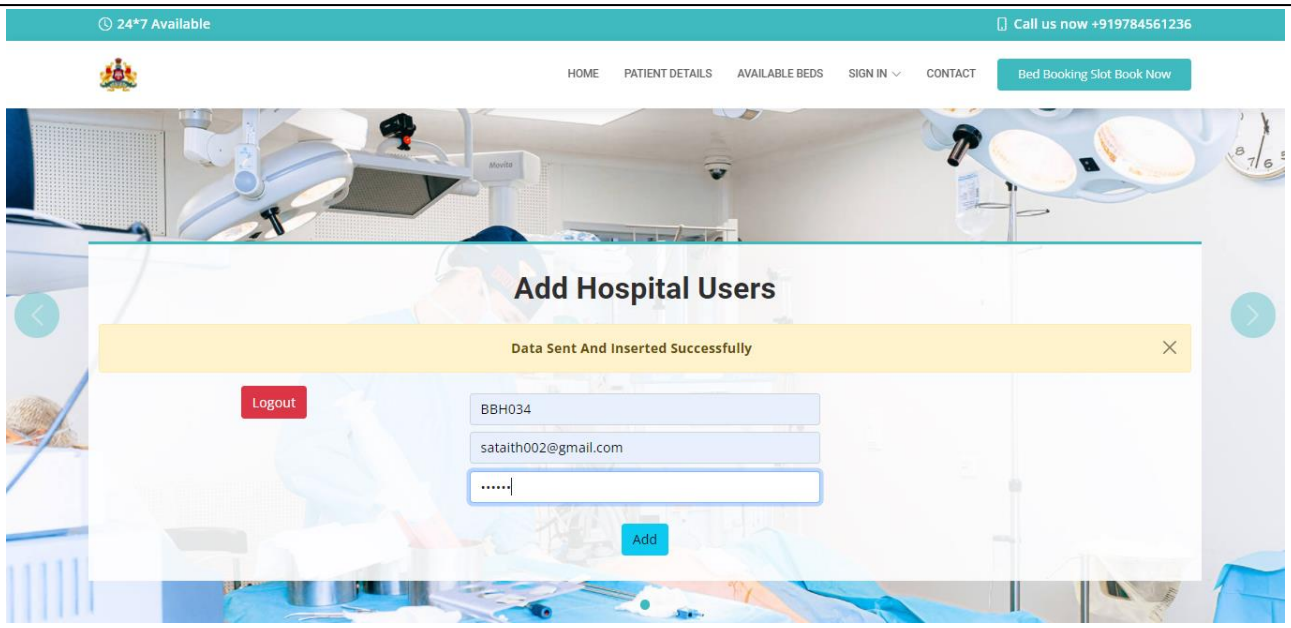



**Fig 6.5 Add hospital user login page**

This page is used to add hospital users by the admin.

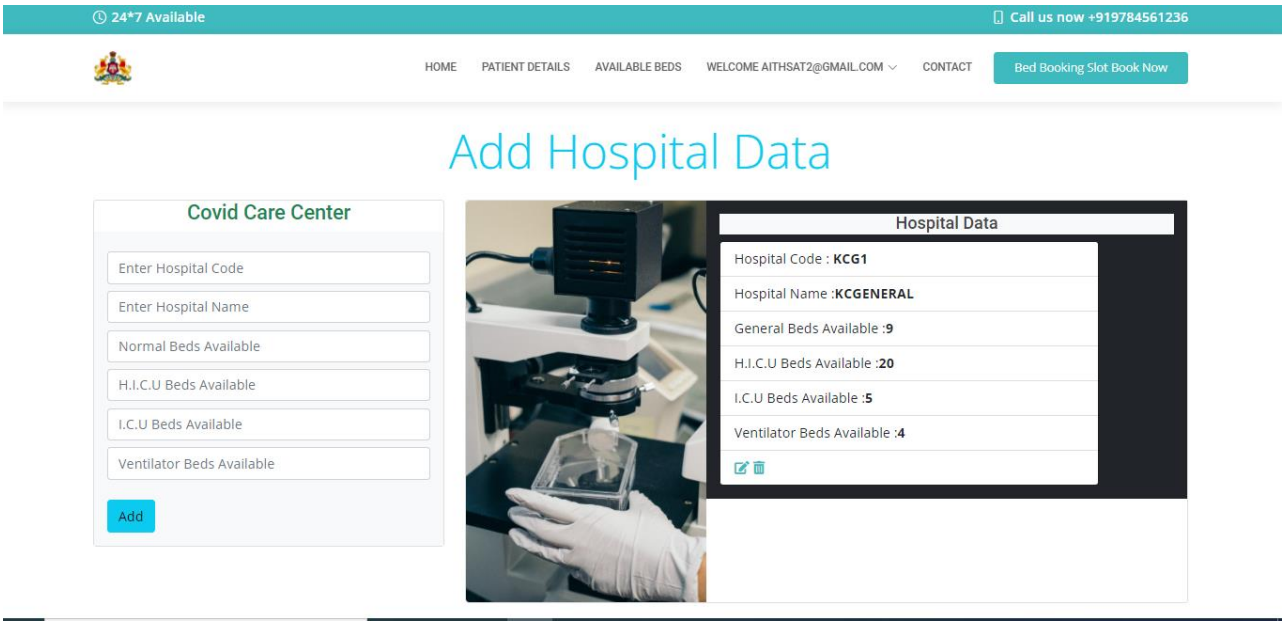



**Fig 6.6 Add hospital information page**

This page is used to add hospital information by the hospital users.

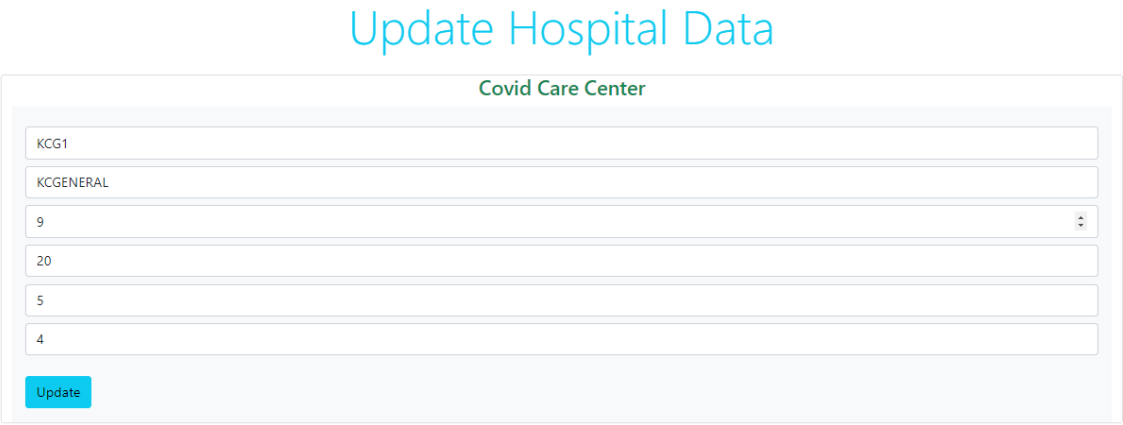



**Fig 6.7 Edit page**

This page is used to edit the hospital information.

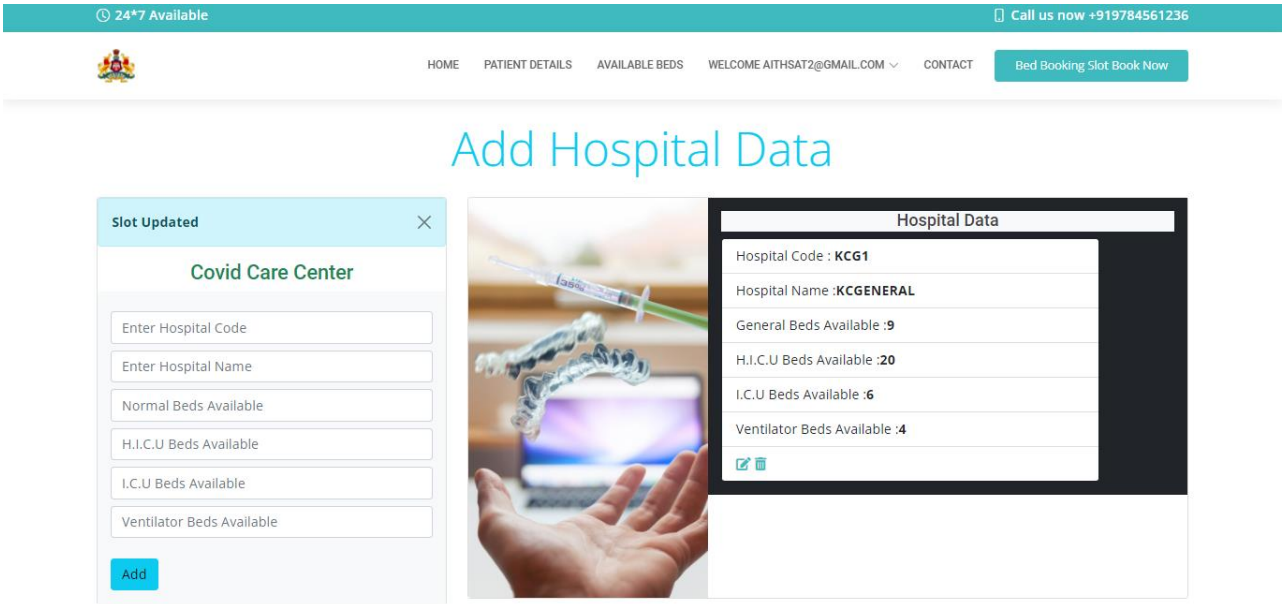



**Fig 6.8 Update Page**

This page is gives us the updated hospital information.

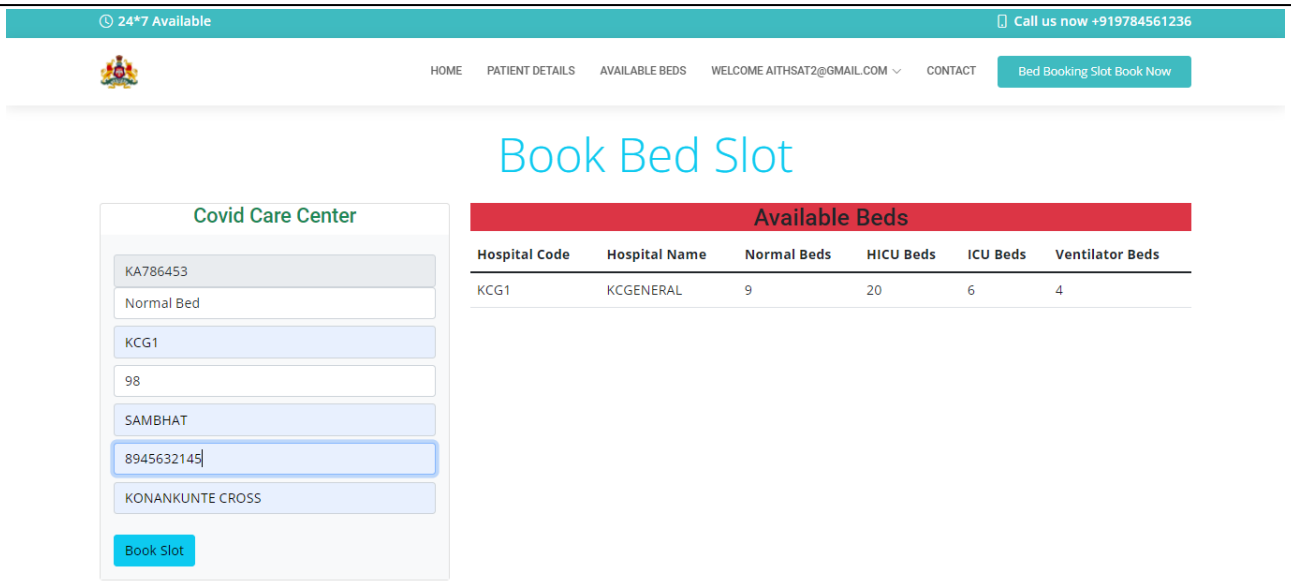



**Fig 6.9 Bed Booking Page**

This page is used to book the bed slot in the hospital by the patient users.

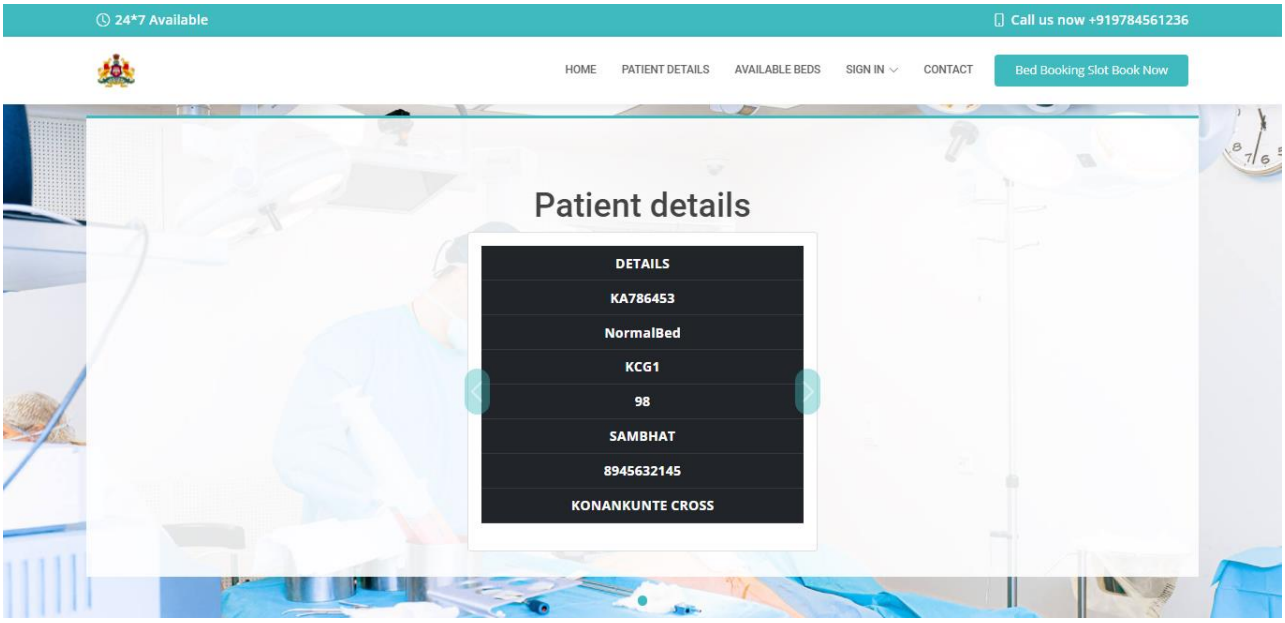
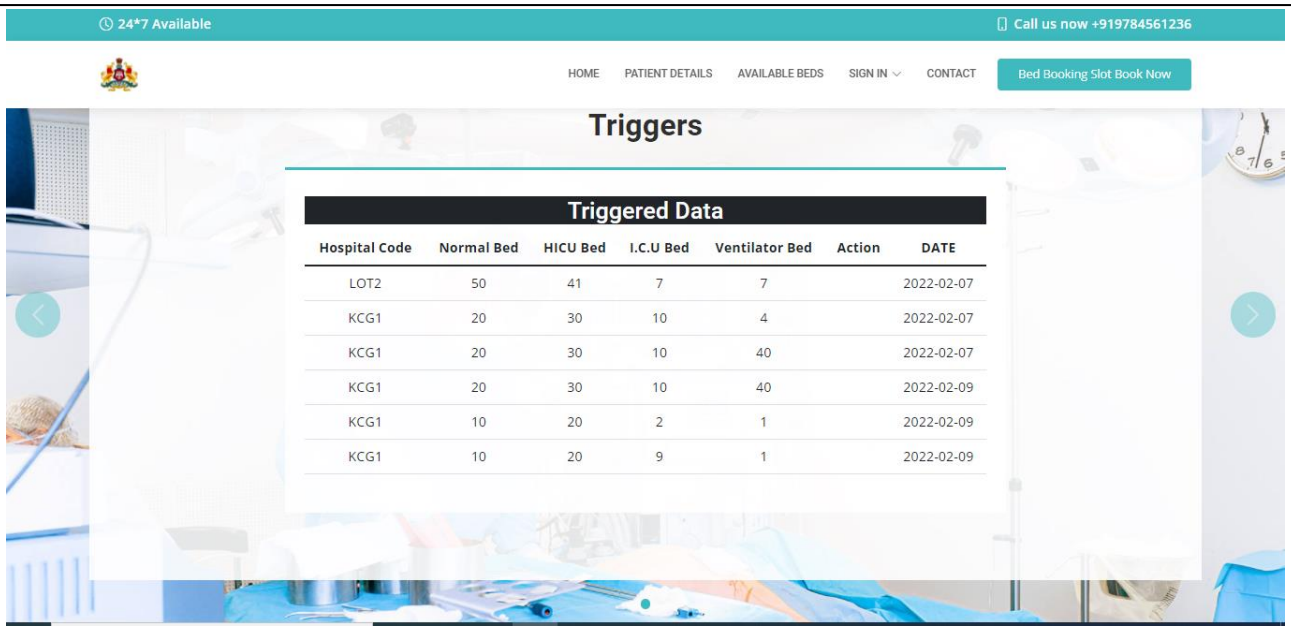



**Fig 6.10 Patient Details page**

This page gives us the patient details as entered in the hospital.




| Hospital Code | Normal Bed | HICU Bed | I.C.U Bed | Ventilator Bed | Action | DATE |
|---|---|---|---|---|---|---|
| LOT2 | 50 | 41 | 7 | 7 | | 2022-02-07 |
| KCG1 | 20 | 30 | 10 | 4 | | 2022-02-07 |
| KCG1 | 20 | 30 | 10 | 40 | | 2022-02-07 |
| KCG1 | 20 | 30 | 10 | 40 | | 2022-02-09 |
| KCG1 | 10 | 20 | 2 | 1 | | 2022-02-09 |
| KCG1 | 10 | 20 | 9 | 1 | | 2022-02-09 |

**Fig 6.11 Trigger Page**

This page gives all the operations like updation,insertion and deletion.

# CONCLUSION

Covid Record Management System project is an attempt to develop a system that automates the processes and activities of hospital bed booking and the purpose is to design a system using which one can perform all operations related to bed booking.

In the present era where time proves to be the most important asset for an individual by replacing the current register system to fully computerize, it not only saves the precious asset that is time, but also accuracy, reliability and uniformity can be maintained. This project is useful for the hospital as it helps them to search the data faster than existing system, to get patient record easily and are generated as per requirement. We believe that we have accomplished our goals and satisfied with the code we developed.

# BIBLIOGRAPHY

1. "Database Systems Models,Languages, Design and Application Programming 7th Edition, 2017,Pearson"- RamezElmasri,BNavathe

2."Database Management Systems 3rd Edition,2014,McGraw Hill" –Ramakrishnan and Gehrke

**Websites Referred**

3. Tutorialspoint
Learned about the basic concept of trigger and how and where can we use a trigger.
https://www.tutorialspoint.com/plsql/plsql triggers.htm

4.W3Schools
Learned how to create a responsive slideshow with CSS and JavaScript.
https://www.w3schools.com/howto/howto is slideshow.asp

5.Stack Overflow
Helped in finding and resolving the error.
http://stackoverflow.com