CHAPTER-1

INTRODUCTION

1.1 PROBLEM STATEMENT

"Bed Management in the pandemic has become an issue that almost all the families of covid patients faced during the pandemic of covid. On considering this not only for this purpose of covid treatment but includes for all kind of emergencies for the need and supply for the bed we can solve this problem using the latest tools and technologies, 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 two apps one for users through which they can request a bed in a hospital and another one for hospitals where they can manage beds".

1.2 OBJECTIVE OF PROJECT

The main objective of this application is to build an online bed slot booking platform, which allows a user to book a bed in a preferred hospital looking at the number of beds available.

1.3 PURPOSE, SCOPE & APPLICABILITY OF PROJECT

The website is designed to solve this problem by providing a bed booking system to the users. The user will be provided bed status of all hospitals in a city and hospital-wise bed status. The patient can request for bed by uploading their srfid for verification. This is an attempt to stop people from jumping queue and bringing transparency to bed allocating beds .Then at the hospital side, the hospital can see the bed request, and then they can approve or reject the patient's request the patient receives a BOOKING ID to show to the reception when they reach the hospital.

The Project **"**Bed-Slots Booking System" as a wide scope as it is generalized software and can be easily used in any hospital system with little or no change. The Changes in software can be easily accommodated. The addition and deletion of the modules in software can be easily adjusted. This project has a lot of scope for further enhancement too. This project can save money and efforts in managing the record, just a mouse click can make the task easy and faster.

1.4 MAJOR MODULES

Admin module

In this module Admin can add list of hospitals and send email to the hospital containing hospital code and password to login. Admin adds various hospital data using his credentials. Only Admin has the right to add hospital lists. Admin module has several sub-modules in it.

- **Login**: Admin uses user_id and password to login.
- **Add Hospital user**: Adds hospital user consisting of email, hospital code, password.
- **Logout**: Admin logs out of the page.

Hospital module

In Hospital module, listed hospitals can add hospital details like various beds (ICU, Normal, Ventilator etc..) available in their hospital. The entered details can be modified (update, delete). Hospital module has several sub-modules in it. They are

- **Login**: Hospital user uses email id and password to login.
- **Add Hospital Data**: Adds hospital data consisting of hospital bed details.
- **Logout**: Hospital user logs out of the page.

Patient module

Patients can login using unique srfid and email. Based on available beds in various hospitals they can book the type of bed they want. The details entered by the patients can be seen in patient detail section.

- **Login**: User uses srf_id and date-of-birth to login.
- **Booking Bed Slot**: User can book slot among the available beds in the listed hospital.
- **View Available beds**: View the beds available in the listed hospital.
- **Patient Details**: User can view the entered patient details

CHAPTER-2

LITERARY SURVEY

2.1 TRADITIONAL SYSTEM

Earlier people used to remember the transaction made by them or they used to store them in the form of hard copy in big and bulky register, but this system is being imposed they can save those records here only without wasting time and money on this.

The traditional approach usually consisted of custom-built data records and registers with information tailored of that household. There are scenarios where people would not even have their own records, and would just remember the transactions made, where lots of confusions would arise.

Initially, these systems were very simple to use as they hardly had any proper planned system to maintain. People have been doing this for years, but this allowed them to have lot extra expenses and very little savings, proving not to be accurate. However, it has become very difficult for individuals to manage and rely on their data because there was no reliable system for them in place to enforce data standards or management.

2.2 PROS AND CONS OF THE TRADITIONAL APPROACH

Pros

- Simple
  - Matched existing people mindset.
  - People were not as interested in funding complicated information systems.
- Initially low-cost
  - Earlier computing was not viewed as beneficial.
  - Saving data in a hard copy was chosen as it was cheap, in order to save on cost.

Cons

- Lack of maintaining record
  - People hardly made a proper record of data regarding expenses and savings.
  - They were unwilling to monitor the income and expenses with a proper plan.
- Unmanaged redundancy
  - Multiple instances of the same data appearing throughout various files, hard copy, and records.
  - Information updated in one place was not replicated to the other locations.
  - Though a record was maintained somehow, data redundancy was a big drawback.
- Data inconsistency
  - Redundant data stored in various locations was usually never stored the same way.
  - The record maintained was not properly managed.
- Lack of accurate data
  - Same data stored in multiple locations of hard copy maintained.
  - Caused unnecessary doubling of efforts for analysing and managing data.
- Highly unwanted expenses
  - Since the income was rarely managed, by keeping a record of expenses and savings, it led to improper management of household expenses.
  - Doubling of expenses as well as very less savings.

2.3 DOWNFALL OF TRADITIONAL MANAGEMENT SYSTEM

Conceived in a relatively under-developed era, the technique of managing the expenses of one's own income was very hard to be found in those people. They fail to have a proper planning of maintaining their household expenses and savings. People hardly had an idea of having a record of their own to manage their income. There are few scenarios where people had proper record of their expenses and savings. However, such traditional practices fall short in serving the needs of today's expectations and essentials for the following reasons:

- Inadequate failover capabilities
- Latency issues
- Insufficient interests by the individuals
- Lack of proper planning at times
- Increasing expenditure and inadequate savings
- Inability to maintain a proper record.

For all these reasons, such traditional practices are unable to deliver results which are accurate and those which help the individuals to have proper savings and limited expenses. Dedicating all the time to maintain a more precise record and plan was a tiresome task for those people. Hence, due to these various reasons, the downfall of the traditional system was inevitable.

2.4 INTRODUCTION TO DBMS

A Database management system (DBMS) refers to the technology for creating and managing databases. Basically, a DBMS is a software tool to organize (create, retrieve, update and manage) data in a database.

The main aim of a DBMS is to supply a way to store 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. A datum is a unit of Data. Meaningful data combines to form information.

Hence, information is interpreted data- data provided with semantics.MS ACCESS is one of the most common examples of database management software.

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 crash or attempts at unauthorized access.

2.5 INDICATIVE AREAS FOR THE USE OF A DBMS

- Airlines: reservations, schedules etc.
- Telecom: calls made, customer details, network usage etc.
- Universities: registration, results, grades, etc.
- Sales: products, purchases, customers etc. • Banking: all transactions etc.

2.6 ADVANTAGES OF DBMS

A Database Management System has many advantages over the traditional file system used inthe earlier days, such as:

- **Data independen**ce: Application programs should be as free or independent as possible from details of data representation and storage. DBMS can supply an abstract view of the data for insulating application code from such facts.
- **Efficient data access**: DBMS utilize a mixture of sophisticated concepts and techniques for storing and retrieving data competently and this feature becomes important in cases where the data is stored on external storage devices.
- **Data integrity and Security:** If data is accessed through the DBMS, the DBMS can enforce integrity constraint on the data.
- **Data administration:** When several users share the data, integrating the administration of data can offer major improvements. Experienced professionals understand the nature of the data being managed and can be responsible for organizing the data representation to reduce redundancy and make the data to retrieve efficiently.

2.7 COMPONENTS OF DBMS

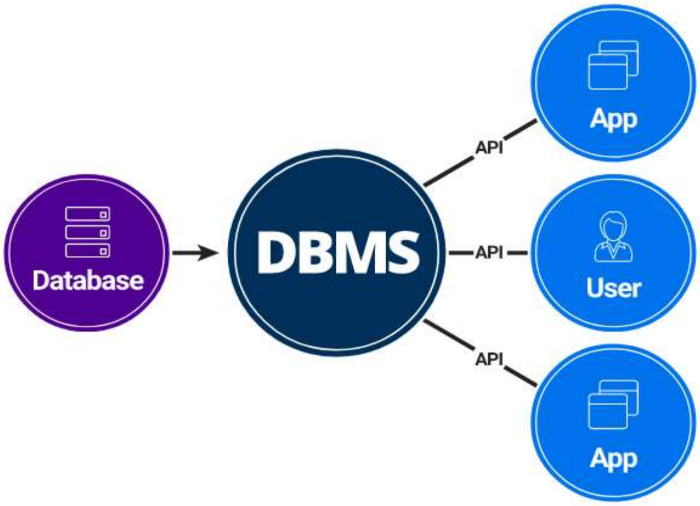


Fig 2.1 Components of a DBMS

Figure 1.1 shows the components of DBMS which describe the different parts that work together for creating, managing the database that forms a complete system named DBMS (database management system).

- **Users**: Users may be of any kind, such as data base administrators, system developers or database users.
- **Database application**: Database application may be Departmental, Personal, Organizational and /or Internal.
- **DBMS**: Software that allows users to create and manipulate database access.
- **Database**: Collection of logical data as a single unit.

CHAPTER-3

SYSTEM REQUIREMENTS

3.1 SOFTWARE REQUIREMENTS

- Technology Implemented: Apache Server, MySQL Server
- Language Used: Python
- Database: MySQL
- User Interface Design: HTML, CSS, Bootstrap.
- Web Browser: Google Chrome, Firefox
- PHPMYADMIN:
- Any text editor supporting Python compilation (VS Code)

3.2 HARDWARE REQUIREMENTS

- Processor - Intel 486/Pentium processor or better
- Processor Speed - 500 MHz or above
- HDD - 20GB (approx.)
- RAM - 512MB or above
- Storage Space - Approx. 2MB

CHAPTER 4

SYSTEM DESIGN

4.1 ER DIAGRAM

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties.

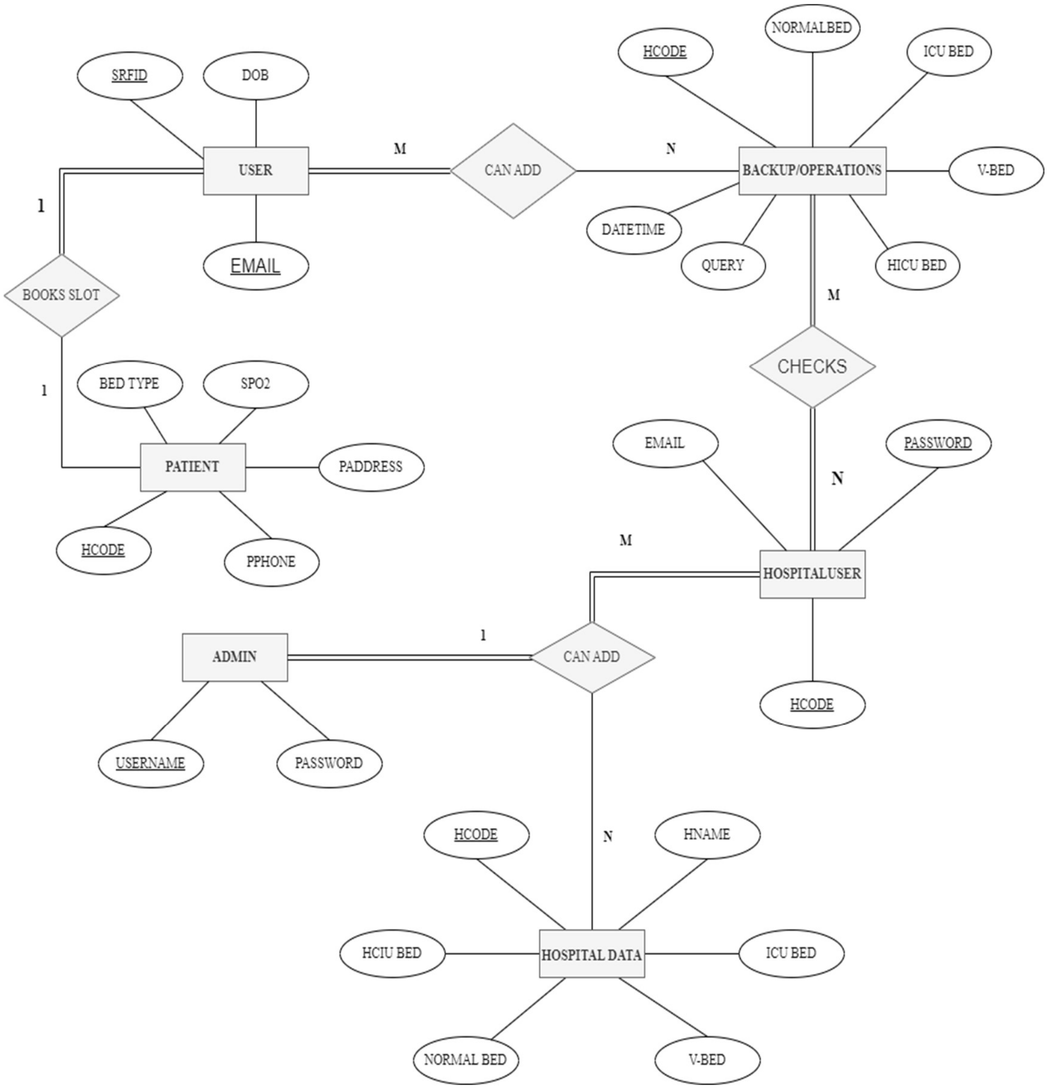


Figure 4.1 ER Diagram of Bed Slot Booking System

4.2 SCHEMA DIAGRAM

A schema diagram is a diagram which contains entities and the attributes that will define that schema. A schema diagram only shows us the database design. It does not show the actual data of the database. Schema can be a single table, or it can have more than one table which is related.

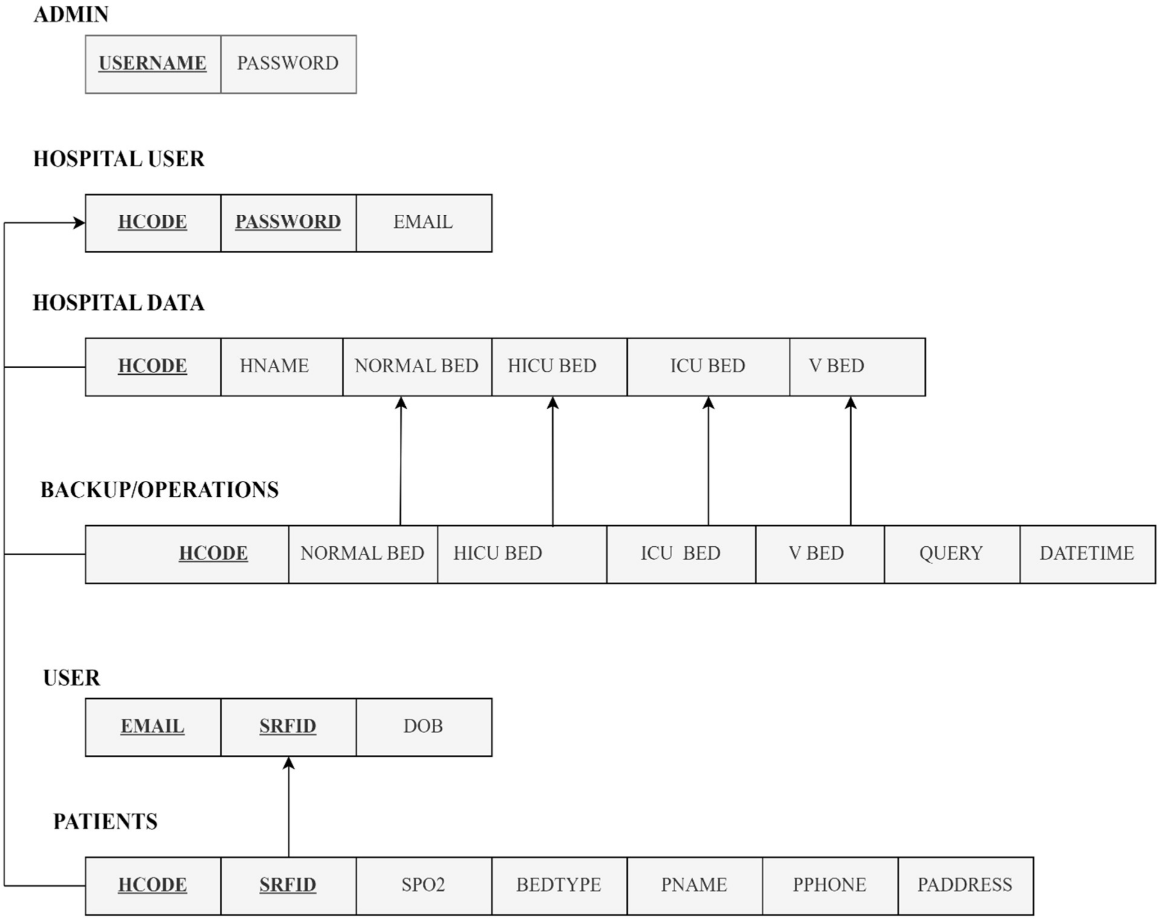


Figure 4.2 Schema Diagram of Bed Slot Booking System

4.3 CONTROL FLOW DIAGRAM

A Control Flow Graph (CFG) is the graphical representation of control flow or computation during the execution of programs or applications. Control flow graphs are mostly used in static analysis as well as compiler applications, as they can accurately represent the flow inside of a program unit.

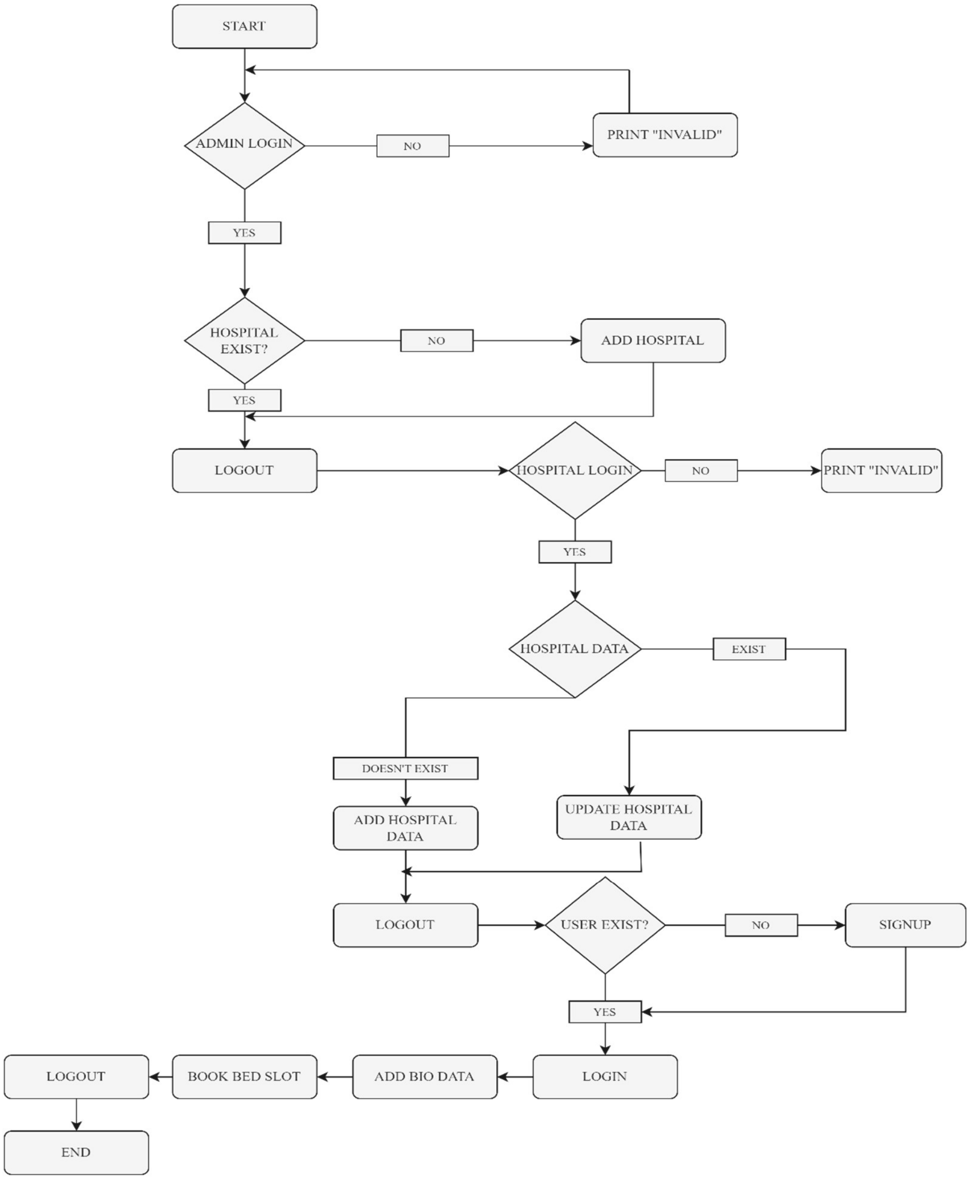


Figure 4.3 Working Flow Chart of Bed Slot Booking System

CHAPTER 5

IMPLEMENTATION

5.1 HTML, CSS AND BOOTSTRAP

HTML-CSS framework BOOTSTRAP, little bit of JAVASCRIPT is utilized to implement the frontend of this project.

HTML (Hyper Text Markup Language)

HTML is a syntax used to format a text document on the web. It is used to design web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. A markup language is used to define the text document within tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly.

CSS (Cascading Style Sheets)

CSS is a style sheet language used for describing the look and formatting of a document written in a markup language. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging web pages, user interfaces for web applications, and user interfaces for many mobile applications.

JAVA SCRIPT

JS is a dynamic computer programming language. It is most used as art of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser communicate asynchronously, and alter the document content that is displayed. Java Script is used to create popup windows displaying different alerts in the system.

BOOTSTRAP

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

5.2 PYTHON (FLASK FRAMEWORK)

Backend is server side of the website. It stores and arranges data, and makes sure everything on the client-side of the website works fine. It is the part of the website that you cannot see and interact with. It is the portion of software that does not come in direct contact with the users.

The Python programming language has powerful features for database programming. Python supports various databases like SQLite, MySQL, Oracle, Sybase, PostgreSQL, etc. Python also supports Data Definition Language (DDL), Data Manipulation Language (DML)

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

5.3 SQL (Structured Query Language)

SQL (Structured Query Language) 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 [SAM 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, e.g., with or without an index.

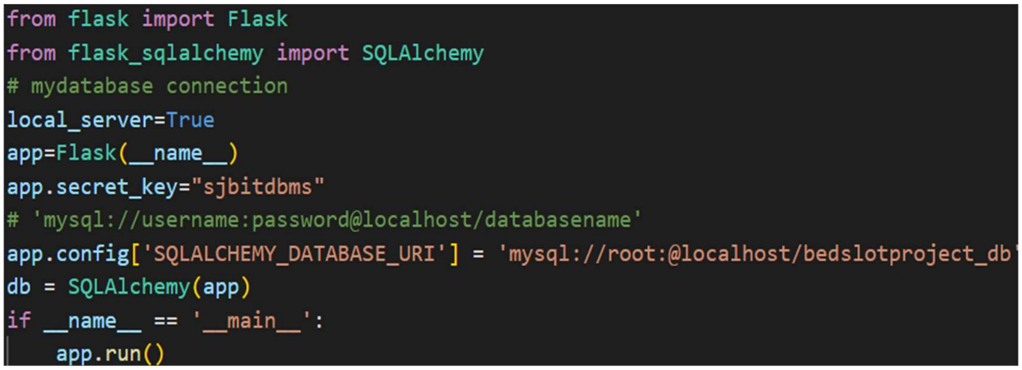
Originally based upon relational algebra and tuple relational calculus, SQL consists of a data definition language, data manipulation language, and data control language. The scope of SQL includes data insert, query, update and delete, schema creation and modification, and data access control. Although SQL is often described as, and to a great extent is, a declarative language (4GL), it also includes procedural elements.

5.4 DATABASE CONNECTIVITY

Flask SQLAlchemy with MySQL

MySQL is developed, distributed, and supported by Oracle Corporation. MySQL is a database system used on the web that runs on a server. MySQL is ideal for both small and large applications. It is very fast, reliable, and easy to use. It supports standard SQL. MySQL can be compiled on a number of platforms. Flask-SQLAlchemy is an extension for Flask that adds support for SQLAlchemy to your application.

This is a code snippet to show how python Flask -SQLAlchemy is used to connect to the local MySQL database using the localhost server.

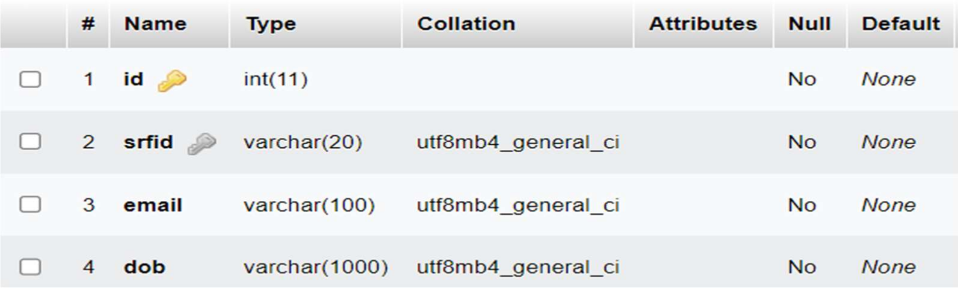


```
from flask import Flask
from flask_sqlalchemy import SQLAlchemy
# mydatabase connection
local_server=True
app=Flask(__name__)
app.secret_key="sjbitdbms"
# 'mysql://username:password@localhost/databasename'
app.config['SQLALCHEMY_DATABASE_URI'] = 'mysql://root:@localhost/bedslotproject_db'
db = SQLAlchemy(app)
if __name__ == '__main__':
    app.run()
```

Figure 5.1 SQL Alchemy Database Configuration with MySQL

5.5 TABLE DESCRIPTIONS

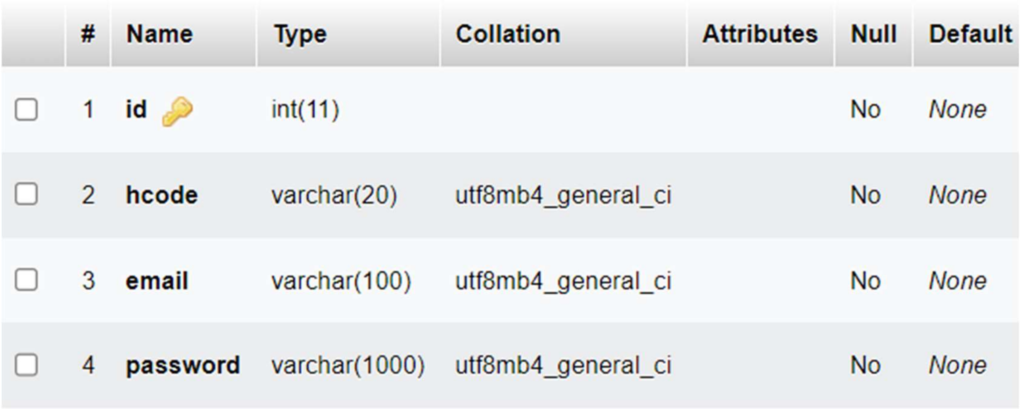
5.5.1 User Table: Table used to store the user information such as srfid, email, dob etc.



	#	Name	Type	Collation	Attributes	Null	Default
□	1	**id**	int(11)			No	*None*
□	2	**srfid**	varchar(20)	utf8mb4_general_ci		No	*None*
□	3	**email**	varchar(100)	utf8mb4_general_ci		No	*None*
□	4	**dob**	varchar(1000)	utf8mb4_general_ci		No	*None*

Table 5-1 User Table

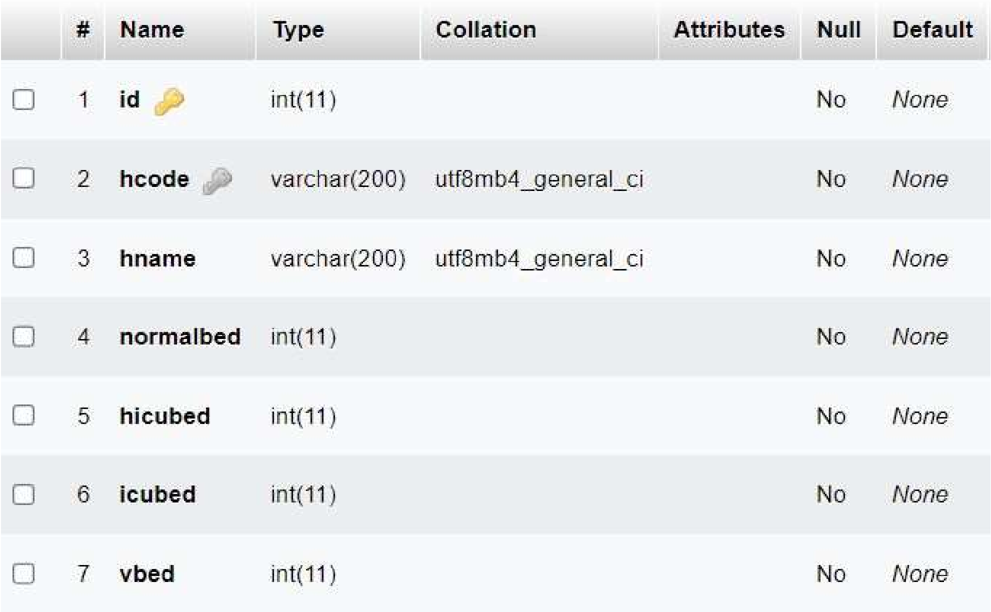
5.5.2 Hospital User Table: Table Used to store the hospital user information such as hcode, password etc.



	#	Name	Type	Collation	Attributes	Null	Default
□	1	**id** 🔑	int(11)			No	*None*
□	2	**hcode**	varchar(20)	utf8mb4_general_ci		No	*None*
□	3	**email**	varchar(100)	utf8mb4_general_ci		No	*None*
□	4	**password**	varchar(1000)	utf8mb4_general_ci		No	*None*

Table 5-2 HospitalUser Table

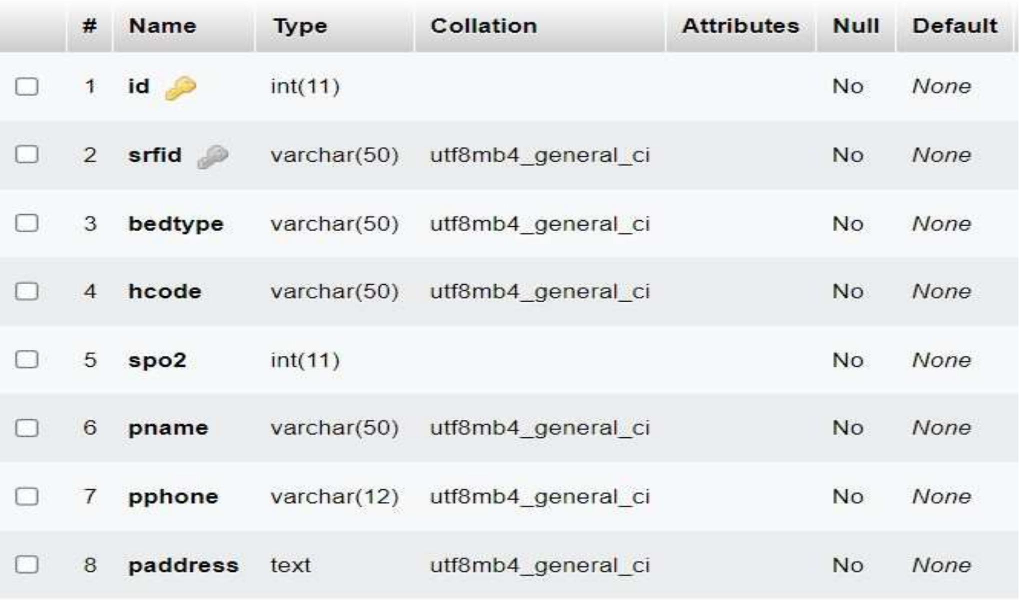
5.5.3 Hospital Data: Table used to store the hospital data which contains available type of beds and hospital name and hcode



	#	Name	Type	Collation	Attributes	Null	Default
□	1	**id** 🔑	int(11)			No	*None*
□	2	**hcode** 🔑	varchar(200)	utf8mb4_general_ci		No	*None*
□	3	**hname**	varchar(200)	utf8mb4_general_ci		No	*None*
□	4	**normalbed**	int(11)			No	*None*
□	5	**hicubed**	int(11)			No	*None*
□	6	**icubed**	int(11)			No	*None*
□	7	**vbed**	int(11)			No	*None*

Table 5-3 HospitalData Table

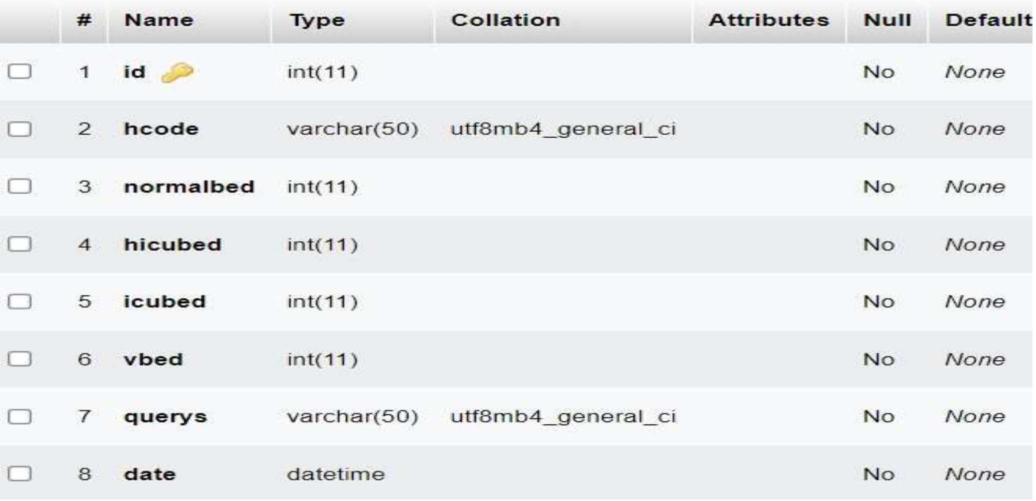
5.5.4 Bookingpatinet Table: Table used to store information about patients booking details such as type of bed and oxygen level , patients name,address etc.



	#	Name	Type	Collation	Attributes	Null	Default
□	1	**id** 🔑	int(11)			No	*None*
□	2	**srfid** 🔑	varchar(50)	utf8mb4_general_ci		No	*None*
□	3	**bedtype**	varchar(50)	utf8mb4_general_ci		No	*None*
□	4	**hcode**	varchar(50)	utf8mb4_general_ci		No	*None*
□	5	**spo2**	int(11)			No	*None*
□	6	**pname**	varchar(50)	utf8mb4_general_ci		No	*None*
□	7	**pphone**	varchar(12)	utf8mb4_general_ci		No	*None*
□	8	**paddress**	text	utf8mb4_general_ci		No	*None*

Table 5-4 PatientsBookingTable

5.5.5 Backup/Operations Table: Table used to store the oprations(insert,update,delete) occurred and datetime of when the operations occurred .



	#	Name	Type	Collation	Attributes	Null	Default
□	1	**id** 🔑	int(11)			No	*None*
□	2	**hcode**	varchar(50)	utf8mb4_general_ci		No	*None*
□	3	**normalbed**	int(11)			No	*None*
□	4	**hicubed**	int(11)			No	*None*
□	5	**icubed**	int(11)			No	*None*
□	6	**vbed**	int(11)			No	*None*
□	7	**querys**	varchar(50)	utf8mb4_general_ci		No	*None*
□	8	**date**	datetime			No	*None*

Table 5-5 Backup Table

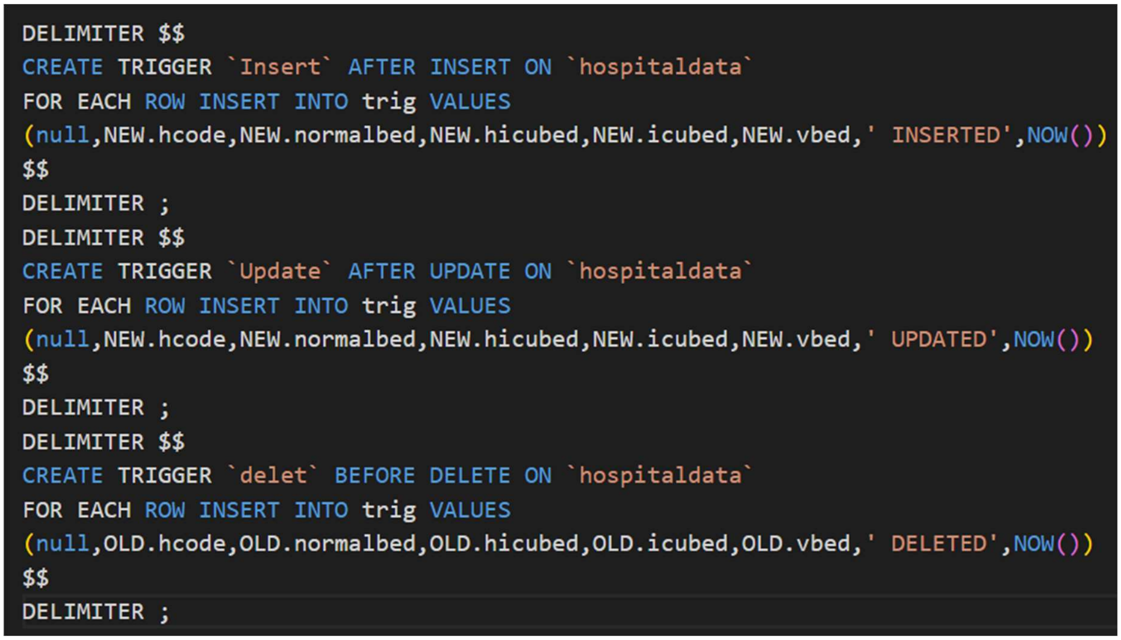
5.6 TRIGGERS

A trigger is a special type of stored procedure that automatically runs when an event occurs in the database server. DML triggers run when a user tries to modify data through a data manipulation language (DML) event.

5.6.1 Syntax:

```
CREATE TRIGGER trigger_name   trigger_time trigger_event

ON table_name FOR EACH ROW

BEGIN

        --variable declarations

--trigger code

END;
```

5.6.2 Triggers for Table Hospital Data



```
DELIMITER $$
CREATE TRIGGER `Insert` AFTER INSERT ON `hospitaldata`
FOR EACH ROW INSERT INTO trig VALUES
(null,NEW.hcode,NEW.normalbed,NEW.hicubed,NEW.icubed,NEW.vbed,' INSERTED',NOW())
$$
DELIMITER ;
DELIMITER $$
CREATE TRIGGER `Update` AFTER UPDATE ON `hospitaldata`
FOR EACH ROW INSERT INTO trig VALUES
(null,NEW.hcode,NEW.normalbed,NEW.hicubed,NEW.icubed,NEW.vbed,' UPDATED',NOW())
$$
DELIMITER ;
DELIMITER $$
CREATE TRIGGER `delet` BEFORE DELETE ON `hospitaldata`
FOR EACH ROW INSERT INTO trig VALUES
(null,OLD.hcode,OLD.normalbed,OLD.hicubed,OLD.icubed,OLD.vbed,' DELETED',NOW())
$$
DELIMITER ;
```

Figure 5.2 Trigger for table hospitaldata

5.7 STORED PROCEDURE

A stored procedure is a prepared SQL code that you can save, so the code can be reused over and over again. So, if you have an SQL query that you write over and over again, save it as a stored procedure, and then just call it to execute it.

5.7.1 Syntax

DELIMITER &&

CREATE PROCEDURE procedure_name [[IN | OUT | INOUT] parameter_name datatype [, parameter datatype])]

BEGIN

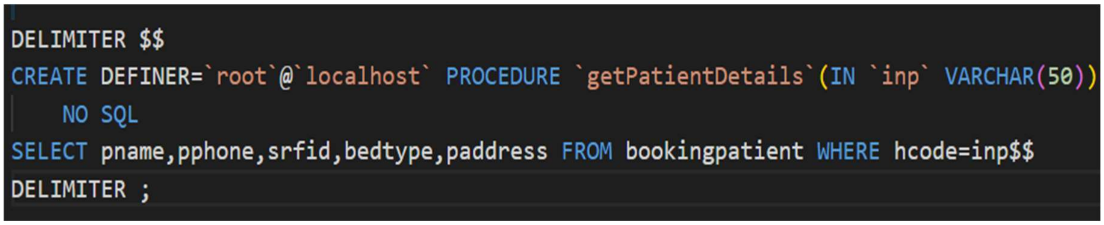
Declaration_section

Executable_section

END &&

DELIMITER;

5.7.2 Procedure for listing the patient details



```
DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getPatientDetails`(IN `inp` VARCHAR(50))
    NO SQL
SELECT pname,pphone,srfid,bedtype,paddress FROM bookingpatient WHERE hcode=inp$$
DELIMITER ;
```

Figure 5.3 Stored procedure

CHAPTER 6

TESTING AND RESULTS

6.1 TESTING

6.1.1 Testing

Testing is evaluation of the software against requirements gathered from users and system specifications. Testing identifies important defects, flaws, or an error in the application code that must be fixed. It also assesses the feature of a system. Testing assesses the quality of the product.

6.1.2 Unit Testing

Unit testing refers to the testing certain functions and areas of the code. It gives the ability to verify that all the functions work as expected.

6.1.3 Integration Testing

Integration testing is basically a logical extension of unit testing. In simple words, two tested units are combined into a component and the interface between them is tested. It identifies problems that occur when different units are combined. The different modules of this project have undergone integration testing while being merged.

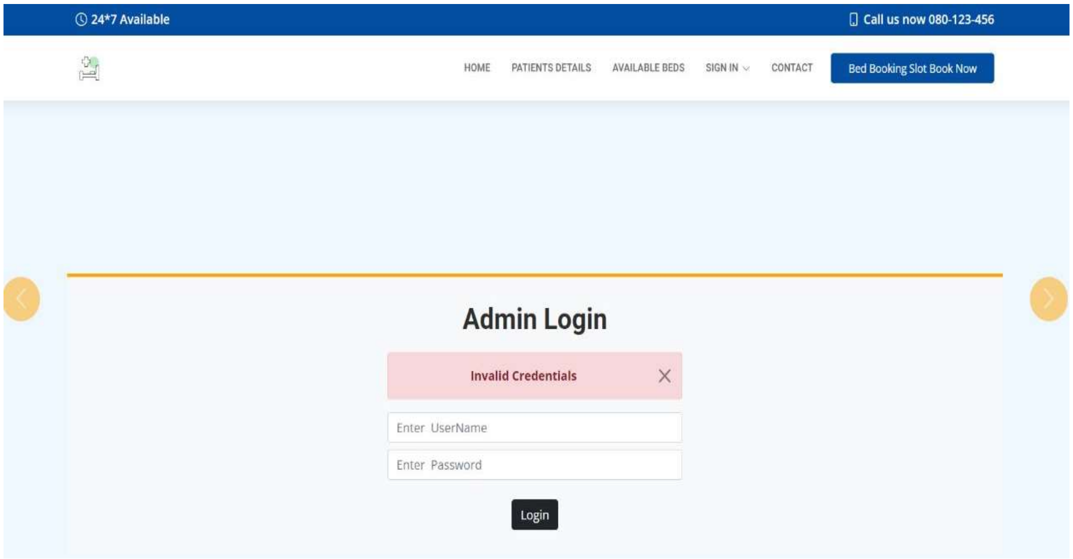
6.1.4 System Testing

System testing tests the behavior of whole system as defined by the scope of the development project. It might include tests based on risks as well as requirement specifications, business process, use cases or other high-level descriptions of system behavior, interactions with the operating systems and system resources. It is most often the final test performed to verify that the system meets the specification and its objectives. System testing has been performed at the completion of each feature and is still taking place to make improvements on the existing system.

Test Case Id	**Test Case Name**	**Test Case Description**	**Input**	**Expected Result**	**Actual Result**	**Result**
TC 01	User login	Validate User name, Password And Captcha	Enter Valid User name, Password And Captcha	User login Should be successful	User login successful	PASS
TC 02	User login	Validate User name and Password And Captcha	Enter Invalid User name And Password And Captcha	“Invalid User name/ Password” Error Message Should Be displayed	“Invalid User name/ Password” Error Message is displayed	PASS
TC 03	User login	Validate User name and Password	To check if any Fields are Left blank	“Fields are empty” Error message should be displayed	“Fields are empty” Error message is displayed	PASS

Table 6-1 Test Cases

6.2 RESULTS

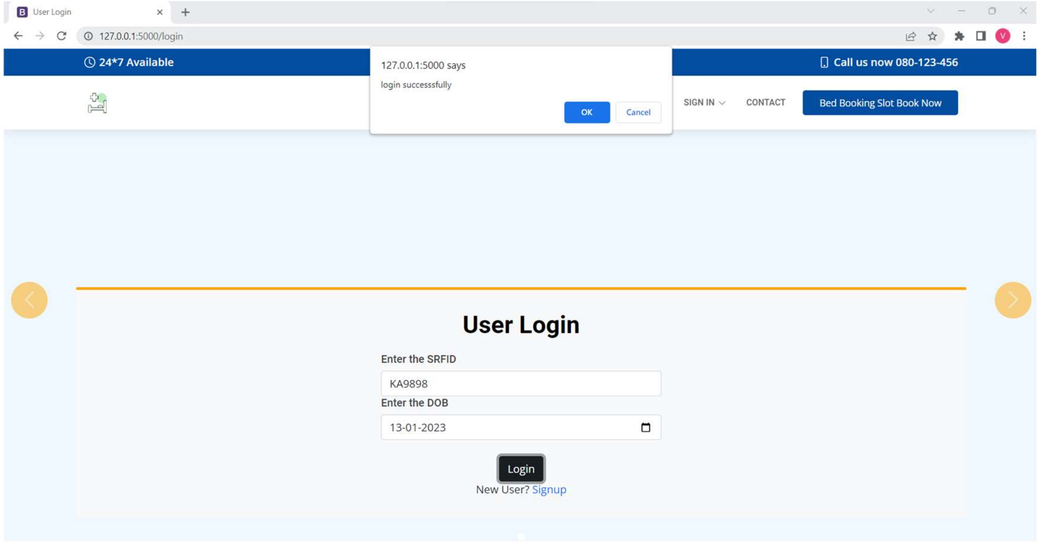


Admin Login

Invalid Credentials

Figure 6.1 Testing for Invalid Username and password

- The above Figure 6.1 shows the screenshot of testcases for invalid credentials in admin login.



127.0.0.1:5000 says

login successsfully

User Login

Enter the SRFID

KA9898

Enter the DOB

13-01-2023

Figure 6.2 Testing for valid credentials

- The above figure 6.2 shows screenshot of valid testcase it says login success when credentials are correct.

6.3 SNAPSHOTS

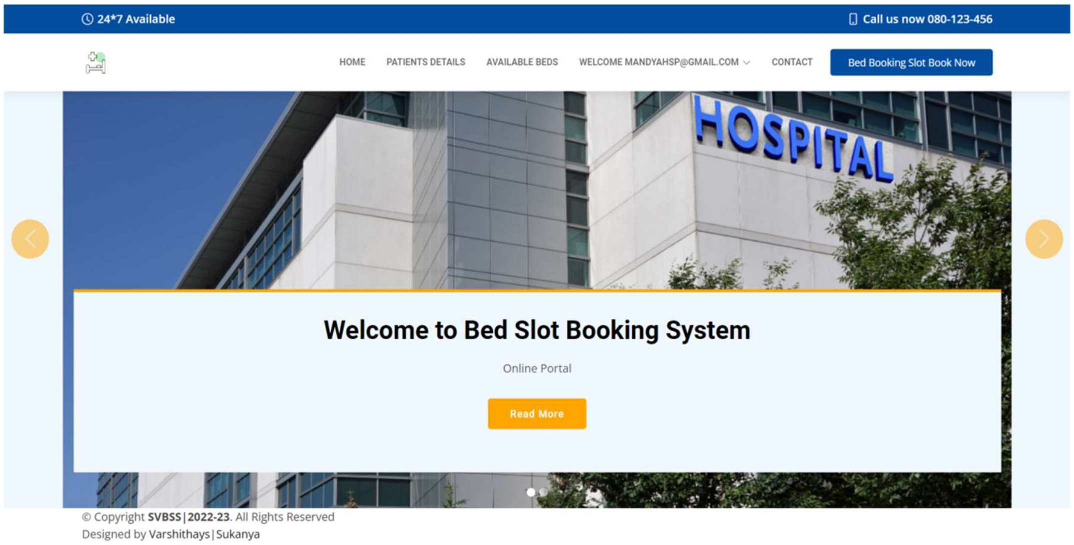

24*7 Available
Call us now 080-123-456
HOME
PATIENTS DETAILS
AVAILABLE BEDS
WELCOME MANDYAHSP@GMAIL.COM
CONTACT
Bed Booking Slot Book Now
HOSPITAL
Welcome to Bed Slot Booking System
Online Portal
Read More
© Copyright SVBSS|2022-23. All Rights Reserved
Designed by Varshithays|Sukanya

Figure 6.3 Home Page of Bed Slot Booking system.

- The above figure 5.3 shows the screenshot of the Welcome Page of our Project.
- It has Admin, User and Hospital Login

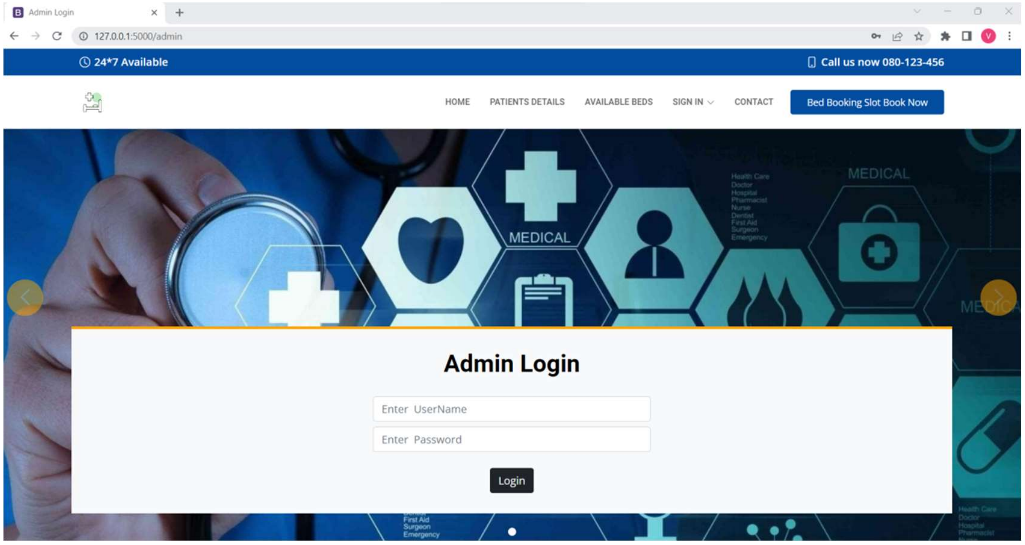
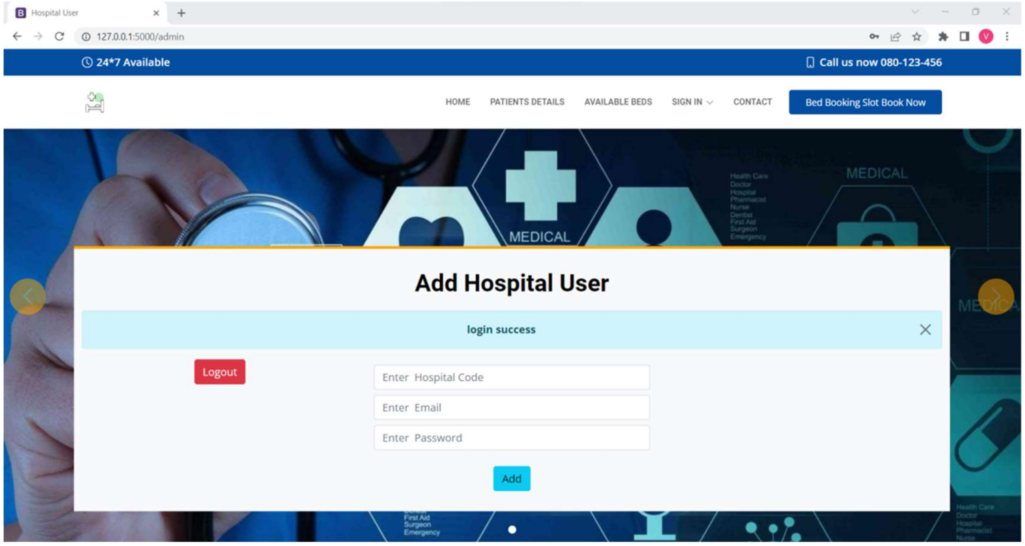


Figure 6.4 Admin Login Page of Bed Slot Booking system

- The above Figure 6.4 shows screenshot admin login page.



Add Hospital User

login success

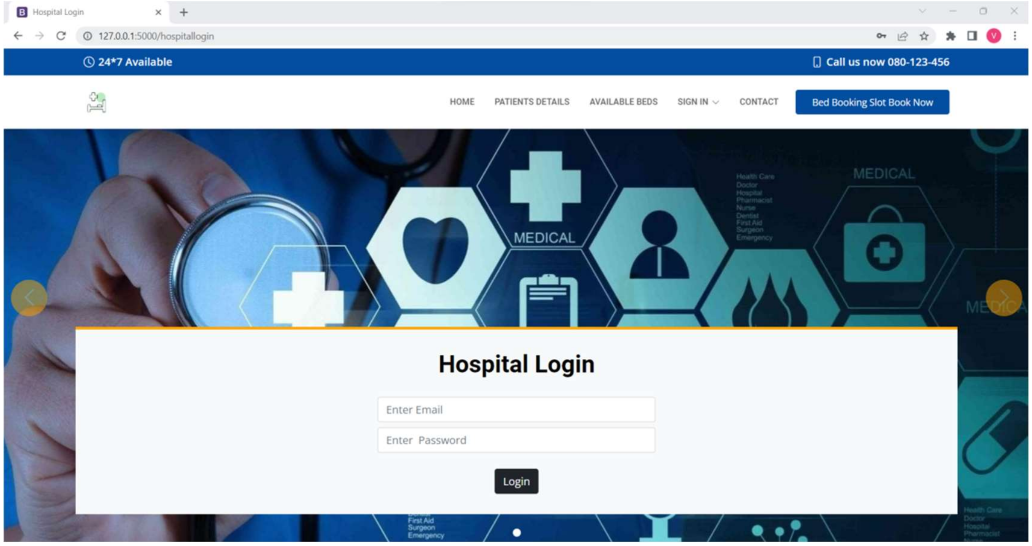
Enter Hospital Code

Enter Email

Enter Password

Figure 6.5 Add Hospital User of Bed Slot Booking system

- The above Figure 6.5 shows the screenshot of add hospital user details page, it will ask for hospital code and email, password.



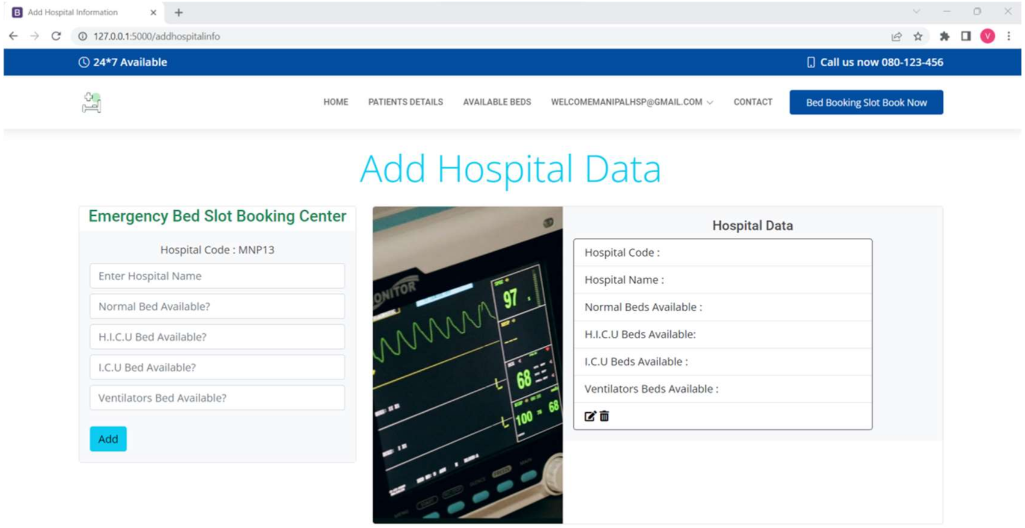
Hospital Login

Enter Email

Enter Password

Figure 6.6 Hospital Login of Bed Slot Booking system

- The above Figure 6.6 shows the screenshot of Hospital Login page.



Add Hospital Data

Emergency Bed Slot Booking Center

Hospital Code : MNP13

Enter Hospital Name

Normal Bed Available?

H.I.C.U Bed Available?

I.C.U Bed Available?

Ventilators Bed Available?

Hospital Data

Hospital Code :

Hospital Name :

Normal Beds Available :

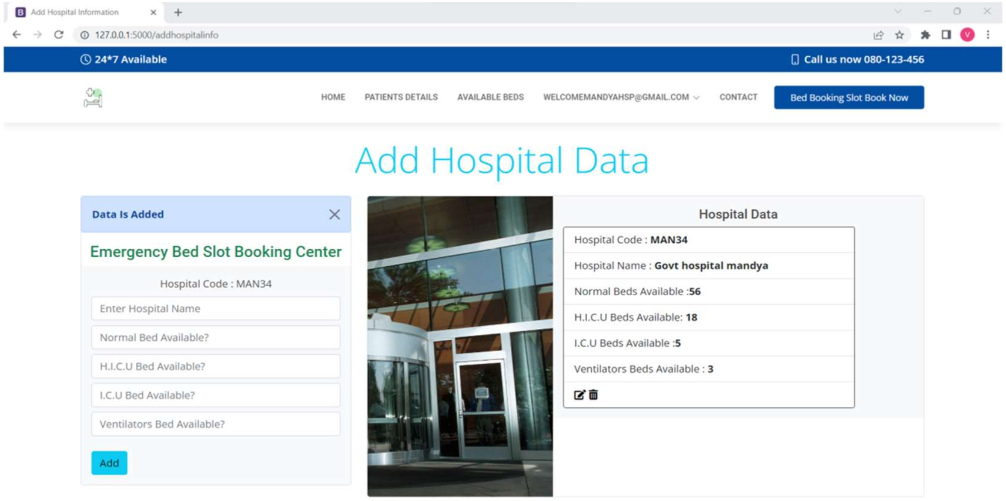
H.I.C.U Beds Available:

I.C.U Beds Available :

Ventilators Beds Available :

Figure 6.7 Add Hospital Data Page of Bed Slot Booking system

- The above Figure 6.7 shows the screenshot of Add Hospital Data page. In this page the admin can add the hospital information.



Add Hospital Data

Data Is Added

Emergency Bed Slot Booking Center

Hospital Code : MAN34

Enter Hospital Name

Normal Bed Available?

H.I.C.U Bed Available?

I.C.U Bed Available?

Ventilators Bed Available?

Hospital Data

Hospital Code : **MAN34**

Hospital Name : **Govt hospital mandya**

Normal Beds Available :**56**

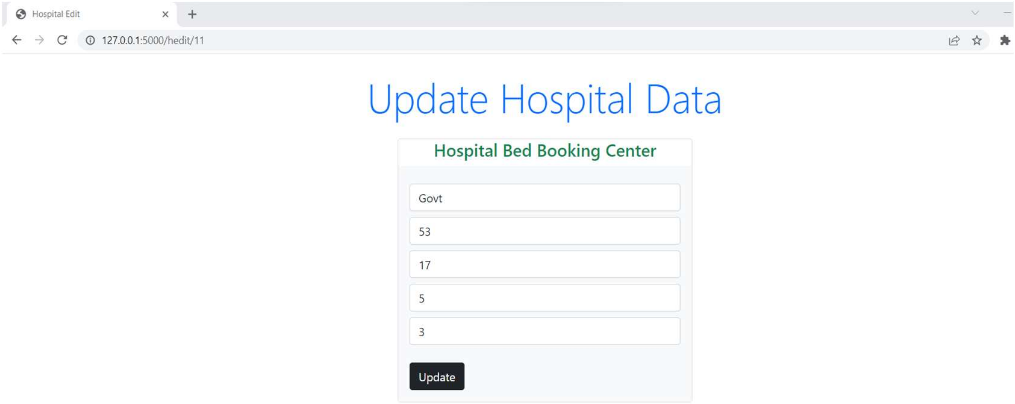
H.I.C.U Beds Available: **18**

I.C.U Beds Available :**5**

Ventilators Beds Available : **3**

Figure 6.8 Display of added Hospital data of Bed Slot Booking system.

- The above Figure 6.8 shows the screenshot of added hospital information such as hospital name, how many beds available etc.



Update Hospital Data

Hospital Bed Booking Center

Govt

53

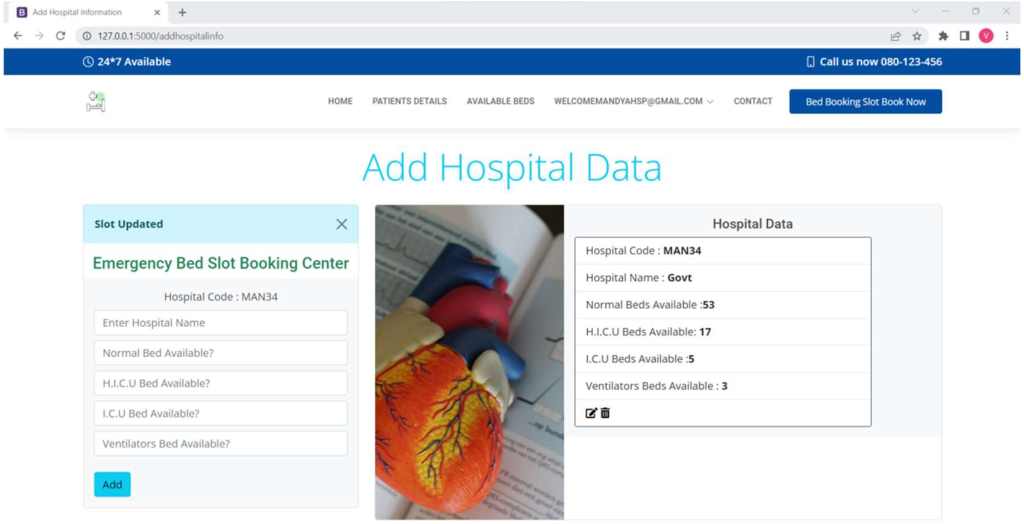
17

5

3

Figure 6.9 Update Hospital Data Page of Bed Slot Booking system

- The above Figure 6.9 shows the screenshot of Update hospital data. As per the beds available in hospital the hospital user can edit the information.



Add Hospital Data

Slot Updated

Emergency Bed Slot Booking Center

Hospital Code : MAN34

Enter Hospital Name

Normal Bed Available?

H.I.C.U Bed Available?

I.C.U Bed Available?

Ventilators Bed Available?

Hospital Data

Hospital Code : **MAN34**

Hospital Name : **Govt**

Normal Beds Available :**53**

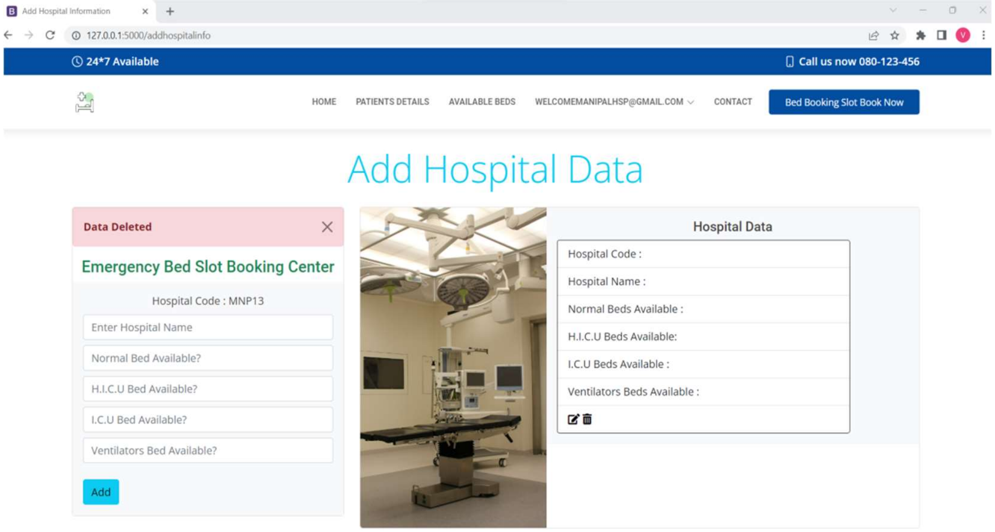
H.I.C.U Beds Available: **17**

I.C.U Beds Available :**5**

Ventilators Beds Available : **3**

Figure 6.10 Updated data of Bed Slot Booking system

- The above Figure 6.10 shows the screenshot of hospital data displayed which is after update the information.



Add Hospital Data

Data Deleted

Emergency Bed Slot Booking Center

Hospital Code : MNP13

Enter Hospital Name

Normal Bed Available?

H.I.C.U Bed Available?

I.C.U Bed Available?

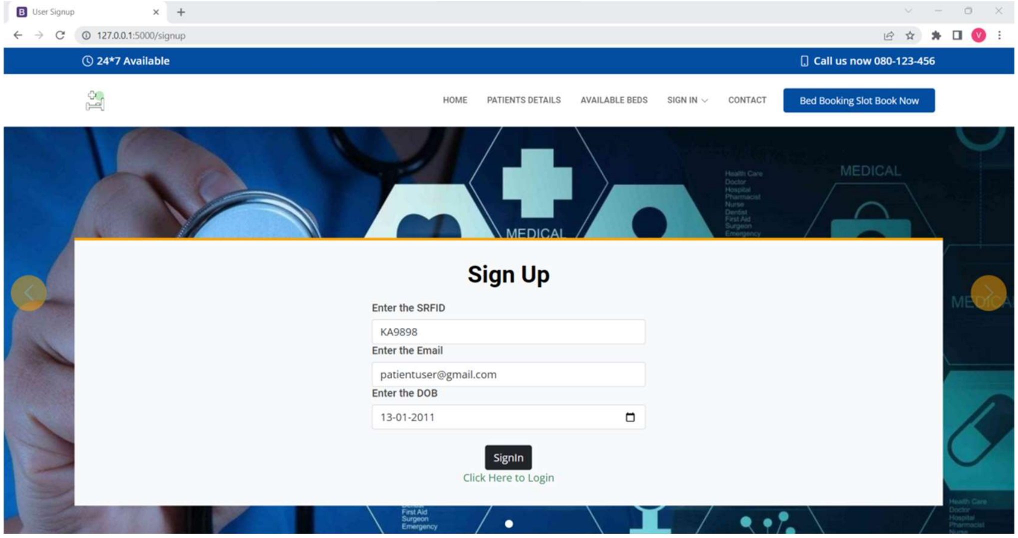
Ventilators Bed Available?

Hospital Data

Hospital Code :
Hospital Name :
Normal Beds Available :
H.I.C.U Beds Available:
I.C.U Beds Available :
Ventilators Beds Available :

Figure 6.11 Delete Hospital data of Bed Slot Booking system.

- The above Figure 6.11 shows the screenshot of hospital data deleted, hospital user can delete the hospital information.



Sign Up

Enter the SRFID

KA9898

Enter the Email

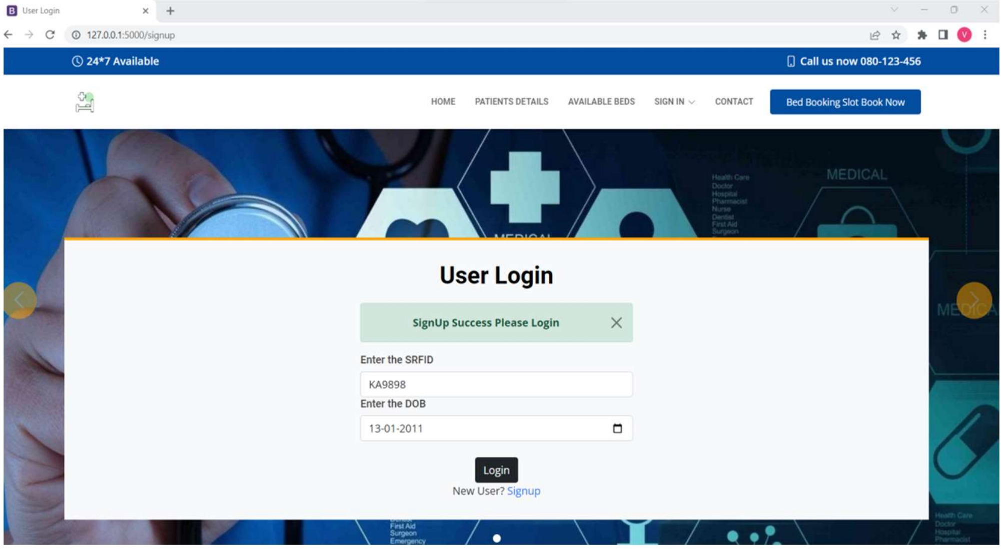
patientuser@gmail.com

Enter the DOB

13-01-2011

Figure 6.11 User Sign Up Page of Bed Slot Booking system

- The above Figure 6.11 shows the screenshot of Sign-Up page. In this page the patients/users can sign up before to login.



User Login

SignUp Success Please Login

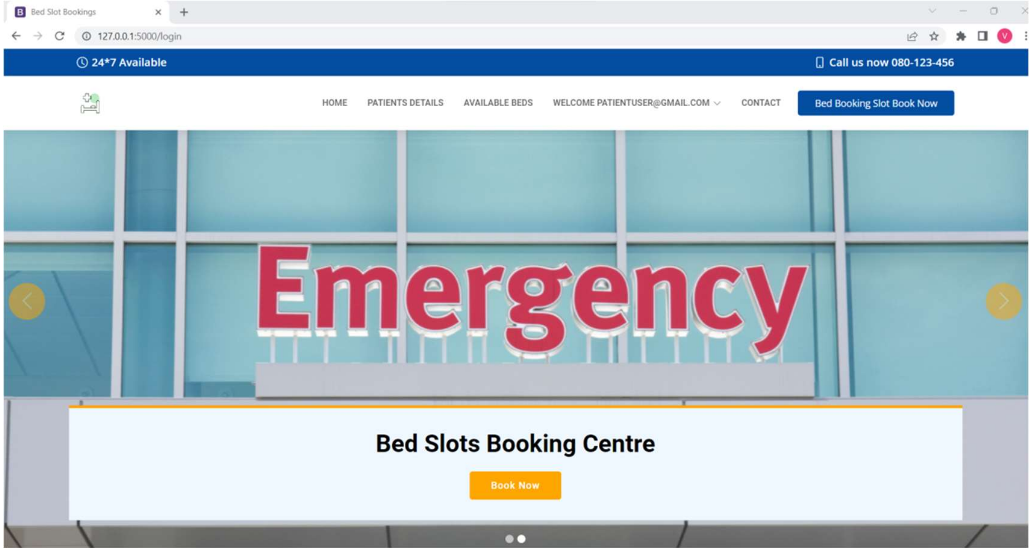
Enter the SRFID: KA9898

Enter the DOB: 13-01-2011

New User? Signup

Figure 6.12 User Login of Bed Slot Booking system

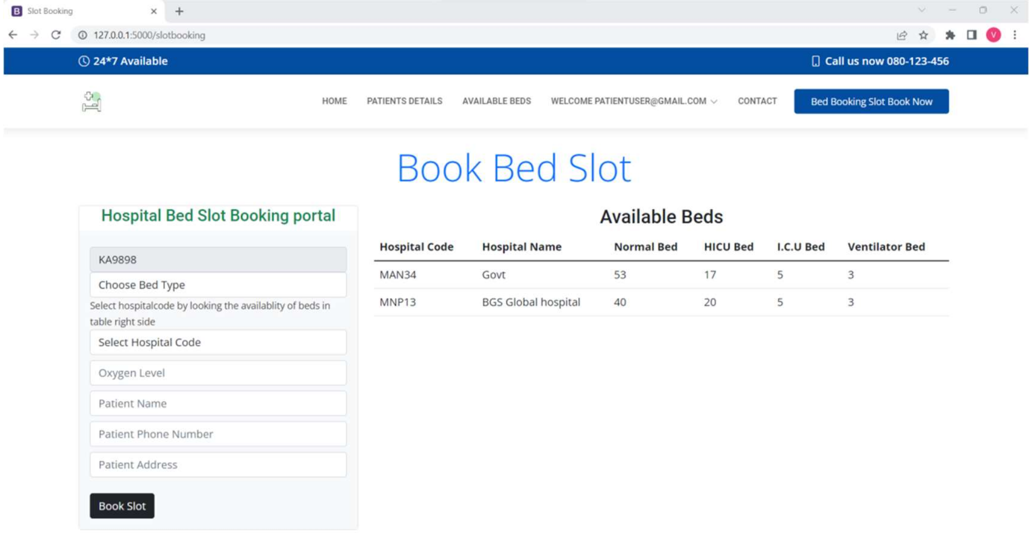
- The above Figure 6.12 shows screenshot of user login page, here user can login with login credentials such as unique srfid and dob.



Bed Slots Booking Centre

Figure 6.13 Home -User Page of Bed Slot Booking system

- The above Figure 6.13 shows screenshot of home page after the user login successfully then it shows this page here user can book the bed slots available.

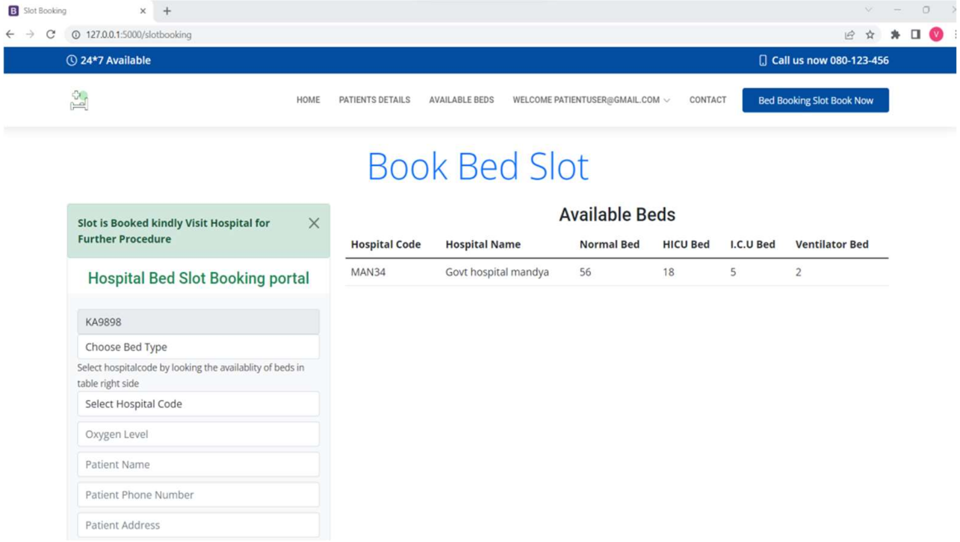


KA9898

Hospital Code	Hospital Name	Normal Bed	HICU Bed	I.C.U Bed	Ventilator Bed
MAN34	Govt	53	17	5	3
MNP13	BGS Global hospital	40	20	5	3

Figure 6.14 Slot Booking page of Bed Slot Booking system

- The above Figure 6.14 shows screenshot of patient bed slot booking page here user can book beds according to the availability of beds in hospitals.



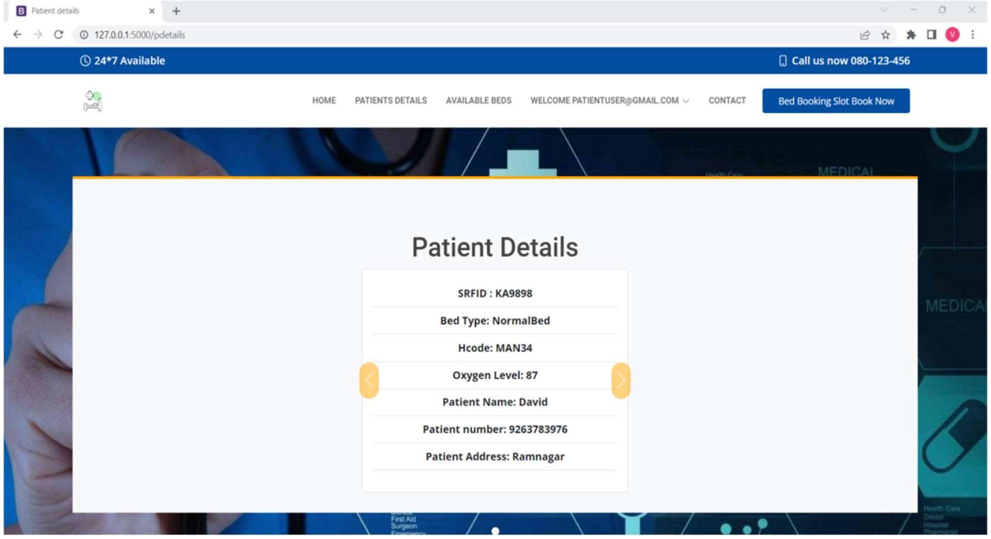
Slot is Booked kindly Visit Hospital for Further Procedure

KA9898

Hospital Code	Hospital Name	Normal Bed	HICU Bed	I.C.U Bed	Ventilator Bed
MAN34	Govt hospital mandya	56	18	5	2

Figure 6.15 Available Beds page of Bed Slot Booking system

- The above Figure 6.15 shows screenshot of available beds after patient booked slot.



Patient Details

SRFID : KA9898

Bed Type: NormalBed

Hcode: MAN34

Oxygen Level: 87

Patient Name: David

Patient number: 9263783976

Patient Address: Ramnagar

Figure 6.15 Patient Details page of Bed Slot Booking system

- The above Figure 6.15 shows screenshot of patient details page it displays the details of patient who booked the slot.