

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANASANGAMA, BELAGAVI - 590018



DBMS Laboratory with Mini Project Report
on

COVID BED ALLOCATION MANAGEMENT SYSTEM

Submitted in partial fulfillment for the award of degree of

Bachelor of Engineering
in
COMPUTER SCIENCE AND ENGINEERING

Submitted by

1BG20CS006 AFSHAN TABASSUM

1BG20CS017 ANUSHA A
Internal Guide

Mrs Jayashree

Assistance Professor, Dept. of CSE
BNMIT, Bengaluru



Vidyayāmruthamashnute

B.N.M. Institute of Technology

An Autonomous Institution under VTU

Approved by AICTE, Accredited as grade A Institution by NAAC. All eligible branches – CSE, ECE, EEE, ISE & Mech. Engg. are Accredited by NBA for academic years 2018-19 to 2024-25 & valid upto 30.06.2025

URL: www.bnmit.org

Department of Computer Science and Engineering
2022 - 2023

B.N.M. Institute of Technology

An Autonomous Institution under VTU

Approved by AICTE, Accredited as grade A Institution by NAAC. All eligible branches – CSE, ECE, EEE, ISE & Mech. Engg. are Accredited by NBA for academic years 2018-19 to 2024-25 & valid upto 30.06.2025

URL: www.bnmit.org

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



Vidyaṁruthamashnute

CERTIFICATE

Certified that the project work entitled **Covid Bed Allocation Management System** carried out by **Ms.Afshan Tabassum (1BG20CS006), Ms.Anusha A (1BG20CS017)** are bonafide students of V Semester, **BNM Institute of Technology** in partial fulfillment for the award of Bachelor of Engineering in **COMPUTER SCIENCE AND ENGINEERING** of Visvesvaraya Technological University, Belagavi during the year 2022-23. It is certified that all corrections / suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said Degree

**Mrs. Jayashree
Assistant Professor
Department of CSE
BNMIT, Bengaluru**

**Dr. Chayadevi M L
Professor and HOD
BNMIT, Bengaluru**

**Dr.Krishnamurthy G N
Principal
BNMIT, Bengaluru**

Examiner 1:

Examiner 2:

ABSTRACT

Bed Management in this pandemic has become an issue that almost all of the families of covid patients face. But by using the latest technology this problem can be solved, The main reason for this problem is hospitals not having a proper portal for managing beds. Our solution to this problem can be used to save many lives of covid patients. Therefore we have tried to address this problem with help of our project where the focus is to make sure that every covid patient gets a bed. We have achieved this by creating website for users through which they can request a bed in a hospital and for hospitals where they can manage beds.

In recent times of this pandemic, the load of hospitals for maintaining and organizing their bed capacity has increased tremendously. A computerized system for systematic bed application for reducing the load for hospital staff due to manual booking of beds as well as easy booking of beds for patients is absolutely essential. The patient can interact with the staff using chat portal. The staff can send individual message or broadcast important messages.

ACKNOWLEDGEMENT

The completion of this project brings with a sense of satisfaction, but it is never complete without thanking the persons responsible for its successful completion.

I take this opportunity to express our profound gratitude to Shri. Narayan Rao R Maanay, Secretary, BNMIT, Bengaluru for his constant support and encouragement.

I would like to express my special thanks to Prof. T. J. Rama Murthy, director, BNMIT, Bengaluru and S Y Kulkarni, director, BNMIT, Bengaluru for their constant guidance towards our goals and professions.

I extend my deep sense of sincere gratitude to Dr. Krishnamurthy G.N, Principal, BNMIT, Bengaluru, for providing us the facilities required for the project.

I would also like to thank Prof Eishwar Maanay, Dean Administration, BNMIT, Bengaluru, for providing us useful suggestions required for the project.

I express my in-depth, heartfelt, sincere gratitude to Dr. Chayadevi ML, Professor and H.O.D, Department of Computer Science and Engineering, BNMIT, Bengaluru, for her valuable suggestions and support.

I extend my heartfelt, sincere gratitude to Prof. Jayashree, Assistant Professor, Department of Computer Science and Engineering, BNMIT, Bengaluru, for completion of the project.

Finally, I would like to thank all the faculty members of Department of Computer Science and Engineering, BNMIT, Bengaluru, for their support. I would like to thank our family and friends for their unfailing moral support and encouragement.

Afshan Tabassum

1BG20CS006

Anusha A

1BG20CS017

TABLE OF CONTENTS

CONTENTS	Page No
ABSTRACT	I
ACKNOWLEDGEMENT	II
1. INTRODUCTION	
1.1 Overview of Database Management Systems	1
1.2 Problem statement	2
1.3 Objectives	2
1.4 Dataset Description	2
2. SYSTEM REQUIREMENTS	
2.1 Software & Hardware	4
3. SYSTEM DESIGN	
3.1 E R Diagram	5
3.2 Schema Diagram	6
3.3 Overview of GUI	7
3.4 Normalization	8
4. IMPLEMENTATION	
4.1 Table creation	10
4.2 Description of Table	12
4.3 Populated Tables	14
4.4 SQL Triggers & Stored Procedures	17
4.5 Database connectivity	20
4.6 Modules	21
5. RESULTS	23
6. CONCLUSION & FUTURE ENHANCEMENTS	30
7. REFERENCES	

LIST OF FIGURES

Chapter No.	Figure No.	Description	Page No.
3	Figure 3.1	E-R Diagram	05
	Figure 3.2	Schema diagram	06
	Figure 3.4.2	Second normal form	09
4	Figure 4.2	Description of tables	12
	Figure 4.3	Populated tables	14
	Figure 4.4.1	Triggers	17
	Figure 4.4.2	Stored Procedures	18
5	Figure 5.1	Home page	23
	Figure 5.2	Admin page	24
	Figure 5.3	Hospital login	24
	Figure 5.5	User login	25
	Figure 5.7	Add Hospital Data	26
	Figure 5.8	Hospitals Data	27
	Figure 5.9	Book Bed Slot	27
	Figure 5.10	Updated Data	28
	Figure 5.11	Patient details	28
	Figure 5.12	Logout page	29

Chapter 1

INTRODUCTION

1.1 Overview of Database Management System

A Database is a collection of related data organized in a way that data can be easily accessed, managed and updated. Any piece of information can be a data, for example name of your school. Database is actually a place where related piece of information is stored and various operations can be performed on it. A DBMS is a software that allows creation, definition and manipulation of database. DBMS is actually a tool used to perform any kind of operation on data in database. DBMS also provides protection and security to database. It maintains data consistency in case of multiple users. Here are some examples of popular DBMS, Sql, Oracle, Sybase, Microsoft Access and IBM DB2.

The database system can be divided into four components:

- The database system can be divided into System developer and End users.
- Database application: Database application may be Personal, Departmental, Enterprise and Internal
- DBMS: Software that allow users to define, create and manages database access, Ex: Sql, Oracle etc.
- Database: Collection of logical data.

Functions of database management system:

- Provides Recovery services
- Provides utility
- Provides data Independence
- Provides a clear and logical view of the process that manipulates data.

Advantages of DBMS:

- Segregation of application program
- Minimal data duplicity
- Reduced development time and maintenance need
- Easy retrieval of data

1.2 Problem Statement

Our problem is to implement covid-19 bed allocation management system for the purpose of checking the vacancy of hospital beds and also allotment of beds to patients through online portal and making it easier to manage the appointments with the hospitals.

1.3 Objective

The objective of this project is to

- The main objective of the project is to design and develop a user friendly-system
- Easy to use and an efficient computerized system.
- To develop an accurate and flexible system, it will eliminate data redundancy.
- To study the functioning of Students management System.
- To make a software fast in processing, with good user interface.
- To make software with good user interface so that user can change it and it should be used for a long time without error and maintenance.
- To provide synchronized and centralized farmer and seller database.
- Computerization can be helpful as a means of saving time and money.
- To provide better Graphical User Interface (GUI).
- Less chances of information leakage.
- Provides Security to the data by using login and password method.
- To provide immediate storage and retrieval of data and information.
- Improving arrangements for students coordination.
- Reducing paperwork.

1.4 Dataset Description

Covid Bed Allotment Management system allows the users to look at the home page and check the availability of beds. If they found a vacant hospital bed, they can fill all their details, upload

their details and submit the application form. The admin and users of the hospital can login using their credentials and review the applicants. They can update the status of the applicants, add a new applicant or delete applicants as well. They can list, delete and maintain bed vacancies. The admin has additional authority to change the contact, email, about information and cover image of the main website.

The home page of the Covid Bed Allotment Management system has information about the hospitals and the various beds listings. The users can book their bed through this page.

The modules used in this system are : Admin, Hospitaluser, Hospitaldata, Trig, User and Bookpatient

- Admin module :Admin can manage the vacancies and applications, and status category. The admin can also change the about, cover image, email, name and contact of the home page.
- HospitalUser module : HospitalUser manages the email, password and hospital code .
- Trig :All of the current bed listings are managed here. New bed availability can be added or registered ones can be deleted. The status of the vacancies can be changed to either open or closed.
- User category : This contains information about the various users. Each user is assigned a SRFID by which they can login. They can be book or unbook the bed according to their requirements and availabilities.
- Bookpatient category : The information provided by the user (i.e Patient details) will be saved in this particular part like Hospital code, Bedtype, patient address, patient phonenumber and SRFID.

The tables along with their attributes are:

1. Admin (Username, Password)
 2. HospitalUser (Email, Password, HOCE)
 3. HospitalData (NBED, VBED, ICUBED, HICUBED, HName, HOCE)
 4. Trig (NBED, VBED, DATE, HICUBED, ICUBED, HOCE, Query)
 5. User (Email, DOB, SRFID)
 6. Bookpatient (HOCE, Bedtype, PAddress, Pphno, SRFID, Sp02);
-

Chapter 2

SYSTEM REQUIREMENTS

2.1 Software and Hardware

Software Configuration:

Operating system: Windows 10 ,64 bit

Front end: Html, CSS, Javascript, Bootstrap

Server side language: Php

Back end: Python

Database: Mysql

Web server: Apache

Browser: Chrome

Application software: XAMPP

Hardware Configuration:

Processor: Intel Core i3

RAM: 4 GB

Hard disk: 400GB

Chapter 3

SYSTEM DESIGN

3.1 E-R Diagram

An entity-relationship diagram (ERD) is a data modeling technique that graphically illustrates an information system's entities and the relationships between those entities. An ERD contains different symbols and connectors that visualize two important information: The major entities within the system scope and the inter relationships among these entities. At first look, an ER diagram looks very similar to the flowchart. However, ER Diagram includes many specialized symbols, and its meanings make this model unique. The purpose of ER Diagram is to represent the entity framework infrastructure.

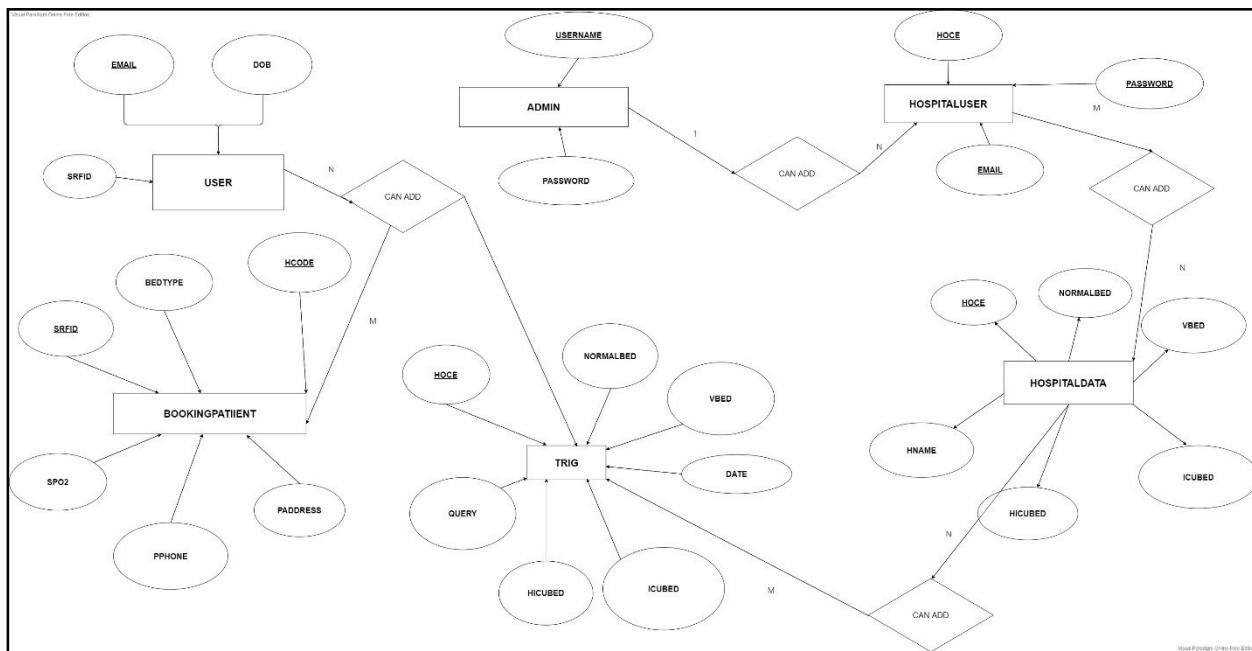


Fig:3.1 E-R Diagram of Covid Bed Allocation Management System

The above Figure:3.1 illustrates the entity Relation Diagram of Covid Bed Allotment Management System. The entities are Admin, HospitalUser, HospitalData, Trig, User and Bookpatient.

3.2 Schema Diagram

A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and the relations among them are associated. It formulates all the constraints that are to be applied on data. A database schema defines its entities and relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams.

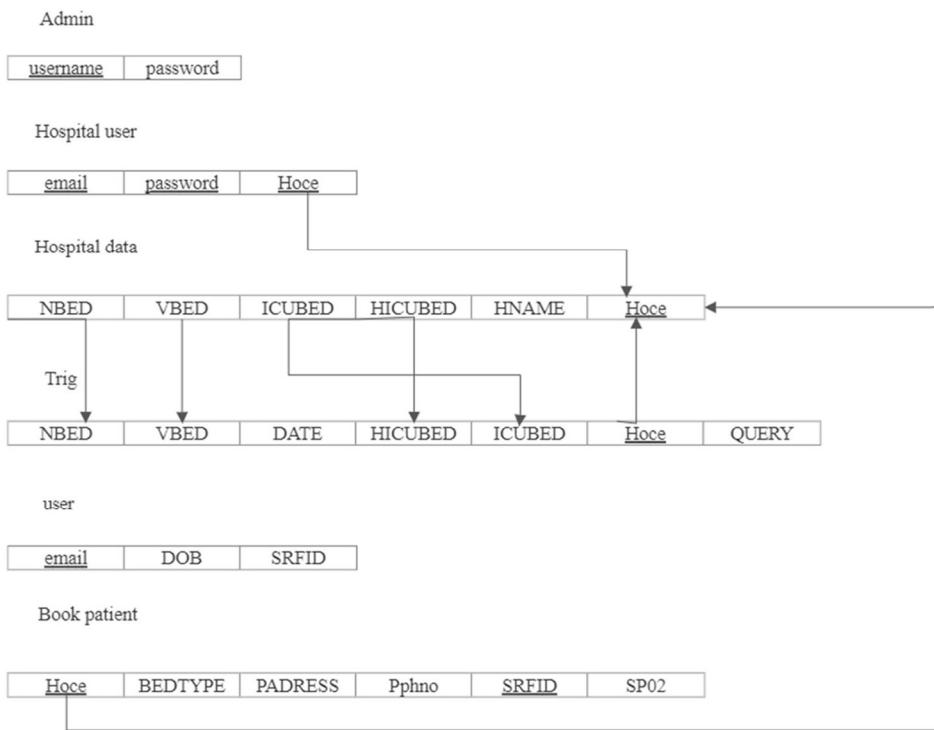


Fig:3.2 Schema Diagram

The above Figure:3.2 depicts the Schema diagram of Covid Bed Allotment Management System. It shows the various relations and references between entities.

3.3 Overview of GUI

GUI is a program interface that takes advantage of the computer's graphics capabilities to make the program easier to use. Well-designed graphical user interfaces can free the user from learning complex command languages. On the other hand, many users find that they work more effectively with a command-driven interface, especially if they already know the command language.

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript and PHP. 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. HTML can embed programs written in a scripting language such as PHP, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content.

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate '.css' file, and reduce complexity and repetition in the structural content.

The PHP Hypertext Preprocessor (PHP) is a programming language that allows web developers to create dynamic content that interacts with databases. PHP is basically used for developing web based software applications. PHP code is usually processed on a web server by a PHP interpreter implemented as a module, a daemon or as a Common Gateway Interface (CGI) executable. On a web server, the result of the interpreted and executed PHP code – which may be any type of data, such as generated HTML or binary image data – would form the whole or part of an HTTP response.

3.4 Normalization

Normalization is a process of analyzing the given relation schema based on their functional dependencies and primary key to achieve desirable properties of minimizing redundancy and minimizing insert, delete, update anomaly. The normalization process takes a relation schema through a series of tests to certify whether it satisfies a certain normal form. The normal form of a relation refers to the highest normal form condition that it meets, and hence the degree to which it has been normalized.

Normalization rule are divided into following normal form:

1. First Normal Form
2. Second Normal Form
3. Third Normal Form
4. Boyce-code Normal Form

3.4.1 First Normal Form

First normal form states that the domain of an attribute must include only atomic (simple, individual) values and that the value of any attribute in a tuple must be a single value from the domain of attribute.

Consider the relations of Covid Bed Allotment Management System, all the relations are in 1NF as they have neither any multivalued attributes nor any composite attributes. Hence the relations are said to be in 1NF.

3.4.2 Second Normal Form

Second normal form is based on the concept of full functional dependency. A functional dependency $X \rightarrow Y$ is a full functional dependency if removal of any attribute A from X means that the dependency does not hold anymore. A relation schema R is in 2NF if every nonprime attribute A in R is fully functionally dependent on the primary key of R.

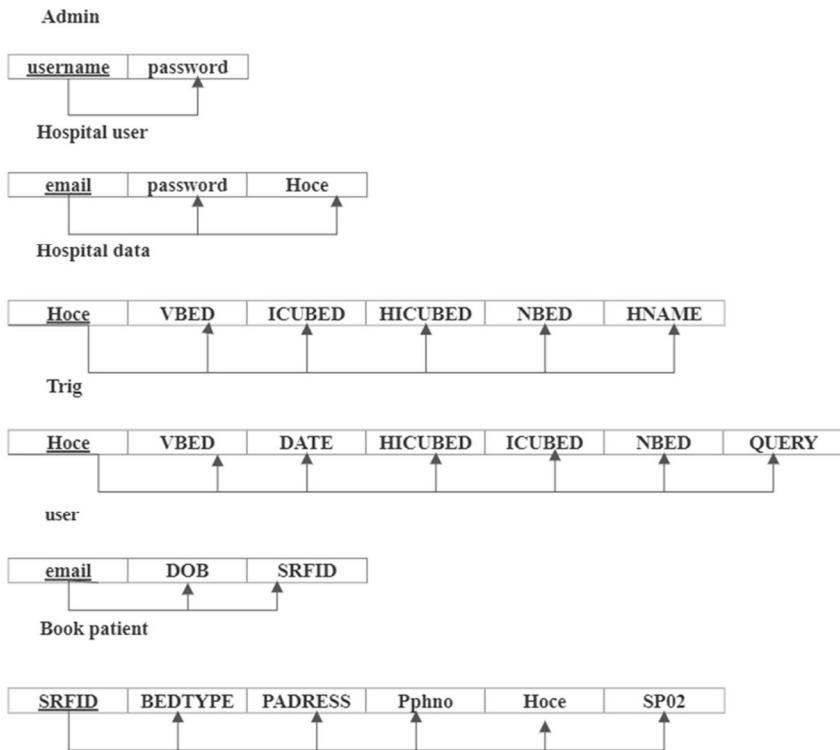


Fig:3.4.2 Second normal form

Consider the relations shown above in figure:3.4.2. Here all the relations are in 2NF as all the nonprime attributes are fully functionally dependent on the set of prime attributes. Hence the relations are in 2NF.

3.4.3 Third Normal Form

Third normal form is based on the concept of transitive dependency. A relation schema R is in 3NF if it satisfies 2NF and no nonprime attribute of R is transitively dependent on the primary key. A relation schema R is in 3NF if every nonprime attribute of R meets both of the following conditions:

- It is fully functionally dependent on every key of R.
- It is non transitively dependent on every key of R.

The relations used in this database are fully functionally dependent on its key attribute and does not hold any transitive dependencies. Hence all the relations are in 3NF.

Chapter 4

IMPLEMENTATION

3.1 Table Creation

```
CREATE TABLE bookingpatient (
    id int(11) NOT NULL,
    srfid varchar(50) NOT NULL,
    bedtype varchar(50) NOT NULL,
    hcode varchar(50) NOT NULL,
    spo2 int(11) NOT NULL,
    pname varchar(50) NOT NULL,
    pphone varchar(12) NOT NULL,
    paddress text NOT NULL)
ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
CREATE TABLE hospitaldata (
    id int(11) NOT NULL,
    hcode varchar(200) NOT NULL,
    hname varchar(200) NOT NULL,
    normalbed int(11) NOT NULL,
    hicubed int(11) NOT NULL,
    icubed int(11) NOT NULL,
    vbed int(11) NOT NULL)
ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
CREATE TABLE hospitaluser (
    id int(11) NOT NULL,
    hcode varchar(20) NOT NULL,
    email varchar(100) NOT NULL,
    password varchar(1000) NOT NULL)
ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
CREATE TABLE test (
    id int(11) NOT NULL,
    name varchar(50) NOT NULL)
ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
CREATE TABLE trig (
    id int(11) NOT NULL,
    hcode varchar(50) NOT NULL,
    normalbed int(11) NOT NULL,
    hicubed int(11) NOT NULL,
    icubed int(11) NOT NULL,
    vbed int(11) NOT NULL,
    querys varchar(50) NOT NULL,
    date date NOT NULL)
ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
CREATE TABLE user (
    id int(11) NOT NULL,
    srfid varchar(20) NOT NULL,
    email varchar(100) NOT NULL,
    dob varchar(1000) NOT NULL)
ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

3.2 Description of Table

In sql we can use the command “desc table_name” or “describe table_name” to describe the list of column definitions for the specified table. We obtain information such as column name, whether the column allows NULL or not and the datatype of the column. The following figures show the description of all the tables used in this system.

Desc bookingpatient;

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id 	int(11)		No	<i>None</i>		AUTO_INCREMENT	 Change  Drop More	
<input type="checkbox"/>	2	srfid 	varchar(50)	utf8mb4_general_ci	No	<i>None</i>			 Change  Drop More	
<input type="checkbox"/>	3	bedtype	varchar(50)	utf8mb4_general_ci	No	<i>None</i>			 Change  Drop More	
<input type="checkbox"/>	4	hcode	varchar(50)	utf8mb4_general_ci	No	<i>None</i>			 Change  Drop More	
<input type="checkbox"/>	5	spo2	int(11)		No	<i>None</i>			 Change  Drop More	
<input type="checkbox"/>	6	pname	varchar(50)	utf8mb4_general_ci	No	<i>None</i>			 Change  Drop More	
<input type="checkbox"/>	7	pphone	varchar(12)	utf8mb4_general_ci	No	<i>None</i>			 Change  Drop More	
<input type="checkbox"/>	8	paddress	text	utf8mb4_general_ci	No	<i>None</i>			 Change  Drop More	

Fig:4.2.1 Description of bookingpatient table

The above figure:4.2.1 shows all the attributes of the table bookingpatient and their types. All the users listed in this table have access to the database.

Desc hospitaldata;

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id 	int(11)		No	<i>None</i>		AUTO_INCREMENT	 Change  Drop More	
<input type="checkbox"/>	2	hcode 	varchar(200)	utf8mb4_general_ci	No	<i>None</i>			 Change  Drop More	
<input type="checkbox"/>	3	hname	varchar(200)	utf8mb4_general_ci	No	<i>None</i>			 Change  Drop More	
<input type="checkbox"/>	4	normalbed	int(11)		No	<i>None</i>			 Change  Drop More	
<input type="checkbox"/>	5	hicubed	int(11)		No	<i>None</i>			 Change  Drop More	
<input type="checkbox"/>	6	icubed	int(11)		No	<i>None</i>			 Change  Drop More	
<input type="checkbox"/>	7	vbed	int(11)		No	<i>None</i>			 Change  Drop More	

Fig:4.2.2 Description of hospitaldata table

The above figure:4.2.2 lists all the attributes of the table hospitaldata and their types. The hospitaldata table holds the list of bed vacancies.

Desc hospitaluser;

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id 	int(11)			No	<i>None</i>		AUTO_INCREMENT	 Change  Drop More
<input type="checkbox"/>	2	hcode	varchar(20)	utf8mb4_general_ci		No	<i>None</i>			 Change  Drop More
<input type="checkbox"/>	3	email	varchar(100)	utf8mb4_general_ci		No	<i>None</i>			 Change  Drop More
<input type="checkbox"/>	4	password	varchar(1000)	utf8mb4_general_ci		No	<i>None</i>			 Change  Drop More

Fig:4.2.3 Description of hospitaluser table

The above figure:4.2.3 shows all the attributes of the table hospitaluser and their types.

Desc test;

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id 	int(11)			No	<i>None</i>		AUTO_INCREMENT	 Change  Drop
<input type="checkbox"/>	2	name	varchar(50)	utf8mb4_general_ci		No	<i>None</i>			 Change  Drop

Fig:4.2.4 Description of test table

The above figure:4.2.4 shows all the attributes of the table test and their types. This table contains all the information about the admin.

Desc trig;

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id 	int(11)			No	<i>None</i>		AUTO_INCREMENT	 Change  Drop More
<input type="checkbox"/>	2	hcode	varchar(50)	utf8mb4_general_ci		No	<i>None</i>			 Change  Drop More
<input type="checkbox"/>	3	normalbed	int(11)			No	<i>None</i>			 Change  Drop More
<input type="checkbox"/>	4	hicubed	int(11)			No	<i>None</i>			 Change  Drop More
<input type="checkbox"/>	5	icubed	int(11)			No	<i>None</i>			 Change  Drop More
<input type="checkbox"/>	6	vbed	int(11)			No	<i>None</i>			 Change  Drop More
<input type="checkbox"/>	7	querys	varchar(50)	utf8mb4_general_ci		No	<i>None</i>			 Change  Drop More
<input type="checkbox"/>	8	date	date			No	<i>None</i>			 Change  Drop More

Fig:4.2.5 Description of trig table

Desc user;

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id 	int(11)		No	<i>None</i>		AUTO_INCREMENT	 Change  Drop More	
<input type="checkbox"/>	2	srfid 	varchar(20)	utf8mb4_general_ci	No	<i>None</i>			 Change  Drop More	
<input type="checkbox"/>	3	email	varchar(100)	utf8mb4_general_ci	No	<i>None</i>			 Change  Drop More	
<input type="checkbox"/>	4	dob	varchar(1000)	utf8mb4_general_ci	No	<i>None</i>			 Change  Drop More	

Fig:4.2.6 Description of user table

3.3 Populated Tables

The SELECT statement is used to select data from a database. The data returned is stored in a result table, called the result-set. We can use the command “select * from table_name” to obtain all the values of the table. The following figures show the values of each table.

Select * from bookingpatient;

id	srfid	bedtype	hcode	spo2	pname	pphone	paddress
1	aa11	nbed	BB11	12	Abhi	1234567898	Mysore
2	ss22	icubed	AABB22	16	Raksha	7897876543	Mumbai
7		NormalBed	AA1100	12	afshan	8197422159	bsk
8	anuafs	NormalBed	AA1100	15	afshan	8197422159	bsk

Fig:4.3.1 Values of bookingpatient table

The above figure:4.3.1 lists the populated table bookingpatient with the list of all users.

Select * from hospitaldata;

id	hcode	hname	normalbed	hicubed	icubed	vbed
2	AABB11	Confido	10	8	11	6
3	CC11	Victoria	11	12	13	9
4	BB44	Manipal	22	12	17	7
7	AA1100	Apolo	9	3	-6	5

Fig:4.3.2 Values of hospitaldata table

The above figure:4.3.2 lists the populated table hospitaldata with the list of all available beds.

Select * from hospitaluser;

id	hcode	email	password
3	AABB22	sakshi123@gmail.com	sakshi
4	BB11	anilk77@gmail.com	anil22
15	AA1100	afshant213@gmail.com	pbkdf2:sha256:260000\$rxdg6xKjd49J8taG\$03384cb0fbf9...
16	AA1100	anua72495@gmail.com	pbkdf2:sha256:260000\$GCoRTVgaJr9nmTok\$5d603098252a...

Fig:4.3.3 Values of hospitaluser table

The above figure:4.3.3 lists the populated table hospitaluser with the list of all email and password.

Select * from test;

id	name
1	anees
2	rehman
3	Ana
4	Rahul

Fig:4.3.4 Values of test table

The above figure:4.3.4 lists the populated table test.

Select * from trig;

id	hcode	normalbed	hicubed	icubed	vbed	querys	date
1	BBH01	50	9	2	1	UPDATED	2021-11-26
2	BBH01	50	9	2	1	DELETED	2021-11-26
3	AA1100	15	5	4	2	INSERTED	2021-11-26
4	AA1100	15	10	8	2	UPDATED	2021-11-26
5	AA1100	15	10	7	2	UPDATED	2021-11-26
6	ARK123	12	55	22	22	INSERTED	2022-01-12
7	ARK123	12	50	22	22	UPDATED	2022-01-12
8	ABCD123	11	15	4	20	INSERTED	2022-01-12
9	ABCD123	11	11	4	20	UPDATED	2022-01-12
10	ARK123	12	50	21	22	UPDATED	2022-01-12
11	MAT123	40	4	4	1	DELETED	2022-01-30
12	AA1100	15	10	7	2	DELETED	2022-01-30
13	ARK123	12	50	21	22	DELETED	2022-01-30
14	ABCD123	11	11	4	20	DELETED	2022-01-30
15	AA1100	11	3	-5	5	INSERTED	2023-01-14
16	AA1100	11	3	-6	5	UPDATED	2023-01-14

Fig:4.3.5 Values of trig table

The above figure:4.3.5 lists the populated trig table.

Select * from user;

id	srfid	email	dob
1	aa11	deep11@gmail.com	07/09/2004
3	rr33	liki89@gmail.com	09/01/2002
4	ss22	afshan234@gmail.com	12/2/2001
13	anuafs	anua72495@gmail.com	pbkdf2:sha256:260000\$FybzbPczSbzdgauIT\$8f4304902395...

Fig:4.3.6 Values of user table

The above figure:4.3.6 lists the populated table user.

3.4 SQL Triggers and Stored Procedures

3.4.1 Trigger

A database trigger is procedural code that is automatically executed in response to certain events on a particular table or view in a database. The trigger is mostly used for maintaining the integrity of the information on the database. Triggers execute when a user tries to modify data through a data manipulation language (DML) event. DML events are INSERT, UPDATE, or DELETE statements on a table or view.

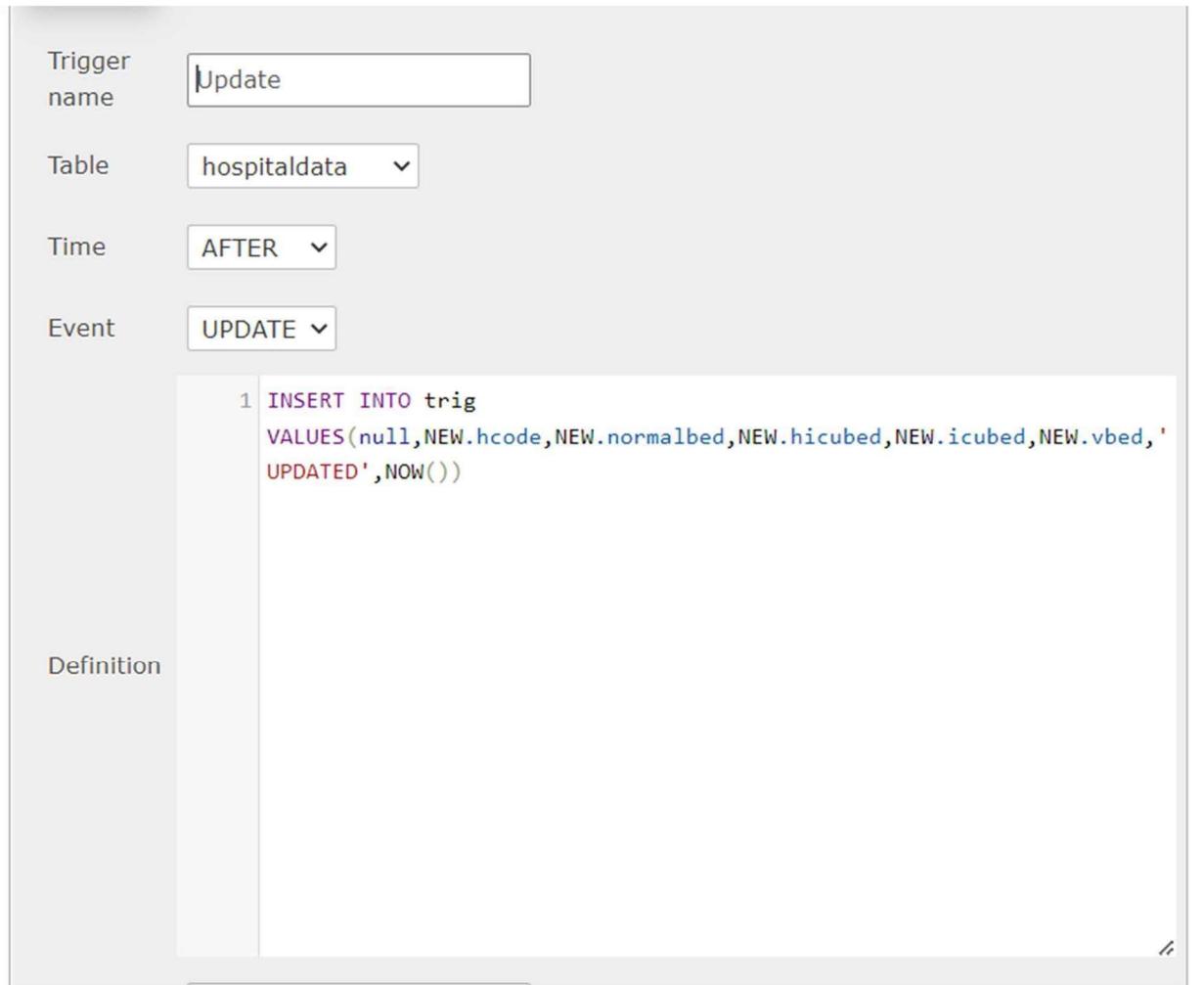


Fig:4.4.1 Screen capture of trigger

As seen in the above figure:4.4.1, we are updating the details of hospitaldata.

3.4.2 Stored Procedure

A stored procedure is a set of Structured Query Language (SQL) statements with an assigned name, which are stored in a relational database management system as a group. So if a query has to be written over and over again, instead of having to write that query each time, it can be saved as a stored product and can be executed just by calling the procedure. In addition, parameters can also be passed to the stored procedure. So depending on the need, the stored procedure can act accordingly.

Details

Routine name	getusers															
Type	PROCEDURE															
Parameters	<table border="1"><thead><tr><th>Direction</th><th>Name</th><th>Type</th><th>Length/Values</th><th>Options</th></tr></thead><tbody><tr><td colspan="5">Add parameter</td></tr><tr><td>1</td><td colspan="4">SELECT * FROM user</td></tr></tbody></table>	Direction	Name	Type	Length/Values	Options	Add parameter					1	SELECT * FROM user			
Direction	Name	Type	Length/Values	Options												
Add parameter																
1	SELECT * FROM user															
Definition																
Is deterministic	<input type="checkbox"/>															

Go **Close**

Fig 4.4.2 Screen capture of stored procedure

The above figure:4.4.2 shows a stored procedure of users.

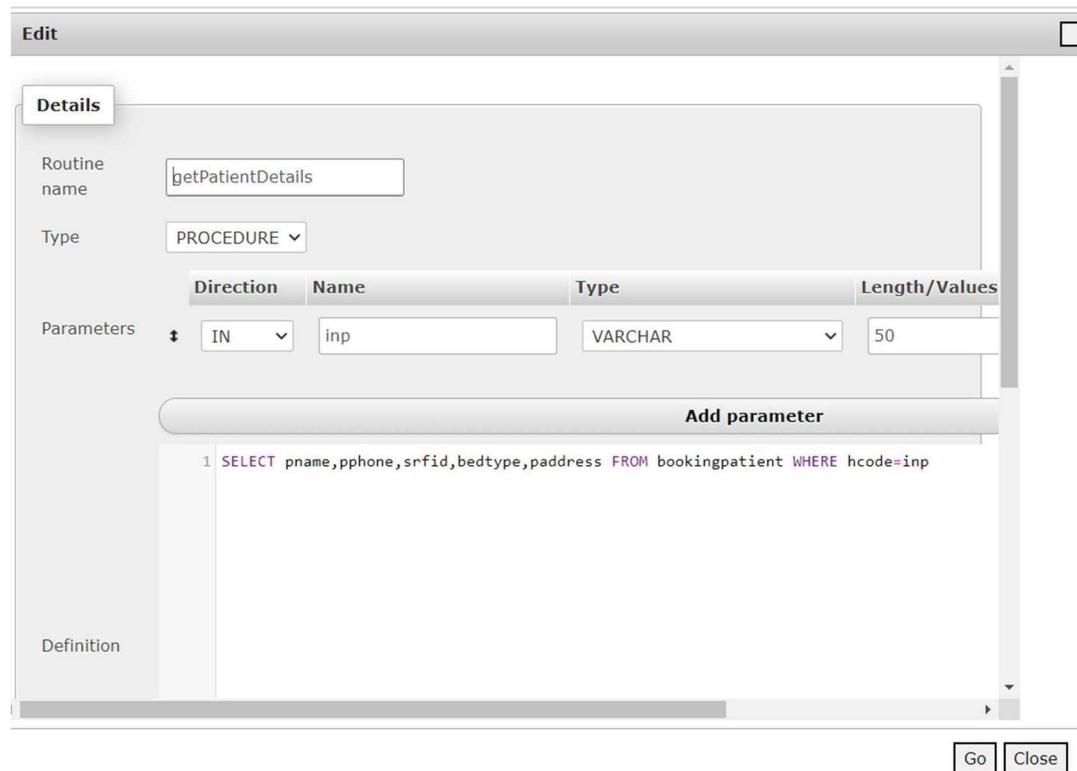


Fig 4.4.3 Screen capture of stored procedure

The above figure:4.4.3 shows a stored procedure that update the details of patient.

3.5 Database Connectivity

A Database connection is a facility in computer science that allows client software to talk to database server software, whether on the same machine or not. A connection is required to send commands and receive answers, usually in the form of a result set. PHP has a pretty straight forward method to working with MySQL databases.

There are five steps to make PHP database interaction

1. Create a connection
2. Select database
3. Perform database query
4. Use return data
5. Close connection

```
<?php
session_start()
//1. Create a database connection
$conn=mysql_connect('localhost','root','');
if($conn)
echo "Connection Successful";

//2. Select a database to use
$db=mysql_select_db('covid_db',$conn);
If($db)
echo "Database selected!";

//3. Perform database query
$sql="Select username,password from user";
If(mysql_query($sql,$conn)
echo "Rows selected";
$result = mysql_query($sql,$conn);
//4. Use returned data
While($row=mysql_fetch_array($result) {
echo $row['username'];
echo $row['password'];
}
//5. Close connection
mysql_close($conn);
?>
```

3.6 Modules

The covid bed allocation management system consists of the following modules:

- Admin login Page : This is the page where the admin can login using their username and password to access the hospital details.
- Hospitaluser login Page : This is the page where the user enter their valid credentials along with the particular hospital code in which they want to search for the availability of beds.
- User login Page: This is the page where all the users can login using their valid credentials .
- patientdetails Page: This is the page where a particular patient's details are listed and can be managed or viewed by admin.
- Available beds Page : This page will give the information about number of beds available in the hospital to the users .
- Operations Page : This page will show the insertion, deletion and updation of the beds in particular hospital depending on the registrations.

Chapter 5

RESULT

This chapter contains screenshots of the final results of the website and its various modules.

Home page : The below figure:5.1 shows the home page which contain list of available beds, sign in pages, contact of hospital, and patient details . The users can search for required type of bed using the “Available beds” Button. The user can click on any particular button to view information about it.

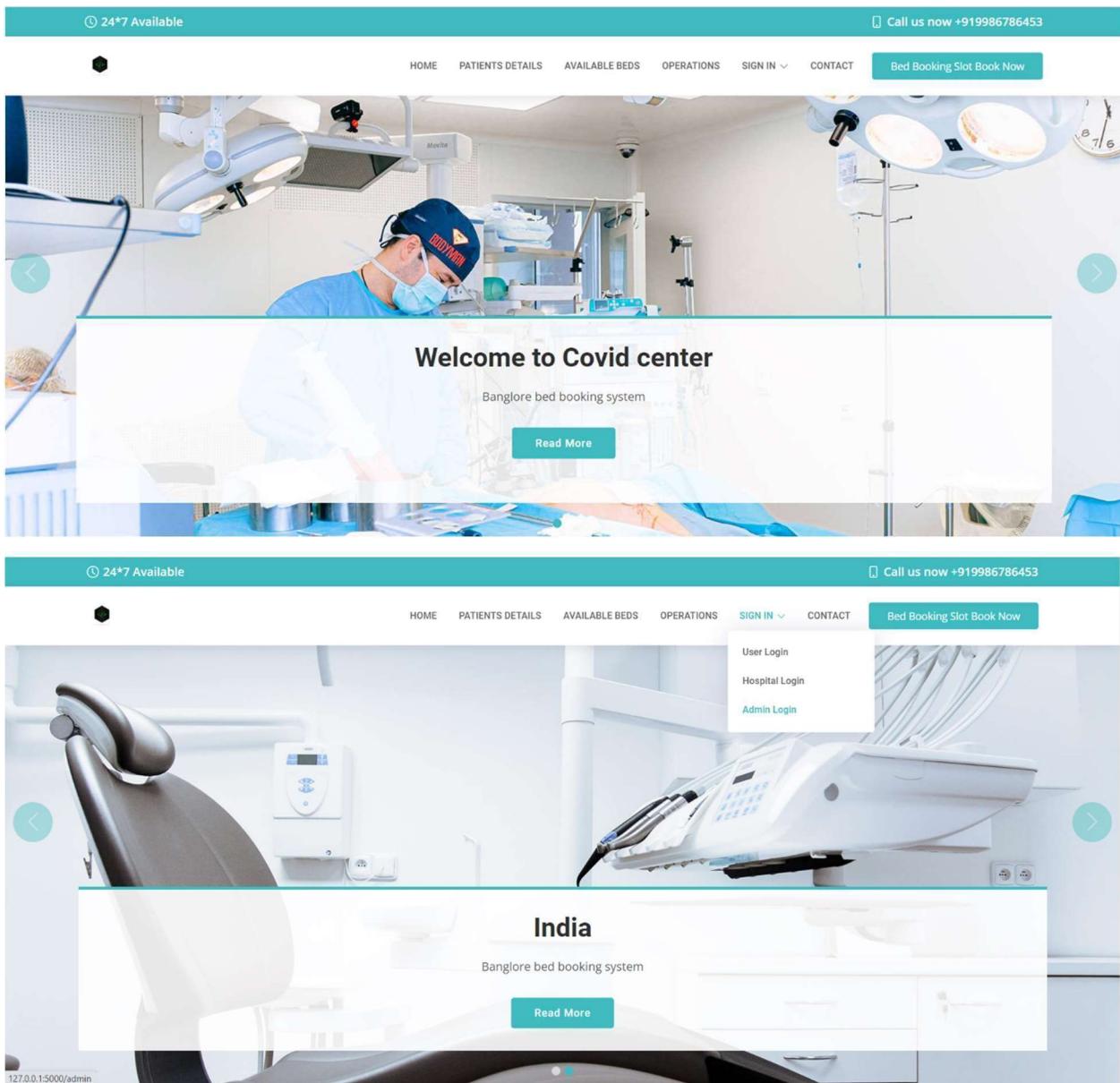


Fig:5.1 Screen capture of home page

Covid Bed Allocation Management System

Admin login page : In the below figure:5.2, the admin has to provide his valid credentials like username and password to access the page.

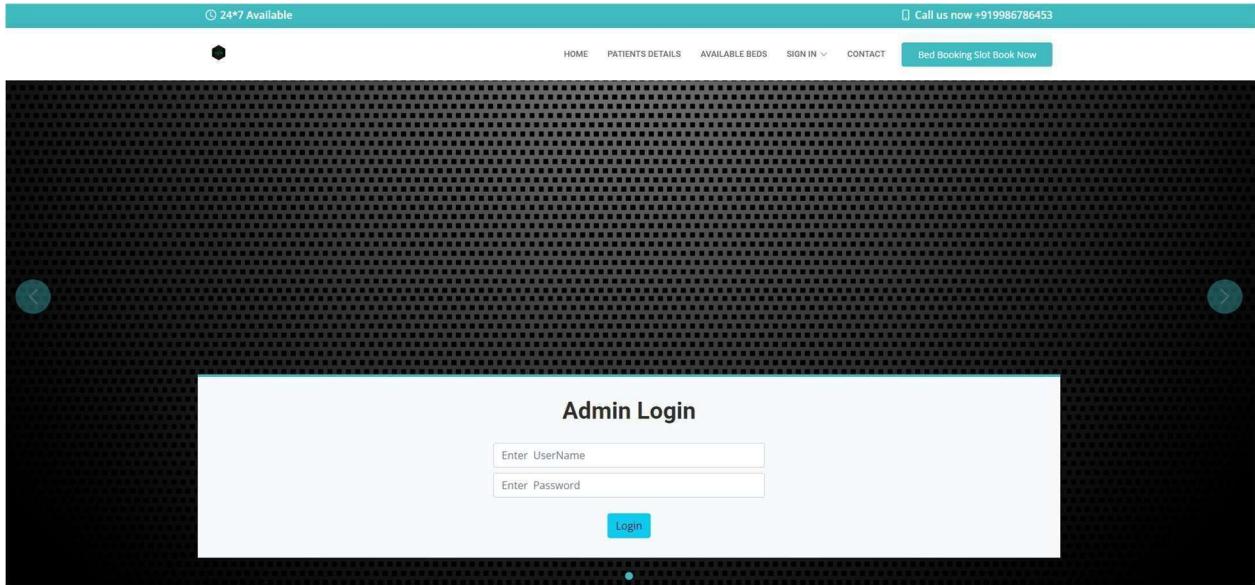


Fig:5.2 Screen capture of Admin login

Hospital login page : When the user clicks on the Hospital login button, a login page is provided as shown in fig 5.3. The user should enter the particular Hospital code in which he/she wants to check for the availability of the beds. By providing email and password they have to login. If the credentials provided are correct then it will show login successful as shown in fig 5.4 .

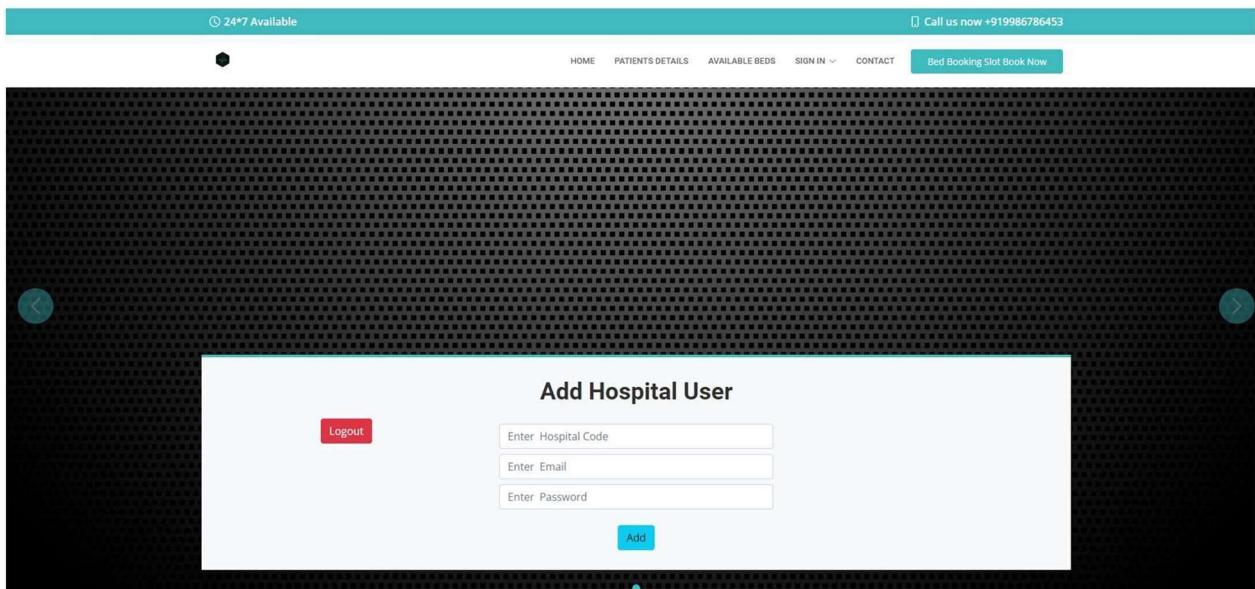


Fig:5.3 Screen capture of Hospital login page

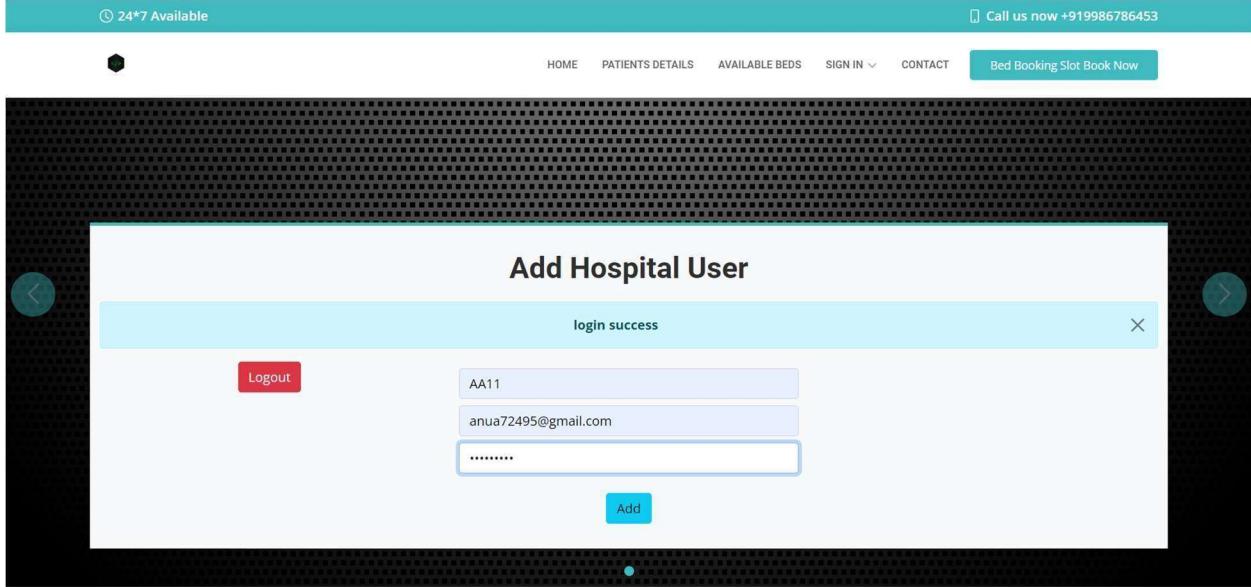


Fig:5.4 Screen capture of Hospital login successful.

User login page : The user should signup first by providing the required credentials as shown in fig 5.5 and then he/she should login as shown in fig 5.6 in order to access the information of the Hospital and to book the required bed.

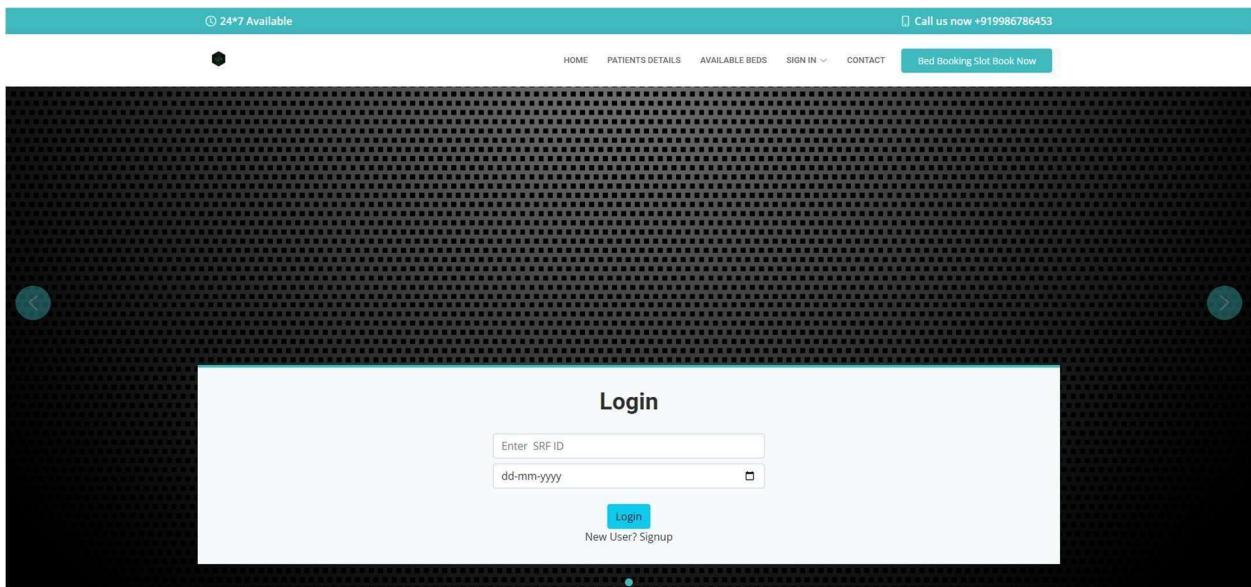


Fig:5.5 Screen capture of user login

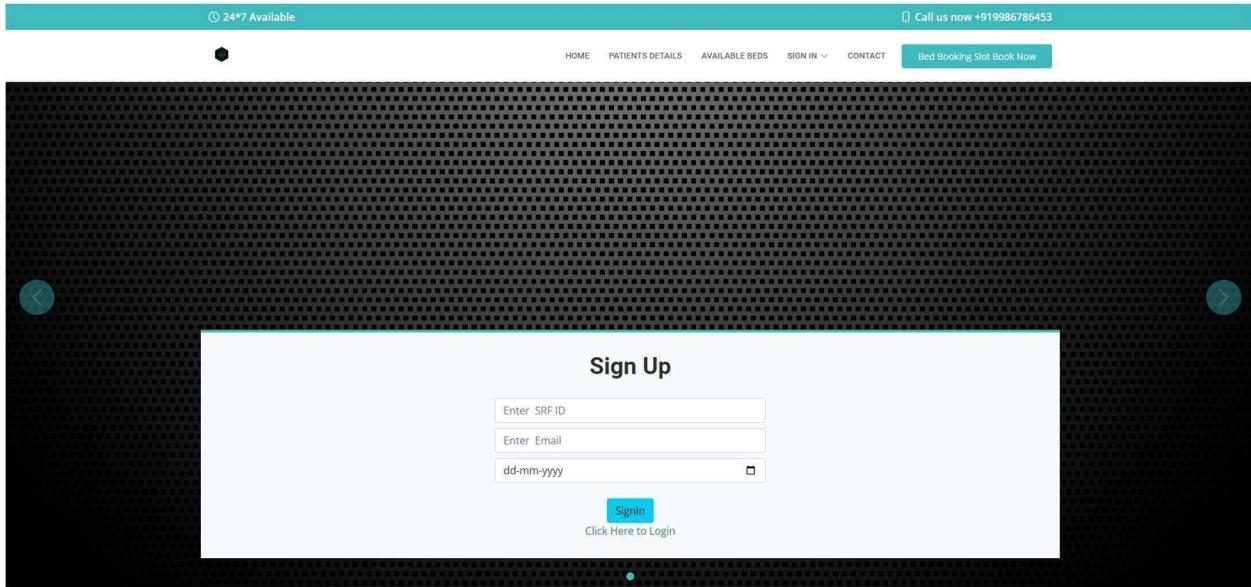


Fig:5.6 Screen capture of user signup

Add Hospital Data : The Hospital data will be added and updated by the hospital staff depending upon the patient admission. The below fig 5.7 shows the details of Archarya Hospital.

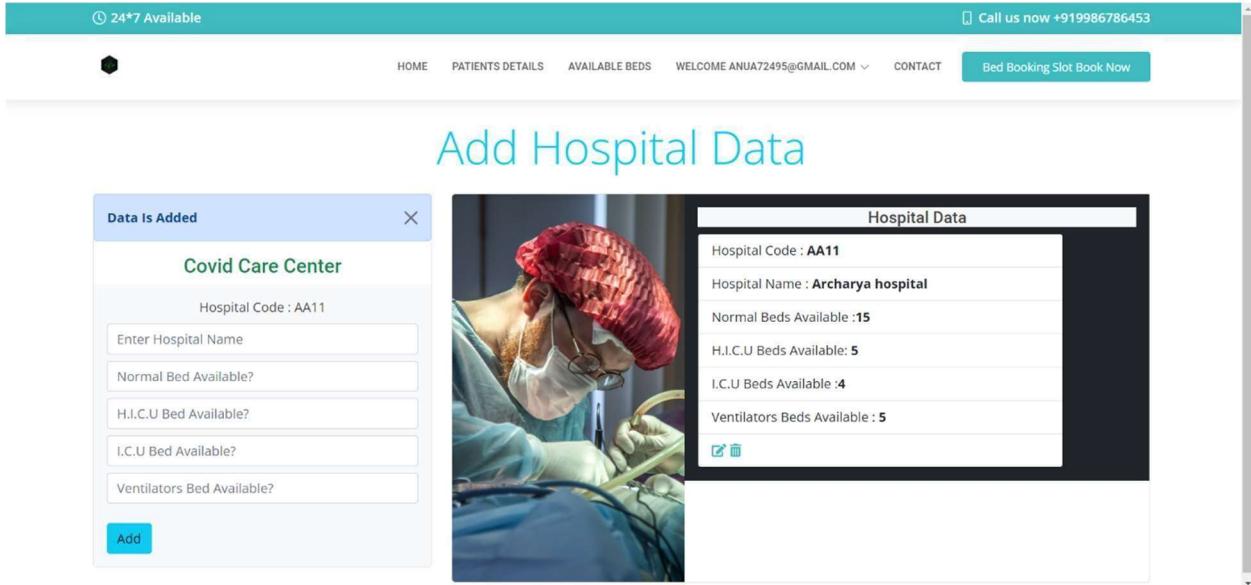


Fig:5.7 Screen capture of adding Hospital Data

The added and updated hospital data will be stored as triggers in separate table as shown in below fig 5.8 .

Triggered Data						
Hospital Code	Normal Bed	HICU Bed	I.C.U Bed	Ventilator Bed	Action	DATE
BBH01	50	9	2	1	UPDATED	2021-11-26
BBH01	50	9	2	1	DELETED	2021-11-26
AA1100	15	5	4	2	INSERTED	2021-11-26
AA1100	15	10	8	2	UPDATED	2021-11-26
AA1100	15	10	7	2	UPDATED	2021-11-26
ARK123	12	55	22	22	INSERTED	2022-01-12
ARK123	12	50	22	22	UPDATED	2022-01-12
ABCD123	11	15	4	20	INSERTED	2022-01-12
ABCD123	11	11	4	20	UPDATED	2022-01-12
ARK123	12	50	21	22	UPDATED	2022-01-12
MAT123	40	4	4	1	DELETED	2022-01-30
AA1100	15	10	7	2	DELETED	2022-01-30
ARK123	12	50	21	22	DELETED	2022-01-30
ABCD123	11	11	4	20	DELETED	2022-01-30
AA11	15	5	4	5	INSERTED	2022-01-26
AA11	15	5	8	5	UPDATED	2022-01-26

Fig:5.8 Screen capture of Hospitals Data

Bed booking page : After sign in the user has click on Bed Booking slot Book Now button to book the bed by providing the details about infected person.

Available Beds					
Hospital Code	Hospital Name	Normal Bed	HICU Bed	I.C.U Bed	Ventilator Bed
AA11	Archarya	15	5	8	5

Fig:5.9 Screen capture of Book Bed Slot

After booking the ICU bed in Archarya hospital , now the number of ICU beds in it will be updated to 7 as shown in below fig 5.10 .

The screenshot shows a web application interface for booking a bed slot. At the top, there's a teal header bar with the text "24*7 Available" and "Call us now +919986786453". Below the header, a navigation bar includes links for "HOME", "PATIENTS DETAILS", "AVAILABLE BEDS", "WELCOME AFSHANT213@GMAIL.COM", and "CONTACT", along with a "Bed Booking Slot Book Now" button.

Book Bed Slot

Available Beds					
Hospital Code	Hospital Name	Normal Bed	HICU Bed	I.C.U Bed	Ventilator Bed
AA11	Archarya	15	5	7	5

Covid Care Center

KA0011
Choose Bed Type
Select hospital code by looking the availability of beds in table right side
Select Hospital Code
Oxygen Level
Patient Name
Patient Phone Number
Patient Address

Book Slot

Fig:5.10 Screen capture of updated data.

Patient page : After booking the bed successfully, the patient details will be updated in the Patient details page which is accessed by the hospital staff to allocate the bed .

The screenshot shows a "Patient Details" page. At the top, there's a teal header bar with the text "24*7 Available" and "Call us now +919986786453". Below the header, a navigation bar includes links for "HOME", "PATIENTS DETAILS", "AVAILABLE BEDS", "WELCOME AFSHANT213@GMAIL.COM", and "CONTACT", along with a "Bed Booking Slot Book Now" button.

Patient Details

DETAILS
KA0011
ICUBed
ICUBed
AA11
89
Afshan
8798656567
BSK Bangalore

Fig:5.11 Screen capture of Patient details page

User logout page :After booking the bed the user can logout by clicking on the logout button as shown in fig 5.12 . Fig 5.13 shows logout successful

The screenshot shows a web-based application for managing COVID-19 bed allocation. At the top, there's a teal header bar with the text "24*7 Available" and a phone number "Call us now +919986786453". Below the header, the main content area has a light gray background. On the left, there's a sidebar with a dark background and circular navigation arrows. In the center, there's a table listing patient details. At the top of the table, there are navigation links: "HOME", "PATIENTS DETAILS", "AVAILABLE BEDS", "WELCOME ANUJA72495@GMAIL.COM", and "CONTACT". To the right of the table, there's a green button labeled "Bed Booking Slot Book Now". The table itself has columns for ID, Age, Gender, and other details. A yellow "Logout" button is visible at the top of the table. The bottom of the screen shows a URL "127.0.0.1:5000/logout".

Fig:5.12 Screen capture of logout page

This screenshot shows the same application after a user has successfully logged out. The interface is identical to Fig 5.12, with the teal header, sidebar, and central table. However, a yellow modal window titled "Login" is displayed in the center. The modal contains the message "Logout SuccessFul" and includes input fields for "Enter SRF ID" and "dd-mm-yyyy", along with a "Login" button and a "New User? Signup" link. The background of the main content area is now a solid black color.

Fig:5.13 Screen capture of user logout successful

Chapter 6

CONCLUSION AND FUTURE ENHANCEMENTS

6.1 Conclusion

The proposed Covid Bed Allocation management system provides a user-friendly platform for users to book a bed in particular hospital. It makes searching and booking of beds easier. The system will help the hospital to insert, delete and update the number of beds . Shortage of resources, such as hospital beds, needed for health care especially in times of crisis can be a serious challenge for many countries. Depending on the type of hospital wards and the health condition of patients, this method can help users to identify hospitals with the best potential and emergency services for the purpose of allocation of resources, which can help to save the patient from virus.

The project teaches us the essential skills like:

- Understanding the database handling and query processing.
- Implement, analyze and evaluate the project developed for an application.
- Demonstrate the working of different concepts of DBMS.

6.2 Future Enhancements

The system is designed in such a way that provisions can be given for further enhanced without affecting the system presently developed. The enhancements that can be incorporated are:

- Creating an Covid Bed Allocation app which will be more easier.
- To make easier for the users the website can mark the hospitals as Red, Green and Yellow. Red means no beds available, yellow means less than 50 beds available and Green means more than 50 beds are available.
- An additional page for providing oxygen cylinders for home isolation by uploading patients valid photo ID, Aadhaar card and COVID-19 positive report.

REFERENCES

- 1 <https://www.youtube.com/watch?v=xgOW-hNdX5A&t=2873s>
- 2 <https://arkprocoder.com/#/project/2>