VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI -590 018, KARNATAKA

DBMS MINI PROJECT REPORT

ON

“Hospital Bed Slot Booking System”

Submitted in the partial fulfillment of requirements for the

5th SEM DBMS MINI PROJECT (21CSL55)

IN

COMPUTER SCIENCE AND ENGINEERING

PROJECT ASSOCIATES

Anushree B Kuppast	**4BD21CS020**
Komal P Jadimath	**4BD21CS062**

PROJECT GUIDES

Dr. Gururaj T Ph.D.,	**Prof. Arjun H M.Tech,**
Associate Professor	**Assistant Professor**

2023-2024

Bapuji Institute of Engineering and Technology
Department of Computer Science and Engineering
Davanagere-577004

Bapuji Institute of Engineering and Technology
Davanagere – 577004

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that **Anushree B Kuppast and Komal P Jadimath** bearing USN **4BD20CS000 and 4BD20CS000** respectively of Computer Science and Engineering department have satisfactorily submitted the Mini Project report entitled **"Hospital Bed Slot Booking System"** for **5th SEM DBMS MINI PROJECT (21CSL55).** The project report has been approved as it satisfies the academic requirements for the year 2023-24.

Dr. Gururaj T Ph.D.,
Associate Professor
Guide

Prof. Arjun H M.Tech.,
Assistant Professor
Co-Guide

Dr. Nirmala C R Ph.D.,
Head of Department

Signature of Examiners:

Date: **1.**___________________________

Place: Davanagere **2.**___________________________

ACKNOWLEDGEMENT

Salutations to our beloved and highly esteemed institute, **“BAPUJI INSTITUTE OF ENGINEERING AND TECHNOLOGY”** for having well-qualified staff and labs furnished with the necessary equipment.

We express our sincere thanks to our resourceful guides **Dr. Gururaj T**, Associate Professor, Department of Computer Science and Engineering, B.I.E.T., Davanagere, and **Prof. Arjun H**, Assistant Professor, Department of Computer Science and Engineering, BI.E.T., Davanagere, who helped us in every aspect of our project. We are indebted to her discussions about the technical aspects and suggestions pertaining to our project.

We are grateful to **Dr. Nirmala C R**, Professor and H.O.D, Department of Computer Science and Engineering, B.I.E.T., Davanagere, for endeavoring encouragement, facilities, and extended support.

We also express our wholehearted gratitude to our respected Principal, **Dr. H B Aravind** for his moral support and encouragement.

We would like to extend our gratitude to all staff of the **Department of Computer Science and Engineering** for the help and support rendered to us. We have benefited a lot from the feedback, and suggestions given by them.

We would like to extend our gratitude to all our family members and friends especially for their advice and moral support.

Anushree B Kuppast (4BD21CS020)
Komal P Jadimath (4BD21CS062)

Bapuji Educational Association (Regd.)
Bapuji Institute of Engineering and Technology, Davangere-577004

Vision and Mission of the Institute

Vision

―To be a centre of excellence recognized nationally internationally, in distinctive areas of engineering education and research, based on a culture of innovation and invention.‖

Mission

―BIET contributes to the growth and development of its students by imparting a broadbased engineering education and empowering them to be successful in their chosen field by inculcating in them positive approach, leadership qualities and ethical values.‖

Vision and Mission of the Computer Science and Engineering Department

Vision

―To be a centre-of-excellence by imbibing state-of-the-art technology in the field of Computer Science and Engineering, thereby enabling students to excel professionally and be ethical.‖

Mission

1.	Adapting best teaching and learning techniques that cultivates Questioning and Reasoning culture among the students.
2.	Creating collaborative learning environment that ignites the critical thinking in students and leading to the innovation.
3.	Establishing Industry Institute relationship to bridge skill gap and make them industry ready and relevant.
4.	Mentoring students to be socially responsible by inculcating ethical and moral values.

Program Educational Objectives (PEOs):

PEO1	To apply skills acquired in the discipline of computer science and engineering for solving Societal and industrial problems with apt technology intervention.
PEO2	To continue their carrier ion industry /academia or pursue higher studies and research.

PEO3	To become successful entrepreneurs, innovators to design and develop software
	products and services that meets societal, technical and business challenges.
PEO4	To work in the diversified environment by acquiring leadership qualities with effective communication skills accompanied by professional and ethical values.

Program Specific Outcomes (PSOs):

PSO1	Analyse and develop solutions for problems that are complex in nature but applying the knowledge acquired from the core subjects of this program.
PSO2	To develop secure, scalable, resilient and distributed applications for industry and societal Requirements.
PSO3	To learn and apply the concepts and contract of emerging technologies like artificial intelligence, machine learning, deep learning, big-data analytics, IOT, cloud computing etc for any real time problems.

Course Learning Objectives:

- Foundation knowledge in database concepts, technology and practice to groom students into well-informed database application developers.
- Strong practice in SQL programming through a variety of database problems.
- Develop database applications using front-end tools and back-end DBMS.

Course Outcomes:

CO1: Create, Update and query on the database.

CO2: Demonstrate the working of different concepts of DBMS

CO3: Implement, analyze and evaluate the project developed for an application.

ABSTRACT

The Hospital Bed Slot Booking System is a database management system (DBMS) project designed to facilitate efficient management and booking of hospital beds. With the ongoing challenges in healthcare, especially during emergencies like the COVID-19 pandemic, efficient allocation and tracking of hospital beds become critical. This system aims to provide a solution to manage bed availability, patient admissions, and discharge processes effectively. The system will maintain a database of all beds in the hospital, including details like bed type (ICU, general, etc.), availability status, and location within the hospital. The primary objective of this DBMS project is to create a robust system that allows hospital staff to manage bed availability, assign beds to patients, track patient admissions, and facilitate timely discharge when required. The system should also provide reporting capabilities for analyzing bed utilization, occupancy rates, and overall hospital efficiency.

CONTENT

CHAPTER 1

1.1 INTRODUCTION

The project hospital bed slot allotment system includes registration of patients, storing their details into the system. The software has the ability to give a unique id to each patient and stores the details of every patient and their vaccination details automatically. Users can search availability of the bed and the details of a patient using the id. The hospital bed slot allotment system can be entered by using a username and password. It is accessible either by an administrator or receptionist. Only they can and data into the database. The data can be retrieved easily. The interface is very user friendly. The data is very well protected for personal use and makes the data processing very fast. Hospital bed slot allotment system is a very powerful, flexible and easy to use and is designed for multi purposes. It is an integrated end to end Hospital bed slot allotment system that provides relevant information and decision making for the user in the interface for a seamless flow. Hospital bed slot allotment system is a software product suite designed to improve the time required to allot a new patient a bed as immediately as possible. This system enables you to develop your organization and improve its effectiveness and quality of work. Managing the key processes efficiently is critical to the success of the entire system.

1.2 DBMS (DATABASE MANAGEMENT SYSTEM)

Database is a collection of related data and data is a collection of facts and figures that can be processed to produce information. Mostly data represents recordable facts. Data aids in producing information, which is based on facts. For example, if we have data about marks obtained by all students, we can then conclude about toppers and average marks.

A database management system (DBMS) is a software package designed to define, manipulate, retrieve and manage data in a database. A DBMS generally manipulates the data itself, the data format, field names, record structure and file structure. It also defines rules to validate and manipulate this data.

A DBMS relieves users of framing programs for data maintenance. Fourth- generation query languages, such as SQL, are used along with the DBMS package to interact with a database.
Some other DBMS examples include:

- MySQL
- SQL Server
- Oracle
- D BASE

1.3 PYTHON FRAMEWORK

Python frameworks automate the implementation of several tasks and give developers a structure for application development. Each framework comes with its own collection of modules or packages that significantly reduce development time. A Python framework can either be fullstack, micro, or asynchronous

1.4 BootStrap

Bootstrap is a free, open source front-end development framework for the creation of websites and web apps. Designed to enable responsive development of mobile-first websites, Bootstrap provides a collection of syntax for template designs.

1.5PROBLEM STATEMENT

Hospital bed slot booking involves creating a system or application to facilitate the efficient management of hospital beds. This includes features such as real-time monitoring of bed availability, allowing patients or healthcare providers to book or reserve beds based on urgency and medical condition, managing cancellations and rescheduling, ensuring fair allocation of resources, and optimizing bed utilization to meet the demands of patients requiring different levels of care.

Develop a digital platform to streamline the allocation of hospital beds, allowing patients or healthcare personnel to book beds online based on availability, urgency, and medical condition, while ensuring equitable distribution of resources and efficient management of cancellations and rescheduling.

Design an integrated system for hospital bed management, enabling real-time tracking of bed availability across multiple healthcare facilities, empowering users to reserve beds according to priority and severity of illness, and implementing protocols for fair allocation and utilization of beds amidst fluctuating demand.

Create a user-friendly application for hospital bed booking, offering patients or caregivers the ability to search for and secure available beds in nearby hospitals, prioritize bookings based on medical urgency, and enable seamless communication between healthcare providers and patients for efficient scheduling and updates.

1.5 OBJECTIVES

- To develop a rental management system that allows the user to view customers‟ data as well as record.
- To develop a system that allows the users to add, edit, search and delete data from the database.
- To study and analyse the requirement specifications of the rental house management system.

CHAPTER 2 REQUIREMENT SPECIFICATION

2.1 HARDWARE REQUIREMENTS

The Hardware required for the development of this project is:

- Processor : Intel Core i5
- System type : 64-bit operating system
- RAM : 8 GB RAM
- Version : Windows 11 Pro
- Hard Disk : 20GB(approx.)

2.2 SOFTWARE REQUIREMENTS

The Software required for the development of this project is:

- Technology Implemented : Apache Server, MySQL Server
- Language Used : Python
- Database : My SQL
- User Interface Design : HTML, CSS, Bootstrap.
- Web Browser : Google Chrome, Firefox
- Software : XAMPP Version: 7.1.10

.

CHAPTER 3

3.1 ER Diagram And description

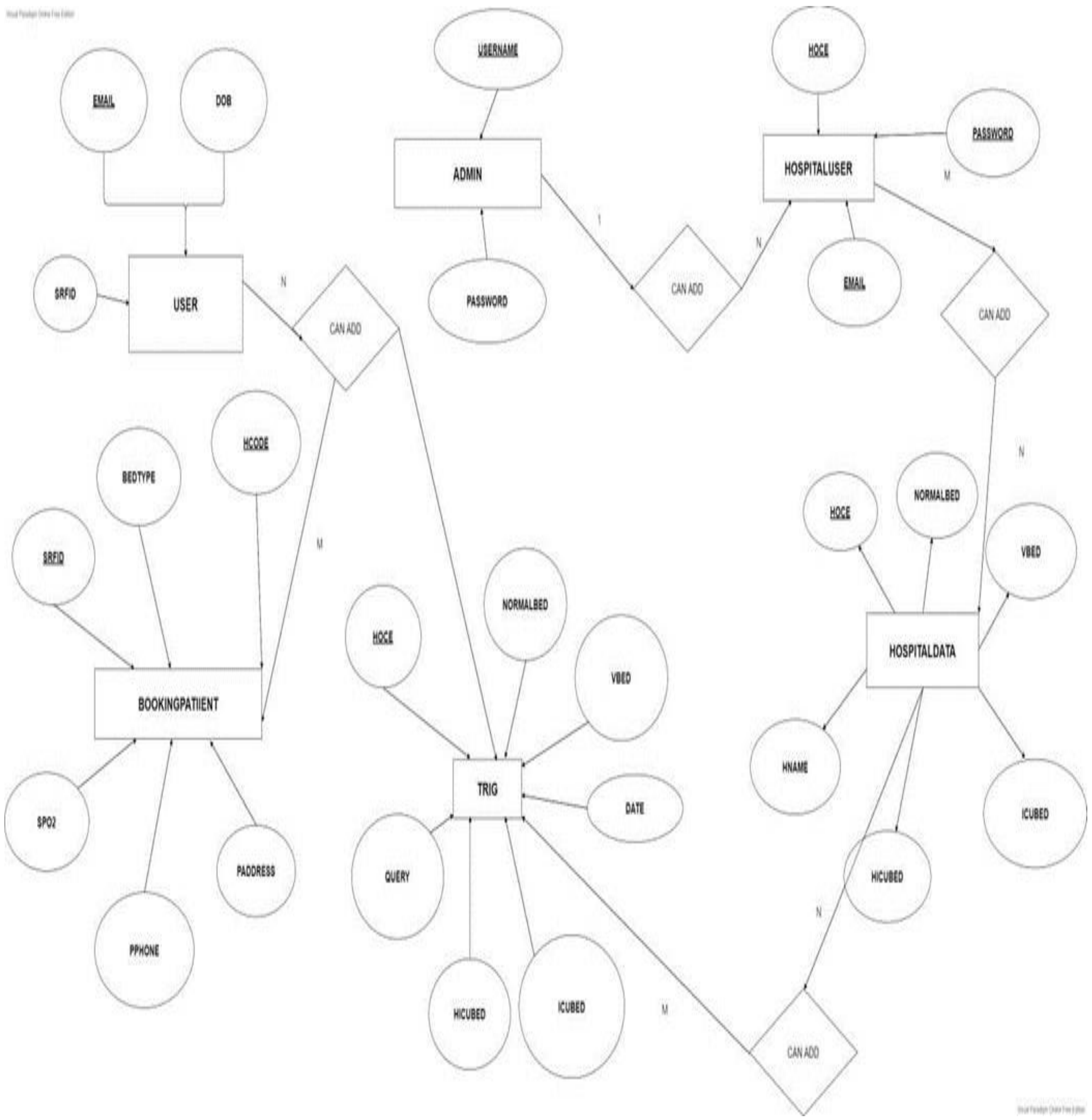


Fig 2.1: ER Diagram of Hospital

3.2 DESCRIPTION

The ER Model figure shows a conceptual view of the database. It works around realworld entities and the associations among them. At view level, the ER model is considered a good option for designing databases. So, let"s see each entity

ADMIN TABLE

This entity stores the information about the admin who registers and logs in using his username and password.

USER TABLE

This entity stores the information about the user who registers. Attributes are id, srfid, email, dob.

HOSPITAL USER TABLE

This entity stores the information about the hospital user who can add the hospital data .Attributes are email, hcode, passward.

HOSPITAL DATA TABLE

This entity stores the information about the hospital and available beds .Attributes are hcode, hicubed, hname, icubed, normalbed, vbed.

BOOKING PATIENT TABLE

This entity stores the information about the patient details. Attributes are id, srfid, bedtype, hcode, pname, pphone, paddress.

SEVEN STEPS FOR ER TO SCHEMA CONVERSION

Step 1: Mapping of Regular Entity Types.

For each regular (strong) entity type E in the ER schema, create a relation R that includes all the simple attributes of E. Include only the simple component attributes of a composite attribute. Choose one of the key attributes of E as the primary key for R. If the chosen key of E is a composite, then the set of simple attributes that form it will together form the primary key of R. If multiple keys were identified for E during the conceptual design, the information describing the attributes that form each additional key is kept in order to specify secondary (unique) keys of relation R. Knowledge about keys is also kept for indexing purposes and other types of analyses.

Step 2: Mapping of Weak Entity Types.

For each weak entity type W in the ER schema with owner entity type E, create a relation R and include all simple attributes (or simple components of composite attributes) of was attributes of R. In addition, include as foreign key attributes of R, the primary key attribute(s) of the relation(s) that correspond to the owner entity type(s); this takes care of mapping the identifying relationship type of W. The primary key of R is the combination of the primary key(s) of the owner(s) and the partial key of the weak entity type W, if any. If there is a weak entity type E2 whose owner is also a weak entity type E1, then E1 should be mapped before E2 to determine its primary key first.

Step 3: Mapping of Binary 1:1 Relationship Types.

For each binary 1:1 relationship type R in the ER schema, identify the relations S and T that correspond to the entity types participating in R. There are three possible approaches:

1. The foreign key approach.
2. The merged relationship approach, and the first approach is the most useful and should befollowed unless special conditions exist, as we discuss below.

1.Foreign key approach:

Choose one of the relations—S, say—and include as a foreign key in S the primary key of T. It is better to choose an entity type with total participation in R in the role of S. Include all the simple attributes (or simple components of composite attributes) of the 1:1 relationship type R as attributes of S.

2.Merged relation approach:

An alternative mapping of a 1:1 relationship type is to merge the two entity types and the relationship into a single relation. This is possible when both participations are total, as this would indicate that the two tables will have the exact same number of tuples at all times.

3.Cross-reference or relationship relation approach:

The third option is to set up a third relation R for the purpose of cross-referencing the primary keys of the two relations S and T representing the entity types. As we will see, this approach is required for binary M: N relationships. The relation R is called a relationship relation (or sometimes a lookup table), because each tuple in R represents a relationship instance that relates one tuple from S with one tuple from T. The relation R will include the primary key attributes of S and T as foreign keys to S and T. The primary key of R will be one of the two foreign keys, and the other foreign key will be a unique key of R. The drawback is having an extra relation, and requiring an extra join operation when combining related tuples from the tables.

Step 4: Mapping of Binary 1: N Relationship Types.

For each regular binary 1: N relationship type R, identify the relation S that represents the participating entity type at the N-side of the relationship type. Include as foreign key in S the primary key of the relation T that represents the other entity type participating in R; we do this because each entity instance on the N-side is related to at most one entity instance on the 1-side of the relationship type. Include any simple attributes (or simple components of composite attributes) of the 1: N relationship type as attributes of S.CS&E Dept, B.I.E.T,

Step 5: Mapping of Binary M: N Relationship Types.

For each binary M: N relationship type R, create a new relation S to represent R. Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types; their combination will form the primary key of S. Also include any simple attributes of the M: N relationship type (or simple components of composite attributes) as attributes of S.Notice that we cannot represent an M: N relationship type by a single foreign key attribute in one of the participating relations (as we did for 1:1 or 1: N relationship types) because of the M: N cardinality ratio; we must create a separate relationship relation S.

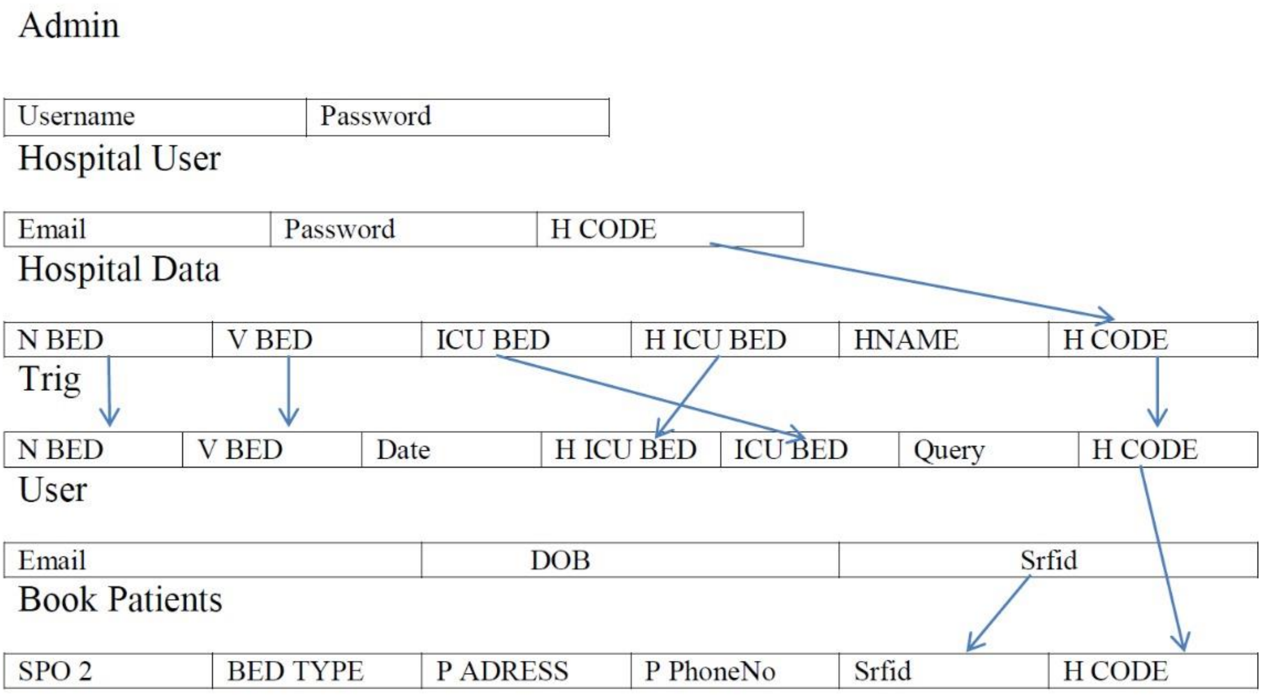
Step 6: Mapping of Multivalued Attributes.

For each multivalued attribute A, create a new relation R. This relation R will include an attribute corresponding to A, plus the primary key attribute K—as a foreign key in R—of the relation that represents the entity type or relationship type that has A as a multivalued attribute. The primary key of R is the combination of A and K. If the multivalued attribute is composite, we include its simple components.

Step 7: Mapping of N-array Relationship Types.

For each n-array relationship type R, where n > 2, create a new relation S to represent R. Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types. Also include any simple attributes of the n-array relationship type (or simple components of composite attributes) as attributes of S. The primary key of S is usually a combination of all the foreign keys that reference the relations representing the participating entity types. However, if the cardinality constraints on any of the entity types E participating in R is 1, then the primary key of S should not include the foreign key attribute that references the relation E _corresponding to E.

3.3 SCHEMA DIAGRAM



Admin

Username	Password

Hospital User

Email	Password	H CODE

Hospital Data

N BED	V BED	ICU BED	H ICU BED	HNAME	H CODE

Trig

N BED	V BED	Date	H ICU BED	ICU BED	Query	H CODE

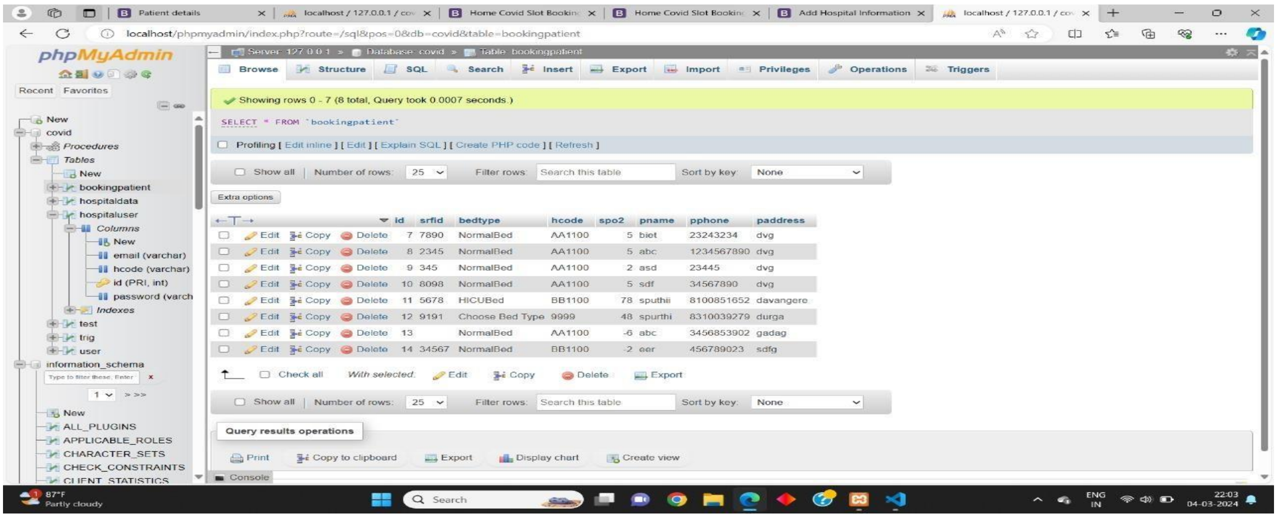
User

Email	DOB	Srfid

Book Patients

SPO 2	BED TYPE	P ADRESS	P PhoneNo	Srfid	H CODE

3.5 DATABASE DESCRIPTION



localhost/phpmyadmin/index.php?route=/sql&pos=0&db=covid&table=bookingpatient

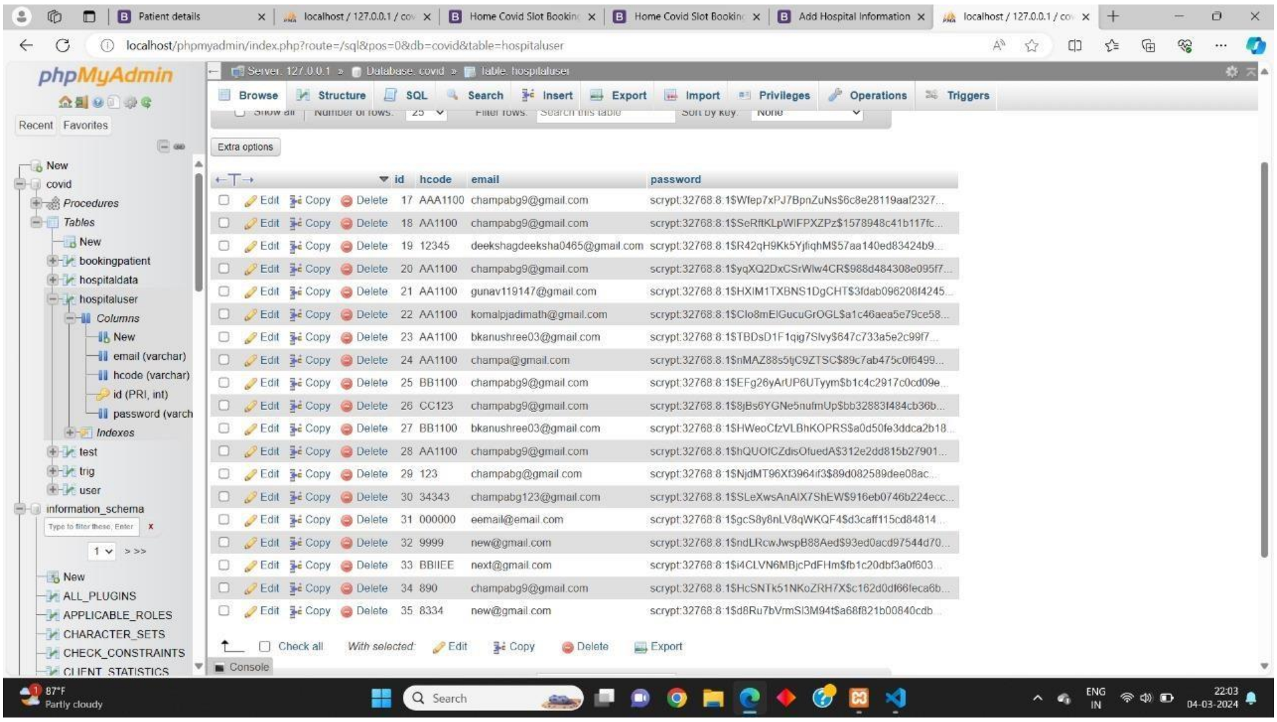
Server: 127.0.0.1 » Database: covid » Table: bookingpatient

✓ Showing rows 0 - 7 (8 total, Query took 0.0007 seconds.)

```
SELECT * FROM `bookingpatient`
```

←T→	id	srfid	bedtype	hcode	spo2	pname	pphone	paddress
□ Edit Copy Delete	7	7890	NormalBed	AA1100	5	biet	23243234	dvg
□ Edit Copy Delete	8	2345	NormalBed	AA1100	5	abc	1234567890	dvg
□ Edit Copy Delete	9	345	NormalBed	AA1100	2	asd	23445	dvg
□ Edit Copy Delete	10	8098	NormalBed	AA1100	5	sdf	34567890	dvg
□ Edit Copy Delete	11	5678	HICUBed	BB1100	78	sputhii	8100851652	davangere
□ Edit Copy Delete	12	9191	Choose Bed Type	9999	48	spurthi	8310039279	durga
□ Edit Copy Delete	13		NormalBed	AA1100	-6	abc	3456853902	gadag
□ Edit Copy Delete	14	34567	NormalBed	BB1100	-2	eer	456789023	sdfg

Table 3.4.1: description of booking patients

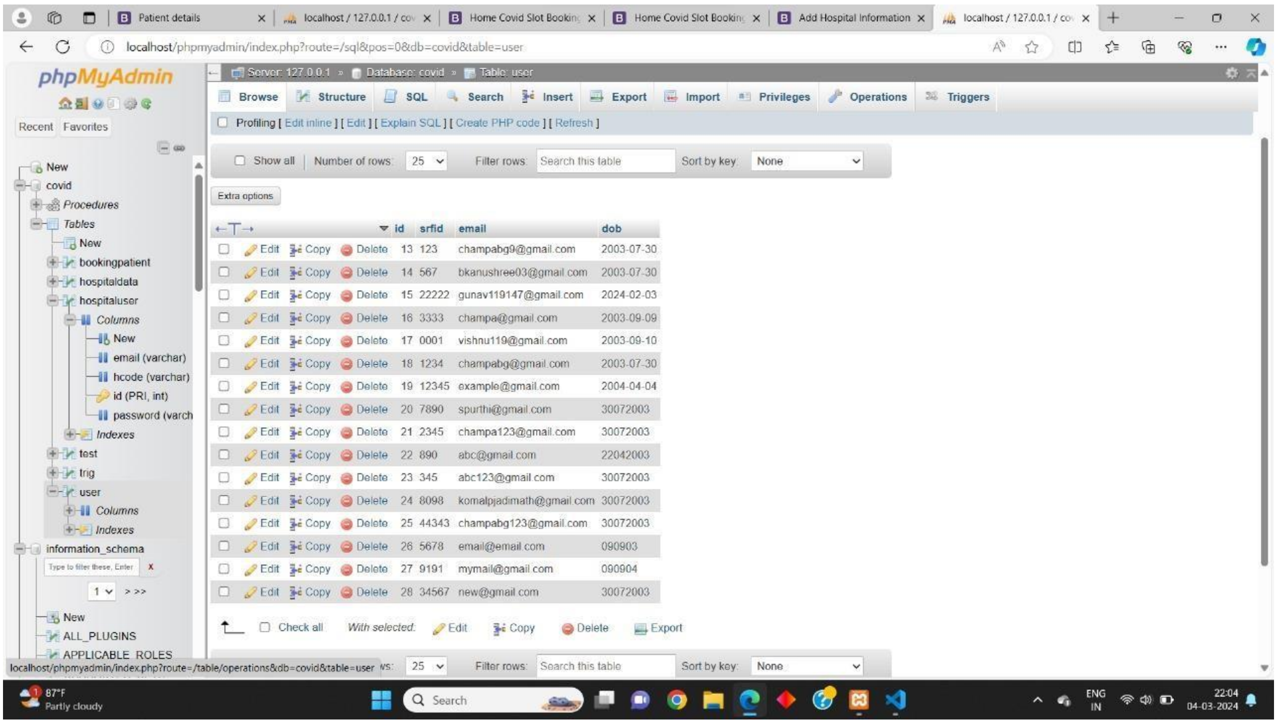


localhost/phpmyadmin/index.php?route=/sql&pos=0&db=covid&table=hospitaluser

Server: 127.0.0.1 » Database: covid » Table: hospitaluser

←T→	id	hcode	email	password
□ Edit Copy Delete	17	AAA1100	champabg9@gmail.com	scrypt:32768:8:1$Wfep7xPJ7BpnZuNs$6c8e28119aaf2327...
□ Edit Copy Delete	18	AA1100	champabg9@gmail.com	scrypt:32768:8:1$SeRftKLpWIFPXZPz$1578948c41b117fc...
□ Edit Copy Delete	19	12345	deekshagdeeksha0465@gmail.com	scrypt:32768:8:1$R42qH9Kk5YjfiqhM$57aa140ed83424b9...
□ Edit Copy Delete	20	AA1100	champabg9@gmail.com	scrypt:32768:8:1$yqXQ2DxCSrWlw4CR$988d484308e095f7...
□ Edit Copy Delete	21	AA1100	gunav119147@gmail.com	scrypt:32768:8:1$HXIM1TXBNS1DgCHT$3fdab096208f4245...
□ Edit Copy Delete	22	AA1100	komalpjadimath@gmail.com	scrypt:32768:8:1$Clo8mElGucuGrOGL$a1c46aea5e79ce58...
□ Edit Copy Delete	23	AA1100	bkanushree03@gmail.com	scrypt:32768:8:1$TBDsD1F1qig7Slvy$647c733a5e2c99f7...
□ Edit Copy Delete	24	AA1100	champa@gmail.com	scrypt:32768:8:1$nMAZ88s5tjC9ZTSC$89c7ab475c0f6499...
□ Edit Copy Delete	25	BB1100	champabg9@gmail.com	scrypt:32768:8:1$EFg26yArUP6UTyym$b1c4c2917c0cd09e...
□ Edit Copy Delete	26	CC123	champabg9@gmail.com	scrypt:32768:8:1$8jBs6YGNe5nufmUp$bb32883f484cb36b...
□ Edit Copy Delete	27	BB1100	bkanushree03@gmail.com	scrypt:32768:8:1$HWeoCfzVLBhKOPRS$a0d50fe3ddca2b18...
□ Edit Copy Delete	28	AA1100	champabg9@gmail.com	scrypt:32768:8:1$hQUOfCZdisOfuedA$312e2dd815b27901...
□ Edit Copy Delete	29	123	champabg@gmail.com	scrypt:32768:8:1$NjdMT96Xf3964if3$89d082589dee08ac...
□ Edit Copy Delete	30	34343	champabg123@gmail.com	scrypt:32768:8:1$SLeXwsAnAIX7ShEW$916eb0746b224ecc...
□ Edit Copy Delete	31	000000	eemail@email.com	scrypt:32768:8:1$gcS8y8nLV8qWKQF4$d3caff115cd84814...
□ Edit Copy Delete	32	9999	new@gmail.com	scrypt:32768:8:1$ndLRcwJwspB88Aed$93ed0acd97544d70...
□ Edit Copy Delete	33	BBIIEE	next@gmail.com	scrypt:32768:8:1$i4CLVN6MBjcPdFHm$fb1c20dbf3a0f603...
□ Edit Copy Delete	34	890	champabg9@gmail.com	scrypt:32768:8:1$HcSNTk51NKoZRH7X$c162d0df66feca6b...
□ Edit Copy Delete	35	8334	new@gmail.com	scrypt:32768:8:1$d8Ru7bVmSI3M94t$a68f821b00840cdb...

Table 3.4.2:hospital user table

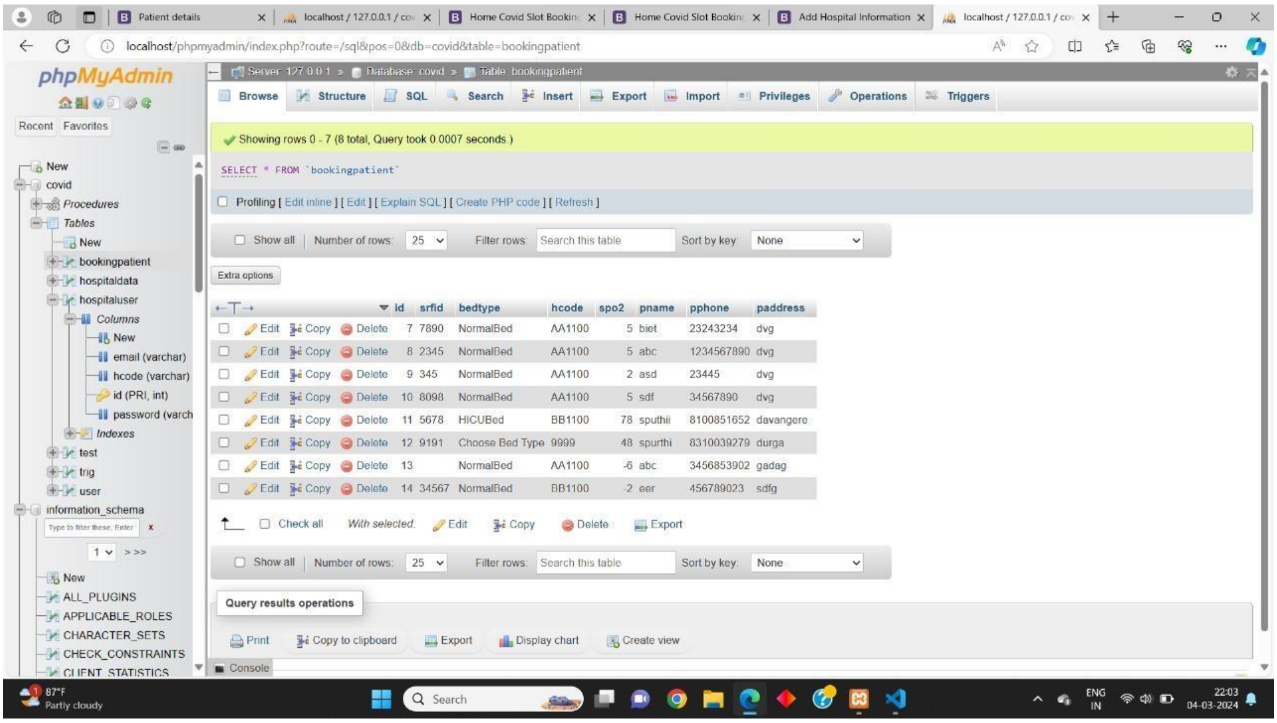


localhost/phpmyadmin/index.php?route=/sql&pos=0&db=covid&table=user

Server: 127.0.0.1 » Database: covid » Table: user

				id	srfid	email	dob
□	Edit	Copy	Delete	13	123	champabg9@gmail.com	2003-07-30
□	Edit	Copy	Delete	14	567	bkanushree03@gmail.com	2003-07-30
□	Edit	Copy	Delete	15	22222	gunav119147@gmail.com	2024-02-03
□	Edit	Copy	Delete	16	3333	champa@gmail.com	2003-09-09
□	Edit	Copy	Delete	17	0001	vishnu119@gmail.com	2003-09-10
□	Edit	Copy	Delete	18	1234	champabg@gmail.com	2003-07-30
□	Edit	Copy	Delete	19	12345	example@gmail.com	2004-04-04
□	Edit	Copy	Delete	20	7890	spurthi@gmail.com	30072003
□	Edit	Copy	Delete	21	2345	champa123@gmail.com	30072003
□	Edit	Copy	Delete	22	890	abc@gmail.com	22042003
□	Edit	Copy	Delete	23	345	abc123@gmail.com	30072003
□	Edit	Copy	Delete	24	8098	komalpjadimath@gmail.com	30072003
□	Edit	Copy	Delete	25	44343	champabg123@gmail.com	30072003
□	Edit	Copy	Delete	26	5678	email@email.com	090903
□	Edit	Copy	Delete	27	9191	mymail@gmail.com	090904
□	Edit	Copy	Delete	28	34567	new@gmail.com	30072003

Table 3.4.3: user table



localhost/phpmyadmin/index.php?route=/sql&pos=0&db=covid&table=bookingpatient

Server: 127.0.0.1 » Database: covid » Table: bookingpatient

Showing rows 0 - 7 (8 total, Query took 0.0007 seconds.)

```
SELECT * FROM `bookingpatient`
```

				id	srfid	bedtype	hcode	spo2	pname	pphone	paddress
□	Edit	Copy	Delete	7	7890	NormalBed	AA1100	5	biet	23243234	dvg
□	Edit	Copy	Delete	8	2345	NormalBed	AA1100	5	abc	1234567890	dvg
□	Edit	Copy	Delete	9	345	NormalBed	AA1100	2	asd	23445	dvg
□	Edit	Copy	Delete	10	8098	NormalBed	AA1100	5	sdf	34567890	dvg
□	Edit	Copy	Delete	11	5678	HICUBed	BB1100	78	sputhii	8100851652	davangere
□	Edit	Copy	Delete	12	9191	Choose Bed Type	9999	48	spurthi	8310039279	durga
□	Edit	Copy	Delete	13		NormalBed	AA1100	-6	abc	3456853902	gadag
□	Edit	Copy	Delete	14	34567	NormalBed	BB1100	-2	eer	456789023	sdfg

Table 3.4.4: hospital user table

CHAPTER 4

IMPLEMENTATION COD

4.2.1 Create Statements

```
-- phpMyAdmin SQL Dump

--version 5.0.2

-- https://www.phpmyadmin.net/

--

-- Host: 127.0.0.1

-- Generation Time: Jan 30, 2022 at 03:04 PM

-- Server version: 10.4.11-MariaDB

-- PHP Version: 7.2.29

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";

START TRANSACTION;

SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;

/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;

/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;

/*!40101 SET NAMES utf8mb4 */;

-- Database: covid

DELIMITER $$

-- Procedures
```

```
    CREATE DEFINER=root@localhost PROCEDURE getPatientDetails (IN inp
VARCHAR(50))  NO SQL

    SELECT pname,pphone,srfid,bedtype,paddress FROM bookingpatient WHERE hcode=inp$$
    CREATE DEFINER=root@localhost PROCEDURE getUsers ()  NO SQL

    SELECT * FROM user$$

DELIMITER ;

    -- Table structure for table bookingpatient

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;

    -- Table structure for table hospitaldata

    --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;

- Triggers hospitaldata

    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 ;

    -- Table structure for table hospitaluser

    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;

    -- Table structure for table test
    CREATE TABLE test (   id
    int(11) NOT NULL,   name
    varchar(50) NOT NULL
    ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
    -- Dumping data for table test
    INSERT INTO test (id, name) VALUES
    (1, 'anees'),
    (2, 'rehman');
    -- Table structure for table trig
    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;

    -- Dumping data for table trig

    INSERT INTO trig (id, hcode, normalbed, hicubed, icubed, vbed, querys, date) VALUES (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');

    -- Table structure for table user

    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;

    -- Indexes for dumped tables

    -- Indexes for table bookingpatient

    ALTER TABLE bookingpatient

      ADD PRIMARY KEY (id),

      ADD UNIQUE KEY srfid (srfid(20));

    -- Indexes for table hospitaldata

    ALTER TABLE hospitaldata   ADD

    PRIMARY KEY (id),

      ADD UNIQUE KEY hcode (hcode);

    -- Indexes for table hospitaluser

    ALTER TABLE hospitaluser   ADD

    PRIMARY KEY (id);

    -- Indexes for table test

    ALTER TABLE test

      ADD PRIMARY KEY (id);

    -- Indexes for table trig

    ALTER TABLE trig

      ADD PRIMARY KEY (id);

    -- Indexes for table user

    ALTER TABLE user
```

```
  ADD PRIMARY KEY (id),

  ADD UNIQUE KEY srfid (srfid);

-- AUTO_INCREMENT for dumped tables

-- AUTO_INCREMENT for table bookingpatient

ALTER TABLE bookingpatient

  MODIFY id int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=7;

-- AUTO_INCREMENT for table hospitaldata

ALTER TABLE hospitaldata

  MODIFY id int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=7;

-- AUTO_INCREMENT for table hospitaluser

ALTER TABLE hospitaluser

  MODIFY id int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=15;

-- AUTO_INCREMENT for table test

ALTER TABLE test

  MODIFY id int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;

-- AUTO_INCREMENT for table trig

ALTER TABLE trig

  MODIFY id int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=15;

-- AUTO_INCREMENT for table user

ALTER TABLE user

  MODIFY id int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=13;

COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
```

```
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;

/*!40101 SET
COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION *
/;
```

CHAPTER 5 SNAPSHOTS

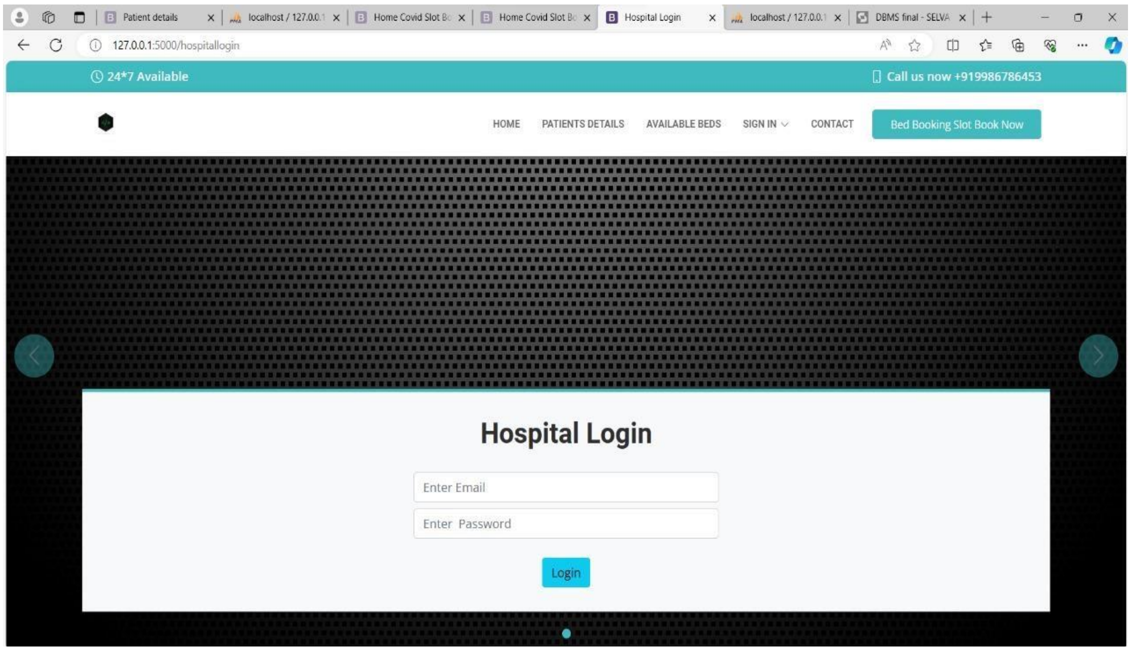


Fig 5.1: Hospital Login Page

Figure 5.1 shows the login page of the hospital bed management system.

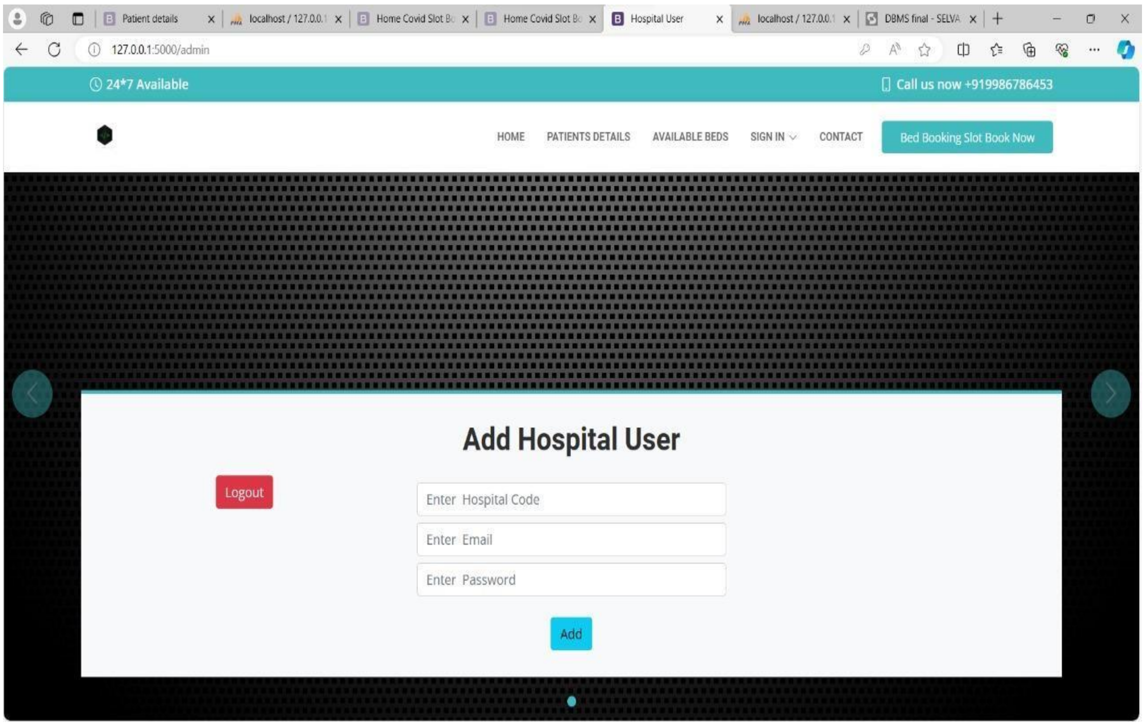
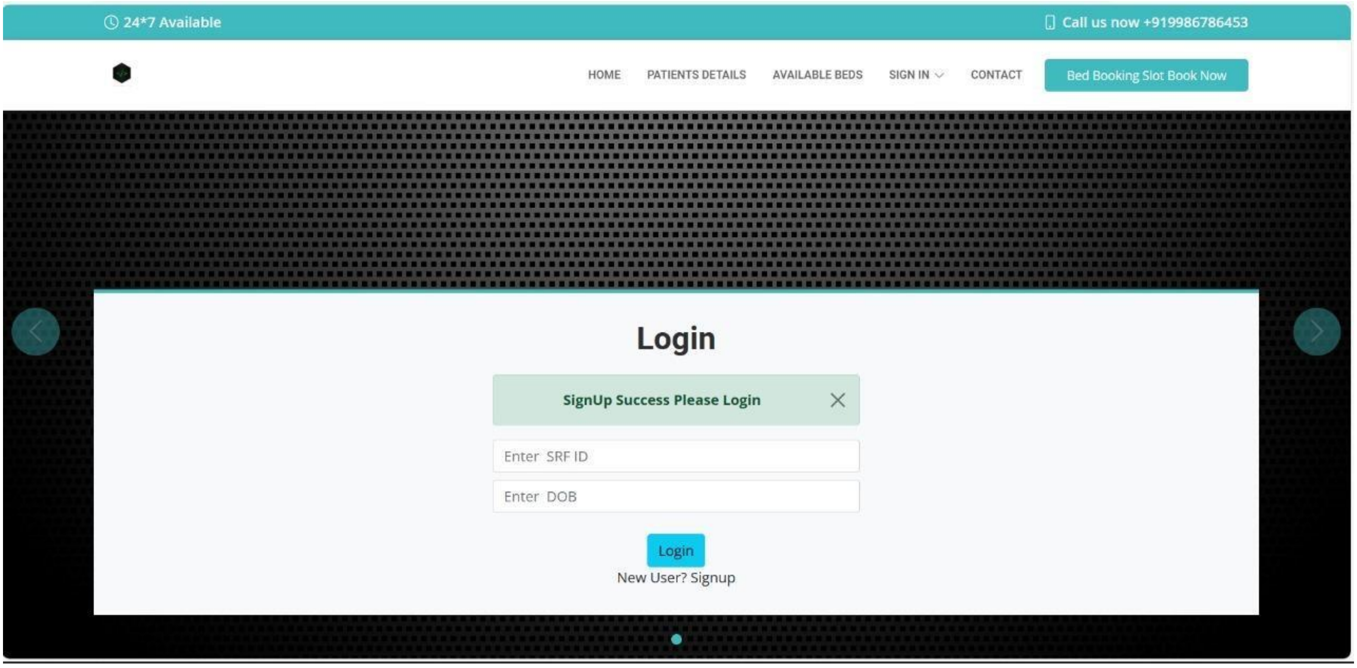


Fig 5.3: Hospital user page

Figure 5.3 shows the hospital users page of the hospital bed management system.



SignUp Success Please Login

Fig 5.4: User Login Page

Figure 5.4 shows the User login page of the hospital bed management system.

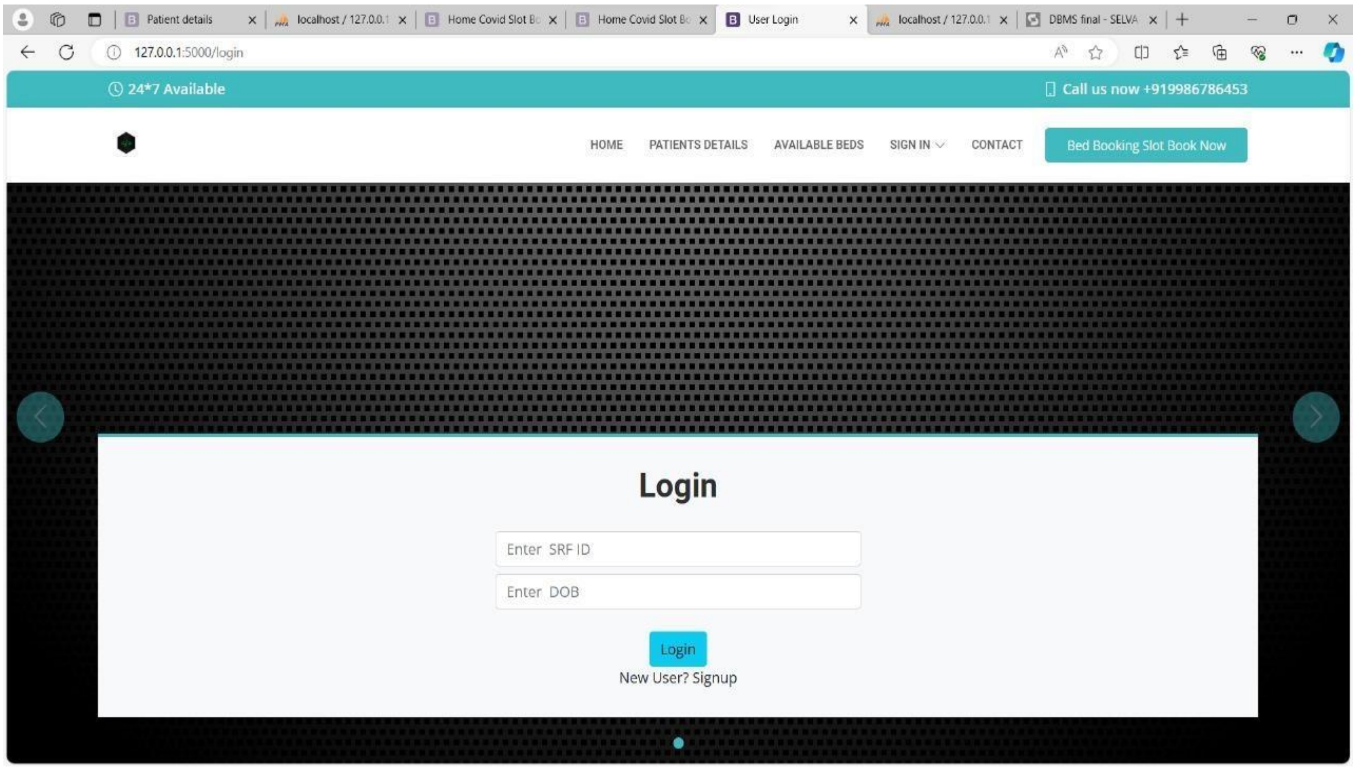
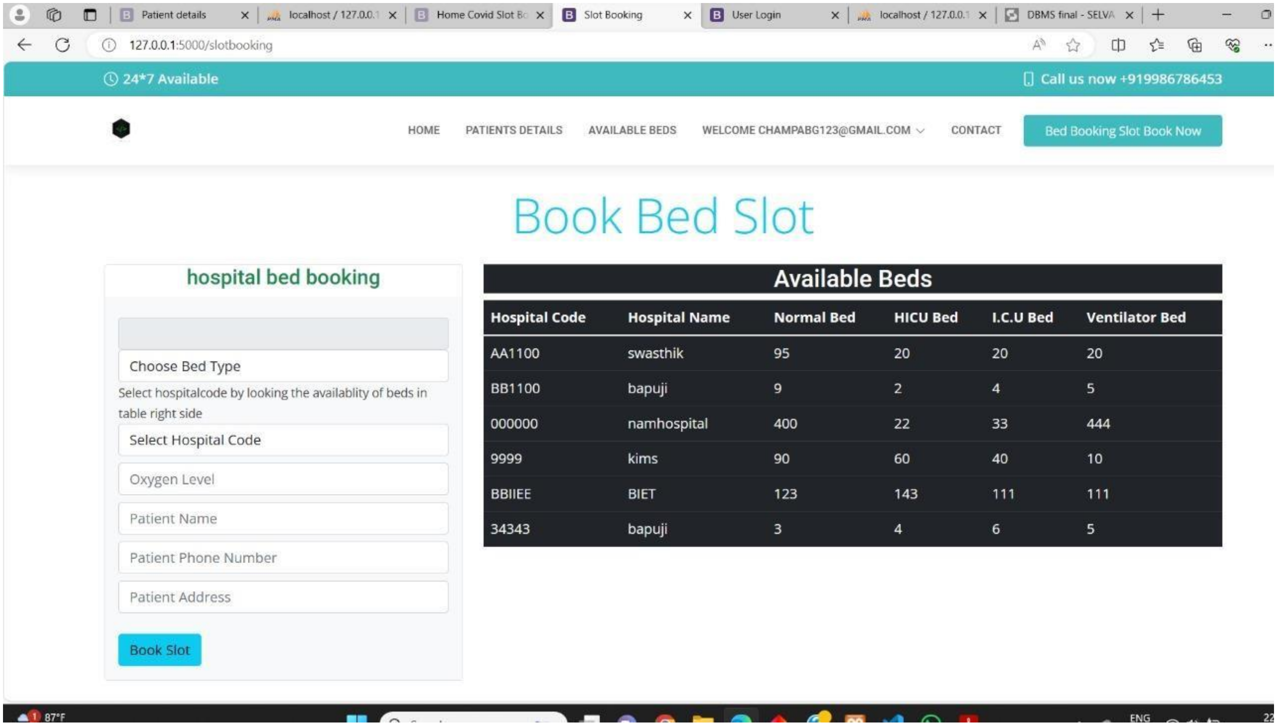


Fig 5.5: User Login Page

Figure 5.5 shows the User login page of the hospital bed management system



Book Bed Slot

hospital bed booking

Choose Bed Type

Select hospitalcode by looking the availablity of beds in table right side

Select Hospital Code

Oxygen Level

Patient Name

Patient Phone Number

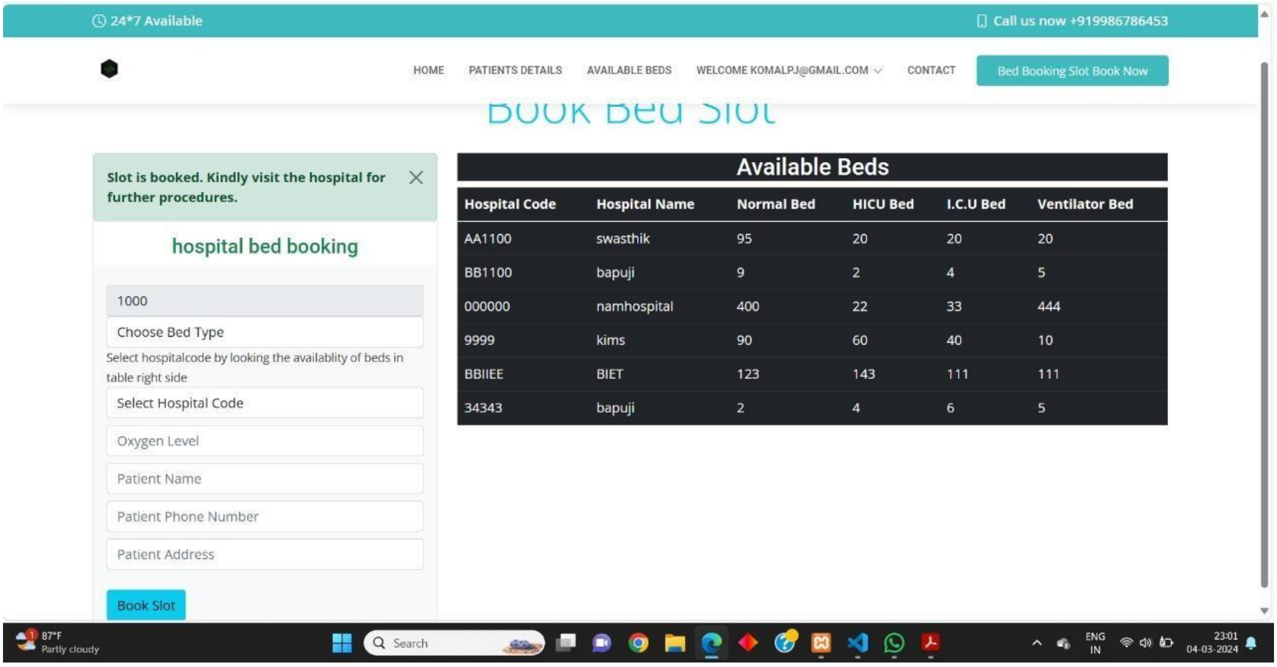
Patient Address

Available Beds

Hospital Code	Hospital Name	Normal Bed	HICU Bed	I.C.U Bed	Ventilator Bed
AA1100	swasthik	95	20	20	20
BB1100	bapuji	9	2	4	5
000000	namhospital	400	22	33	444
9999	kims	90	60	40	10
BBIIEE	BIET	123	143	111	111
34343	bapuji	3	4	6	5

Fig 5.6: book bed slot

Figure 5.5 shows the bed booking of the hospital bed management system



Slot is booked. Kindly visit the hospital for further procedures.

hospital bed booking

1000

Choose Bed Type

Select hospitalcode by looking the availablity of beds in table right side

Select Hospital Code

Oxygen Level

Patient Name

Patient Phone Number

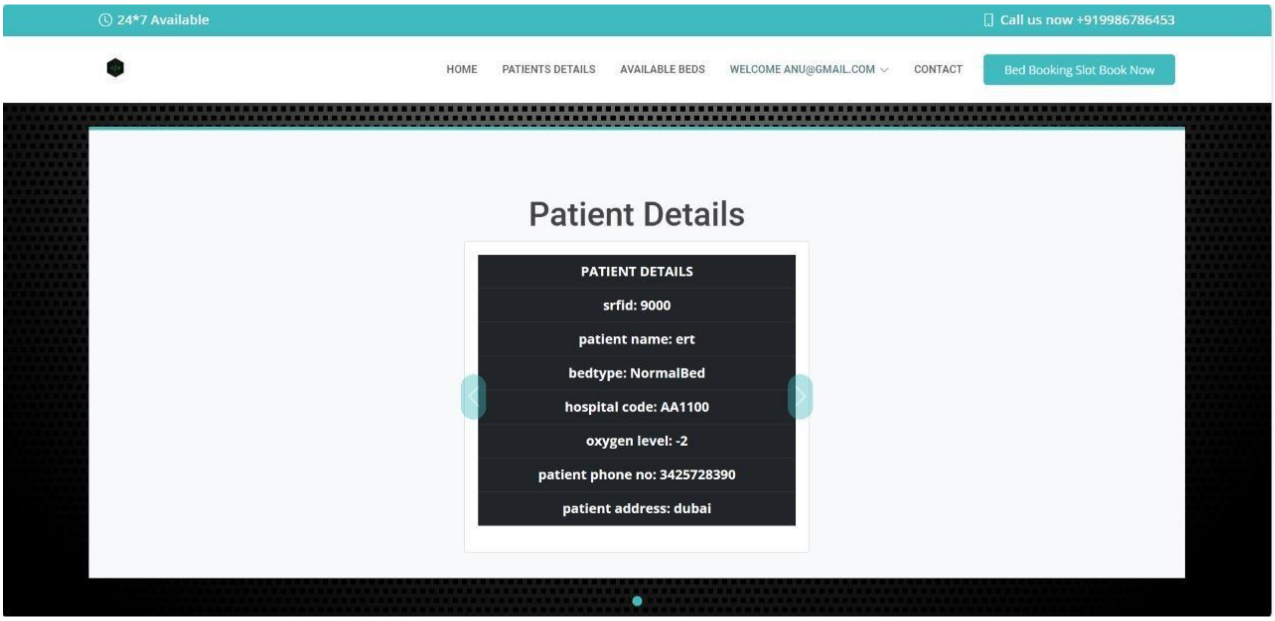
Patient Address

Available Beds

Hospital Code	Hospital Name	Normal Bed	HICU Bed	I.C.U Bed	Ventilator Bed
AA1100	swasthik	95	20	20	20
BB1100	bapuji	9	2	4	5
000000	namhospital	400	22	33	444
9999	kims	90	60	40	10
BBIIEE	BIET	123	143	111	111
34343	bapuji	2	4	6	5

Fig 5.7: bed slot booking succesfull

Figure 5.5 shows the bed booking succesfull of the hospital bed management system



Patient Details

PATIENT DETAILS

srfid: 9000

patient name: ert

bedtype: NormalBed

hospital code: AA1100

oxygen level: -2

patient phone no: 3425728390

patient address: dubai

Fig 5.8: patient details

Figure 5.5 shows the patient details of the hospital bed management system

CONCLUSION

The project Hospital management system is for computerizing and for outreach of making the bed available to the infected user or patient and vaccination related queries. The computerization of the system has speed up the process. In present circumstances the management system is very hectic, confusing and a lot of mistakes can be made. The Hospital management system was thoroughly checked and run along with duplicate data and thus is found to be very reliable. This software takes care of all the requirements. The system also provides the option for backing up all the information available on the system. HOSPITAL BED SLOT BOOKING SYSTEM successfully implemented based on online data filling which helps us in administering the data user for managing all the details related to the covid disease including the vaccination details and successfully allotting beds to the user. The project successfully used various functionalities of Xampp and python flask and also created the fully functional database management system for online portals. Using MySQL as the database is highly beneficial as it is free to download, popular and can be easily customized. The data stored in the MySQL database can easily be retrieved and manipulated according to the requirements with basic knowledge of SQL.

REFERENCES

1. Fundamentals of database systems, Ramez Elmasri and S B Navathe, 7th Edition, 2017, Pearson

2. Database management systems, Ramakrishnan, and Gehrke, 3rd Edition, 2014, McGraw Hill.

3. Coronel, Morris, and Rob, Database Principles Fundamentals of Design, Implementation and Management, Cengage Learning 2012.

4. Silberschatz Korth and Sudharshan, Database System Concepts, 6th Edition, McGraw Hill, 2013.

5. https://www.w3schools.com/

6. https://www.tutorialspoint.com/website_development/index.html