

**Visvesvaraya Technological University
Belagavi-590 018, Karnataka**



A Mini Project Report on

**“BLOOD BANK ORGANISATION
DATABASE”**

**Mini Project Report submitted in partial fulfilment of the requirement for the
DBMS Laboratory with Mini Project [18CSL58]**

**Bachelor of Engineering
in
Computer Science and Engineering**

**Submitted by
Sthuthi S [1JT19CS120]
Laharishree S [1JT19CS119]
Tejaswini N [1JT19CS096]**



**Department of Computer Science and Engineering
Jyothy Institute of Technology
Tataguni, Bengaluru-560082**

Jyothy Institute of Technology
Tataguni, Bengaluru-560082
Department of Computer Science and Engineering



CERTIFICATE

Certified that the mini project work entitled “**BLOOD BANK ORGANISATION DATABASE**” carried out by **Sthuthi S [1JT19CS120]**, **Laharishree S [1JT19CS119]** and **Tejaswini N [1JT19CS096]** bonafide students of Jyothy Institute of Technology, in partial fulfilment for the award of **Bachelor of Engineering in Computer Science and Engineering** department of the **Visvesvaraya Technological University, Belagavi** during the year **2019-2020**. 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. Nikitha S
Guide, Asst. Professor
Dept. of CSE

Dr. Prabhanjan S
Professor & HOD
Dept. of CSE

External Viva Examiner

- 1.
- 2.

Signature with Date :

ACKNOWLEDGEMENT

Firstly, we are very grateful to this esteemed institution “**Jyothy Institute of Technology**” for providing us an opportunity to complete our project.

We express our sincere thanks to our **Principal Dr. Gopalakrishna K** for providing us with adequate facilities to undertake this project.

We would like to thank **Dr. Prabhanjan S, Professor and Head of Computer Science** and Engineering Department for providing for his valuable support.

We would like to thank our guides **Mrs.Nikitha S, Assistant Professor** for their keen interest and guidance in preparing this work.

Finally, we would thank all our friends who have helped us directly or indirectly in this project.

Sthuthi S[1JT19CS120]

Laharishree S[1JT19CS119]

Tejaswini N[1JT19CS096]

ABSTRACT

The application is designed to make the existing system more reliable, fast and easy for all, provides a methodical way of managing large databases. For this application we used the backend as SQL to store the data which is used in the application and for the user interface we have used JAVA.

Blood Bank Organization System is a database application designed to automate the different operations in Blood Bank Organisation. Blood Bank Organization System project makes it easy to give information regarding blood type, date of donation of blood, available blood group and many more. After the implementation of the project, the blood searching process is expected to be faster, easier, and reliable. Admin will view the donor side and view the available blood requested by the users.

Any update in the database like availability of blood, last donation date of the donor, blood bank details be made by the admin.

TABLE OF CONTENTS

SL No	Description	PageNo.
1	INTRODUCTION	1-3
2	DESIGN	4-7
3	IMPLEMENTATION	8-17
4	RESULTS AND SNAPSHOTS	18-27
5	CONCLUSION	28-30

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

1.1 Introduction to DBMS

A database is simply an organized collection of related data, typically stored on disk, and accessible by many concurrent users, it is a logically coherent collection of data with some inherent meaning, representing some aspect of real world and which is designed, built and populated with data for a specific purpose.

Databases are managed by a Database Management System(DBMS) which is a collection of programs that enables user to create and maintain a database.

Advantages of DBMS:

Redundancy is controlled.

Unauthorized access is restricted.

Providing multiple user interfaces.

Enforcing integrity constraints.

Providing backup and recovery.

1.2 Introduction to SQL

Structured Query Language (SQL), is a language used to request data from a database which includes database creation, deletion, retrieval of required tables and even manipulation of data held in a relational database management system.

SQL is considered as a Non-Procedural or a High level language in which the expected result or operation is given without the specific details about how to accomplish the task. So, SQL is a declarative language.

Therefore, SQL is designed at a higher conceptual level of operation than procedural languages as procedural languages includes only the information about opening and closing tables, loading and searching indexes, or flushing buffers and writing data to file systems, but the lower level logical and physical operations are not specified in SQL.

1.3 Introduction to blood bank organization Database

The main goal of the BLOOD BANK ORGANISATION is to manage blood bank data, blood availability, donor data and agency details.

It monitors all the blood transaction and retrieves data of the blood bank, donor, hospital and patient details.

The project's aim is to develop an application system to minimize the manual work for blood bank, donor, blood group management. It monitors all the blood availability information, donor list, patient list.

1.4 Scope and importance of work

The scope of the project is to give a simple application to overcome the drawbacks of the normal file processing system.

The main purpose of this project is to create database application for blood bank organization, this system can be used to check the availability of the blood.

This database avoids accumulating blood from individuals who might not be eligible due to health issues and from individuals whose last donated date is less than three months.

It maintains records of the blood bank associated to the organization, hospitals, patients, donors and it maintains efficiency in the database. It prevents redundancy in the data.

The system helps to identify the nearest blood bank location and allows us to contact them.

The functional areas of application that lies under this system are the management of the availability of donors, hospitals, blood banks under the organization at any time.

This database system allows each individual to be recognized with unique id say donor id, blood bank id, hospital id, patient id.

CHAPTER 2

DESIGN

Theory of ER Diagram

The Entity–Relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as **Entity Relationship Diagram (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. By defining the entities, their attributes, and showing the relationships between them, an ER diagram illustrates the logical structure of database.

ER diagrams are used to sketch out the design of a database.

ENTITIES

An entity is an 'object' in the real world with an independent existence and an entity type defines a collection (or set) of entities that have the same attributes. Each entity type in the database is described by its name and attributes.

An entity type is represented in ER diagrams as a rectangular box enclosing the entity type name.

RELATIONSHIPS

A relationship among two or more entities represents an association among the entities and whenever an attribute of one entity refers to another entity, there exists a relationship between the two entities.

In a relationship, a foreign key of one table refers the primary key of the other table and it is represented by diamond shape in ER diagram.

ATTRIBUTES

An attribute represents some property of interest that further describes an entity and the column header of the table shows the attributes. Each attribute in a table has a certain domain which allows it to accept a certain 'set of values' only.

The attribute values, of each entity, will define its characteristics in the table and is represented by oval in the ER diagram.

ER DIAGRAM

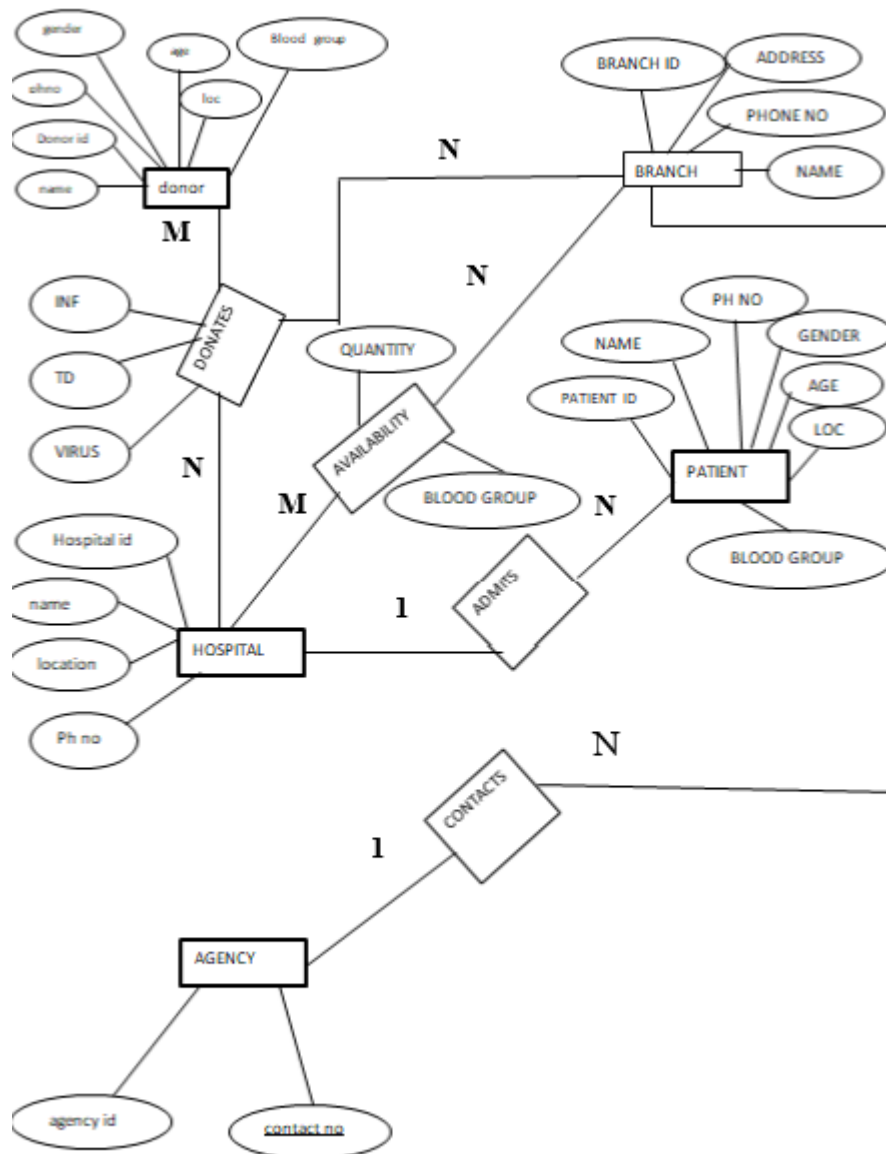


Figure 2.1: ER Diagram for the database

SCHEMA DIAGRAM

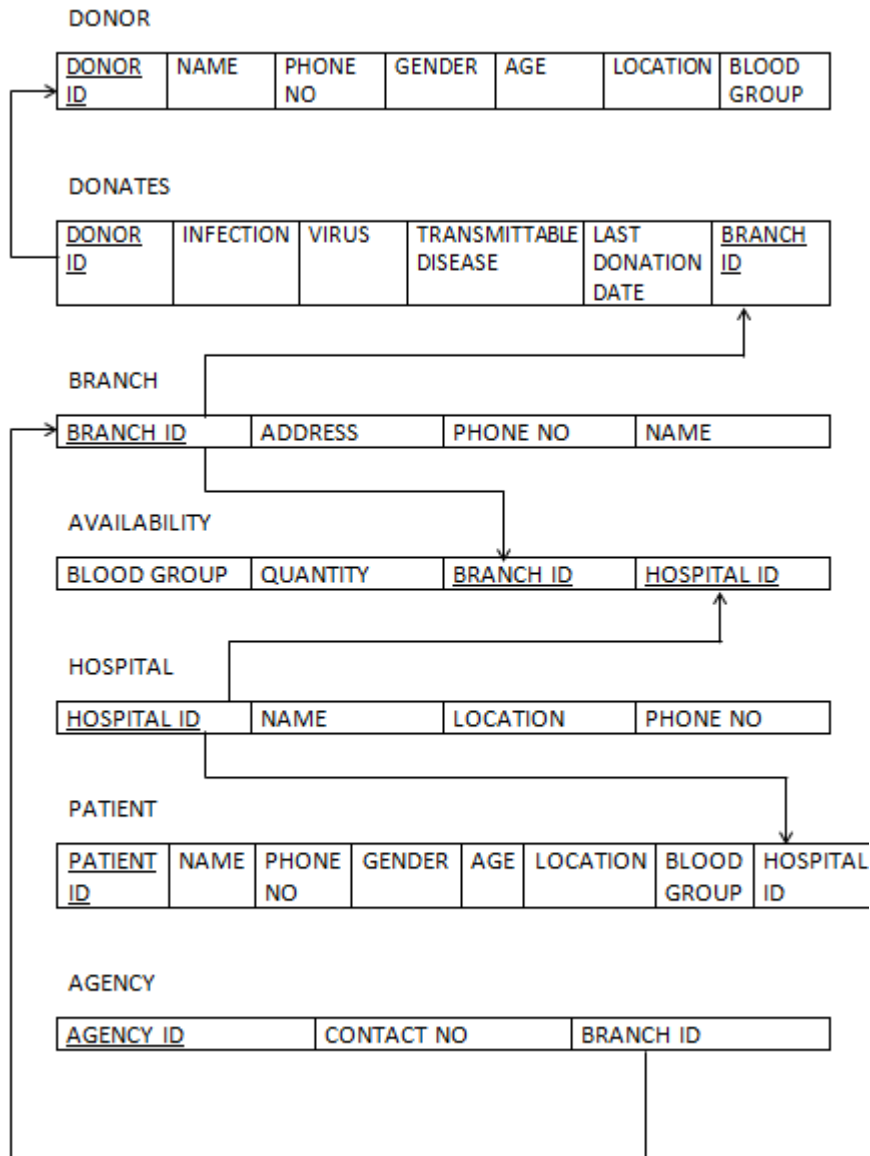


Figure 2.2: Schema Diagram

LIST OF TABLES

1. DONOR: To store the details of donor.
2. DONATES: To store eligibility details.
3. BRANCH: To store branch details.
4. AVAILABILITY: To store availability details.
5. HOSPITAL: To store the hospital details.
6. PATIENT: To store the patients details.
7. AGENCY: To store agency details.

CHAPTER 3

IMPLEMENTATION

Create table commands:-

```
CREATE TABLE DONOR(DONOR_ID INT PRIMARY KEY,NAME
VARCHAR(20),PH_NO VARCHAR(20),GENDER VARCHAR(2),AGE
INT,LOCATION VARCHAR(20),BLOOD_GROUP VARCHAR(20));
```

```
CREATE TABLE DONATES(DONOR_ID INT PRIMARY KEY,INFECTION
VARCHAR(20),VIRUS VARCHAR(20),TRANSMITTABLE_DISEASE
VARCHAR(20),LAST_DONATED DATE, FOREIGN KEY(DONOR_ID)
REFERENCE DONOR(DONOR_ID) ON DELETE CASCADE);
```

```
CREATE TABLE BRANCH(BRANCH_ID INT PRIMARY KEY,ADDRESS
VARCHAR(20),CONTACT VARCHAR(20),NAME VARCHAR(20));
```

```
CREATE TABLE HOSPITAL(HOSPITAL_ID INT PRIMARY KEY,NAME
VARCHAR(20),LOCATION VARCHAR(20),PHONE_NO VARCHAR(20));
```

```
CREATE TABLE AVAILABILITY(BLOOD_GROUP VARCHAR(20),QUANTITY
INT,BRANCH_ID INT PRIMARY KEY,HOSPITAL_ID INT VARCHAR(20)
PRIMARY KEY,FOREIGN KEY(BRANCH_ID) REFERENCES
BRANCH(BRANCH_ID) ON DELETE CASCADE, FOREIGN
KEY(HOSPITAL_ID) REFERENCES HOSPITAL(HOSPITAL_ID) ON DELETE
CASCADE);
```

```
CREATE TABLE PATIENT(PATIENT_ID INT PRIMARY KEY,NAME
VARCHAR(20),PHONE NO VARCHAR(20),GENDER VARCHAR(20),AGE
INT,LOCATION VARCHAR(20),BLOOD_GROUP VARCHAR(20),HOSPITAL_ID
INT,FOREIGN KEY(HOSPITAL_ID) REFERENCES HOSPITAL(HOSPITAL_ID)
ON DELETE CASCADE);
```

```
CREATE TABLE AGENCY(AGENCY_ID INT PRIMARY KEY,CONTACT_NO
VARCHAR(20),BRANCH_ID INT, FOREIGN KEY(BRANCH_ID) REFERENCES
BRANCH(BRANCH_ID) ON DELETE CASCADE);
```

Insertion tables values

Insertion of HOSPITAL table

```
INSERT INTO HOSPITAL VALUES(5001,"JAIN HOSPITAL","BANGALORE SOUTH","080-47091543");
INSERT INTO HOSPITAL VALUES(5002,"MANIPAL HOSPITAL","BANGALORE NORTH","080-56478943");
INSERT INTO HOSPITAL VALUES(5003,"APOLLO HOSPITAL","BANGALORE SOUTH","080-472431543");
INSERT INTO HOSPITAL VALUES(5004,"FORTIS HOSPITAL","BANGALORE EAST","080-23491543");
```

Insertion of AGENCY table

```
INSERT INTO AGENCY("56001","08082822433","JEEVA VOLUNTERY");
INSERT INTO AGENCY("56002","08074652343","RASHTRATHANA");
INSERT INTO AGENCY("56001","08095849309","SMILE FOUNDATION");
```

Insertion of AVAILABILITY table

```
INSERT INTO AVAILABILITY VALUES("O-VE","2L",1001,"5001");
INSERT INTO AVAILABILITY VALUES("O+VE","75ML",1001,"5001");
INSERT INTO AVAILABILITY VALUES("AB-VE","1L",1001,"5001");
```

Insertion of DONOR table

```
INSERT INTO DONOR VALUES(1001,"SAMEER",9723456790,"B-VE","BANGALORE","MALE",20,1003);
INSERT INTO DONOR VALUES(1002,"PAVAN","9916718112","B-VE","BANGALORE","MALE",20,1003);
INSERT INTO DONOR VALUES(1001,"MANASA",9008433503,"B-VE","BANGALORE",20,1003);
```

Insertion of DONATES table

```
INSERT INTO DONATES VALUES(1001,1001,"NO","NO","NO","2020-12-12");
INSERT INTO DONATES VALUES(1002,1001,"NO","NO","NO","2020-12-12");
INSERT INTO DONATES VALUES(1003,1001,"NO","NO","NO","2020-12-12");
```

Insertion of PATIENT table

```
INSERT INTO PATIENT VALUES(1001,"JHON",987654321,"MALE","BANGALORE","AB-VE",5001);

INSERT INTO PATIENT VALUES(1002,"JAME",987623456,"MALE","BANGALORE","B-VE",5001);
```


INSERT INTO PATIENT VALUES
(1001,"RAJ",9876512345,"MALE","BANGALORE","A+VE",5001);

Insertion of BRANCH table

INSERT INTO BRANCH VALUES(1001,"SAHANA BLOOD
BANK","BANGALORE","9807564213");

INSERT INTO BRANCH VALUES(1002,"LION'S BLOOD
BANK","BANGALORE","9901155344");

INSERT INTO BRANCH VALUES(1003,"LIFE CARE BLOOD
BANK","BANGALORE","9807645890");

DESCRIPTION OF TABLES

```
MySQL 8.0 Command Line Client
+-----+
| agency |
| availability |
| branch |
| donates |
| donor |
| hospital |
| login |
| patient |
+-----+
8 rows in set (0.21 sec)

mysql> desc agency;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| agency_id | varchar(10) | NO | PRI | NULL |
| phno | varchar(10) | YES | | NULL |
| agency_name | varchar(25) | YES | | NULL |
+-----+
3 rows in set (0.05 sec)

mysql> desc availability;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| BloodGroup | varchar(5) | YES | | NULL |
| Quantity | varchar(10) | YES | | NULL |
| branch_id | varchar(10) | YES | MUL | NULL |
| hospital_id | int | YES | MUL | NULL |
+-----+
4 rows in set (0.02 sec)

mysql>
```

```
MySQL 8.0 Command Line Client
+-----+
| BloodGroup | varchar(5) | YES | | NULL |
| Quantity | varchar(10) | YES | | NULL |
| branch_id | varchar(10) | YES | MUL | NULL |
| hospital_id | int | YES | MUL | NULL |
+-----+
4 rows in set (0.02 sec)

mysql>
mysql> desc branch;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| branchid | varchar(10) | NO | PRI | NULL |
| branchname | varchar(20) | YES | | NULL |
| address | varchar(50) | YES | | NULL |
| phno | varchar(13) | YES | | NULL |
+-----+
4 rows in set (0.01 sec)

mysql> desc donates;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| DonorID | varchar(10) | YES | MUL | NULL |
| branchId | varchar(10) | YES | MUL | NULL |
| infection | varchar(5) | YES | | NULL |
| virus | varchar(5) | YES | | NULL |
| transmittable_diseases | varchar(5) | YES | | NULL |
| last_donated_date | varchar(50) | YES | | NULL |
+-----+
6 rows in set (0.00 sec)

mysql>
```

```
MySQL 8.0 Command Line Client
+-----+
| bloodgroup | varchar(5) | YES | NULL | | |
| quantity  | varchar(10)| YES | NULL | | |
| branch_id | varchar(10)| YES | NULL | | |
| hospital_id | int       | YES | NULL | | |
+-----+
4 rows in set (0.02 sec)

mysql>
mysql> desc branch;
+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+
| branchid   | varchar(10)| NO   | PRI | NULL    |       |
| branchname | varchar(20)| YES  |     | NULL    |       |
| address    | varchar(50)| YES  |     | NULL    |       |
| phno       | varchar(15)| YES  |     | NULL    |       |
+-----+
4 rows in set (0.01 sec)

mysql>

MySQL 8.0 Command Line Client
+-----+
| DonID      | varchar(10) | YES | NULL | | |
| branchid   | varchar(10) | YES | NULL | | |
| infection  | varchar(5)   | YES | NULL | | |
| virus      | varchar(5)   | YES | NULL | | |
| transmittable_diseases | varchar(5) | YES | NULL | | |
| last_donated_date | varchar(50) | YES | NULL | | |
+-----+
6 rows in set (0.00 sec)

mysql> desc donor;
+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+
| donid      | varchar(10)| NO   | PRI | NULL    |       |
| donname    | varchar(20)| YES  |     | NULL    |       |
| phno       | varchar(12)| YES  |     | NULL    |       |
| bloodgrp   | varchar(5) | YES  |     | NULL    |       |
| address    | varchar(50)| YES  |     | NULL    |       |
| gender     | varchar(6) | YES  |     | NULL    |       |
| age        | int        | YES  |     | NULL    |       |
| branchid   | varchar(10)| YES  |     | NULL    |       |
+-----+
8 rows in set (0.04 sec)

mysql>

MySQL 8.0 Command Line Client
mysql> desc donor;
+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+
| donid      | varchar(10)| NO   | PRI | NULL    |       |
| donname    | varchar(20)| YES  |     | NULL    |       |
| phno       | varchar(12)| YES  |     | NULL    |       |
| bloodgrp   | varchar(5) | YES  |     | NULL    |       |
| address    | varchar(50)| YES  |     | NULL    |       |
| gender     | varchar(6) | YES  |     | NULL    |       |
| age        | int        | YES  |     | NULL    |       |
| branchid   | varchar(10)| YES  |     | NULL    |       |
+-----+
8 rows in set (0.04 sec)

mysql> desc hospital;
+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+
| Hospital_ID | int       | NO   | PRI | NULL    |       |
| Hospital_name | varchar(25)| YES  |     | NULL    |       |
| Location    | varchar(50)| YES  |     | NULL    |       |
| Contact_No  | varchar(13)| YES  |     | NULL    |       |
+-----+
4 rows in set (0.00 sec)

mysql>
```

```
MySQL 8.0 Command Line Client
+-----+
| branchid | varchar(10) | YES | MUL | NULL | |
+-----+
8 rows in set (0.04 sec)

mysql> desc hospital;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| Hospital_ID | int | NO | PRI | NULL | |
| Hospital_name | varchar(25) | YES | | NULL | |
| Location | varchar(50) | YES | | NULL | |
| Contact_No | varchar(13) | YES | | NULL | |
+-----+
4 rows in set (0.00 sec)

mysql> desc login;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| username | varchar(20) | YES | | NULL | |
| password | varchar(20) | YES | | NULL | |
+-----+
2 rows in set (0.00 sec)

mysql> desc patient;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| patid | varchar(10) | NO | PRI | NULL | |
| patname | varchar(20) | YES | | NULL | |
| phno | varchar(12) | YES | | NULL | |
| gender | varchar(6) | YES | | NULL | |
| address | varchar(50) | YES | | NULL | |
| bgrp | varchar(5) | YES | | NULL | |
| Hospital_ID | int | YES | MUL | NULL | |
+-----+
7 rows in set (0.00 sec)

mysql>

mysql> select * from agency;
+-----+
| agency_id | phno | agency_name |
+-----+
| 56001 | 0802568954 | Jeeva Voluntary |
| 56002 | 0806795421 | Rashtrotthana |
| 56003 | 0807658943 | Smile Foundation |
| 56004 | 0807658959 | Doctors without Borders |
| 56005 | 0807656421 | Light for the world |
+-----+
5 rows in set (0.00 sec)

mysql>
```

```
MySQL 8.0 Command Line Client

mysql> desc login;
+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+
| username | varchar(20) | NO   | PRI | NULL    |       |
| password | varchar(20) | YES  |     | NULL    |       |
+-----+
2 rows in set (0.08 sec)

mysql>

MySQL 8.0 Command Line Client

mysql> select * from availability;
+-----+
| BloodGroup | Quantity | branch_id | hospital_id |
+-----+
| O+ve       | 2L       | 1001      | 5001        |
| O+ve       | 750ml    | 1001      | 5001        |
| A+ve       | 1.75L    | 1001      | 5001        |
| A+ve       | 500mL    | 1001      | 5001        |
| B+ve       | 1.25L    | 1001      | 5001        |
| B+ve       | 450mL    | 1001      | 5001        |
| AB+ve      | 1L       | 1001      | 5001        |
| AB+ve      | 300mL    | 1001      | 5001        |
| O+ve       | 3L       | 1001      | 5002        |
+-----+
9 rows in set (1.29 sec)

mysql>
```

```
MySQL 8.0 Command Line Client

mysql> select * from donates;
+-----+-----+-----+-----+-----+-----+
| DonID | branchID | infection | virus | transmittable_diseases | last_donated_date |
+-----+-----+-----+-----+-----+-----+
| 1007 | 1001 | No | No | No | Fri Oct 01 00:02:22 PDT 2021 |
| 1008 | 1001 | No | No | No | Sat Oct 02 00:11:53 PDT 2021 |
| 1009 | 1003 | No | No | No | Fri Oct 01 01:36:49 PDT 2021 |
| 1010 | 1004 | No | No | No | Tue Oct 19 01:38:33 PDT 2021 |
| 1011 | 1002 | No | No | No | Thu Sep 23 01:40:05 PDT 2021 |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

```
MySQL 8.0 Command Line Client

mysql> select * from donor;
+-----+-----+-----+-----+-----+-----+-----+
| donid | donname | phno | bloodgrp | address | gender | age | branchid |
+-----+-----+-----+-----+-----+-----+-----+
| 1001 | Sameer | 9113660769 | B+ve | Bangalore | Male | 20 | 1003 |
| 1002 | Smriti | 7019937514 | A+ve | Girinagar, Bangalore | Female | 18 | 1002 |
| 1003 | Mamatha | 8722114189 | A+ve | Bangalore, Girinagar | Female | 48 | 1004 |
| 1004 | Sathish | 9844061634 | A+ve | Bangalore | Male | 52 | 1001 |
| 1005 | Sridhar | 9870657890 | O+ve | Shimoga | Male | 50 | 1001 |
| 1006 | Rashmi | 8907654890 | B+ve | Shimoga | Female | 45 | 1001 |
| 1007 | Shashi | 9870645890 | AB+ve | Shimoga | Male | 50 | 1001 |
| 1008 | Sanjana | 8976549087 | O+ve | Bangalore | Female | 18 | 1001 |
| 1009 | Srinivas | 9870659870 | B-ve | Hosakerehalli, Bangalore | Male | 50 | 1003 |
| 1010 | Suma | 7895468906 | AB+ve | Jayanagar, Bangalore | Female | 35 | 1004 |
| 1011 | Jamie | 6753209478 | O-ve | St.Joseph Street, Bangalore | Female | 28 | 1002 |
+-----+-----+-----+-----+-----+-----+-----+
11 rows in set (0.00 sec)

mysql>
```

```
MySQL 8.0 Command Line Client

mysql> select * from login;
+-----+-----+
| username | password |
+-----+-----+
| admin1   | admin1@root123 |
| admin2   | admin2@root123 |
| admin3   | admin3@root123 |
| Lahari   | root123      |
| Sthuthi  | root123      |
+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

```
MySQL 8.0 Command Line Client

mysql> desc login;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| username | varchar(20) | NO | PRI | NULL | |
| password | varchar(20) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.08 sec)

mysql>
```

```
MySQL 8.0 Command Line Client
mysql> select * from patient;
```

patid	patname	phno	gender	address	bgrp	Hospital_ID
1001	John	7894561235	Male	Bangalore	AB-ve	5001
1002	Raj	9874321078	Male	Jayanagar, Bangalore	B+ve	5002
1003	Namita	9684451379	Female	Banashankari, Bangalore	O+ve	5003
1004	Shaher	9248694965	Male	KR Road, Bangalore	O-ve	5004
1005	Tom	7906548907	Male	Church Street, Bangalore	B-ve	5005

```
5 rows in set (0.00 sec)

mysql>
```


GUI implementation

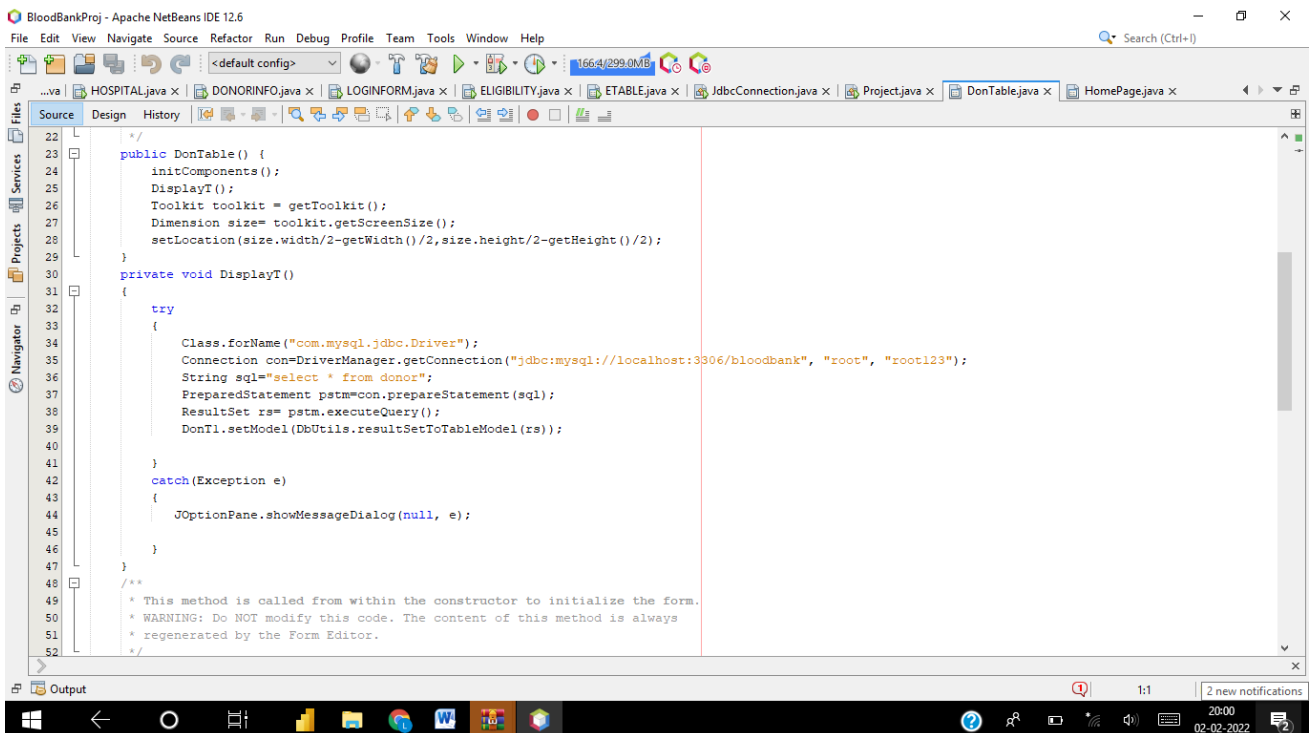


Figure 3.1: SELECTING DONOR TABLE

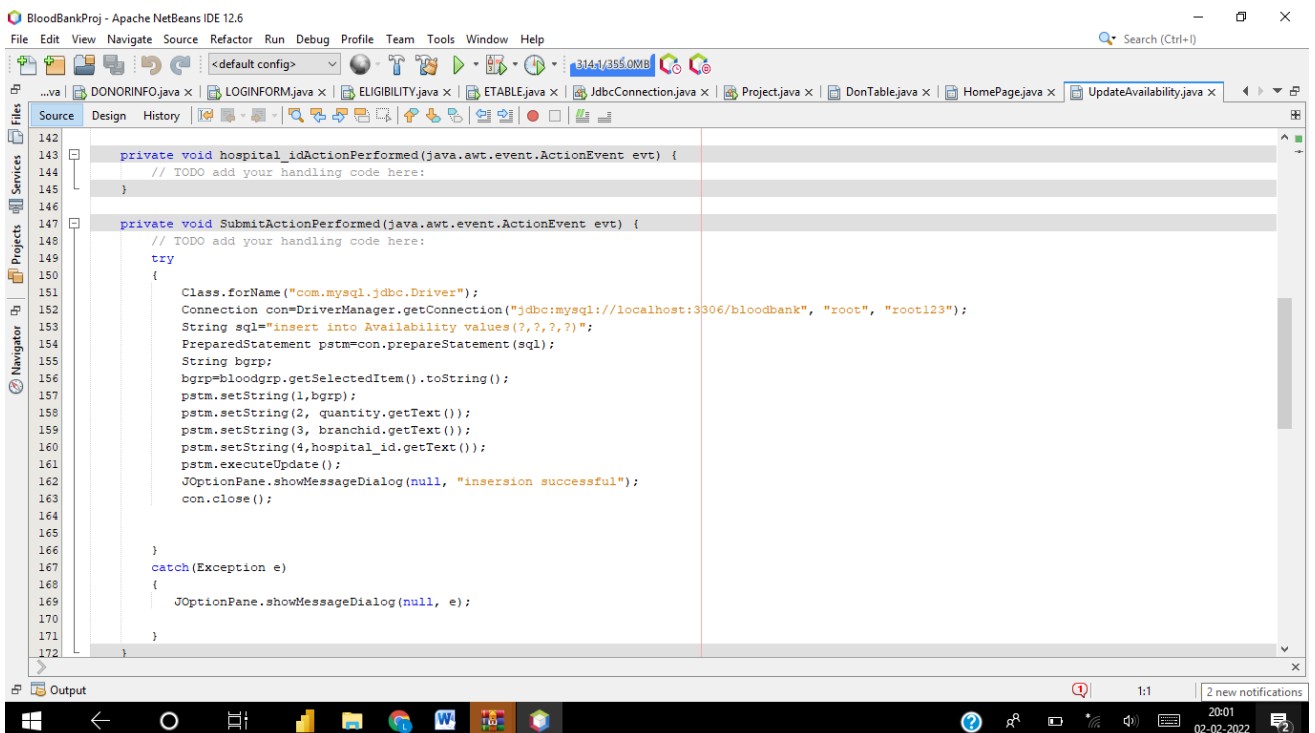


Figure 3.2: INSERTING INTO AVAILABILTY TABLE

CHAPTER 4

RESULTS AND

SNAPSHOTS

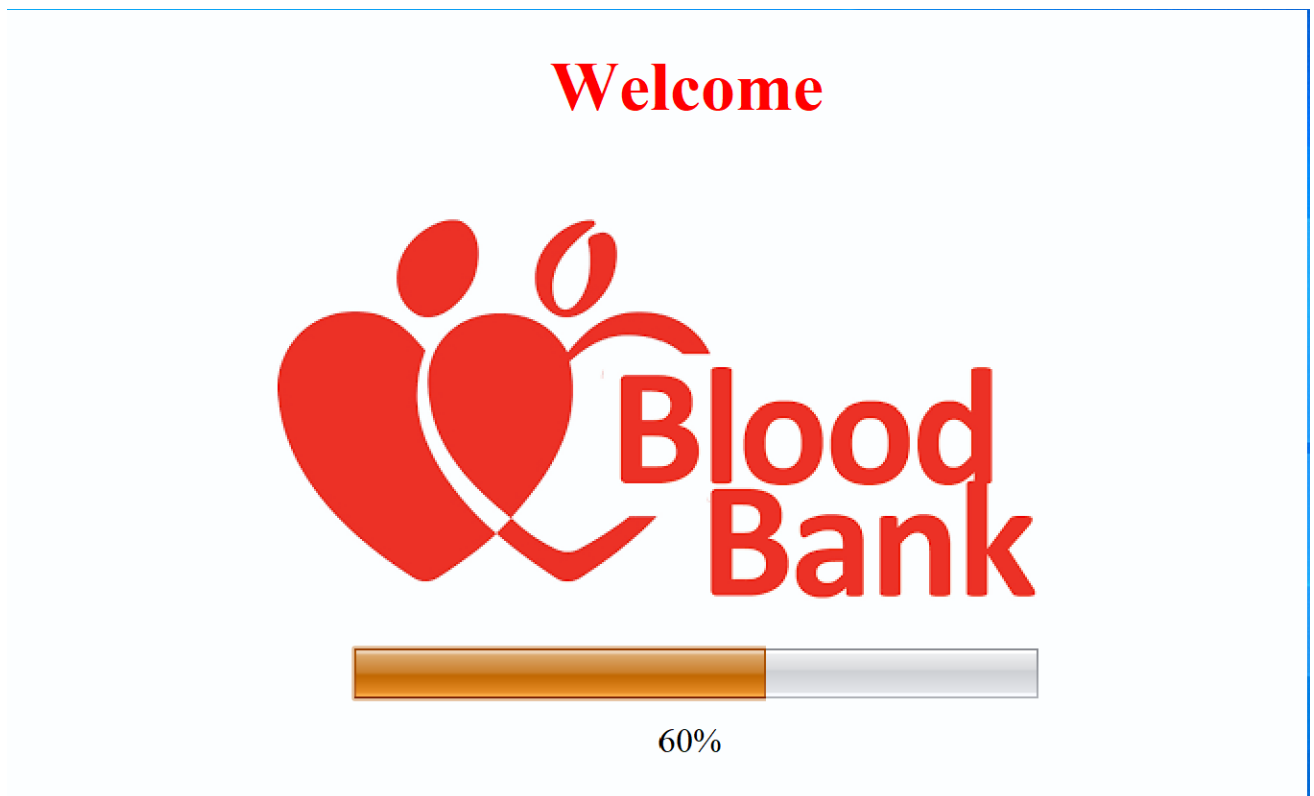


Figure 4.1: LOADING PAGE

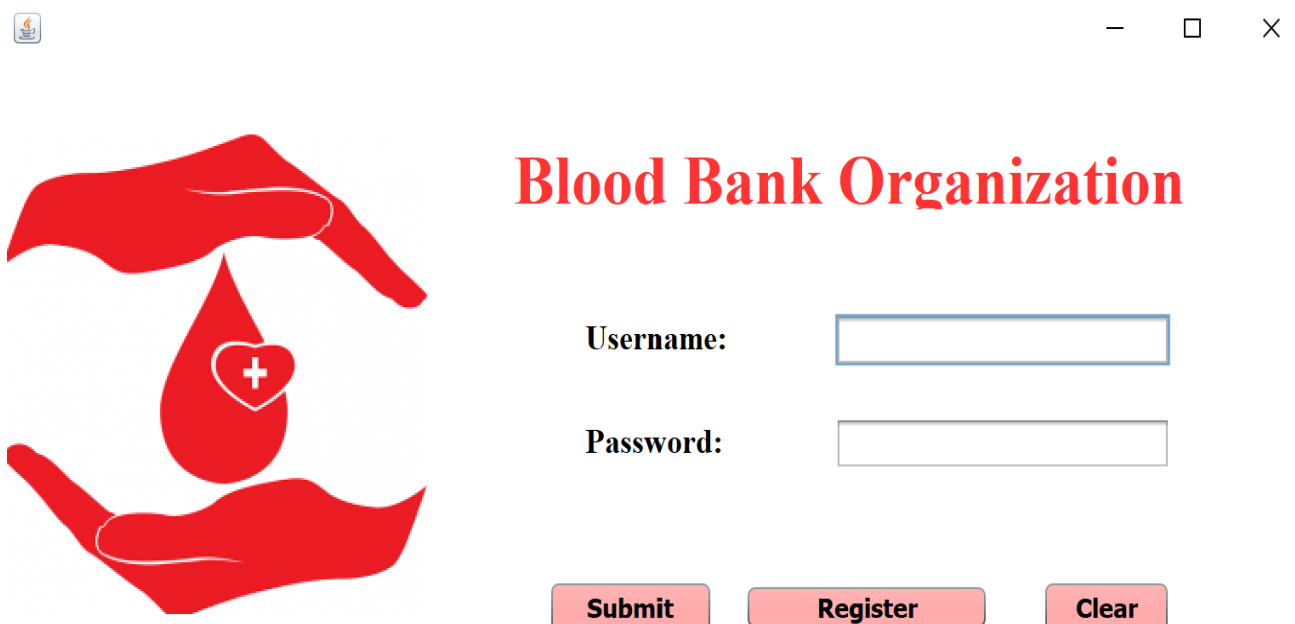


Figure 4.2: LOGIN PAGE

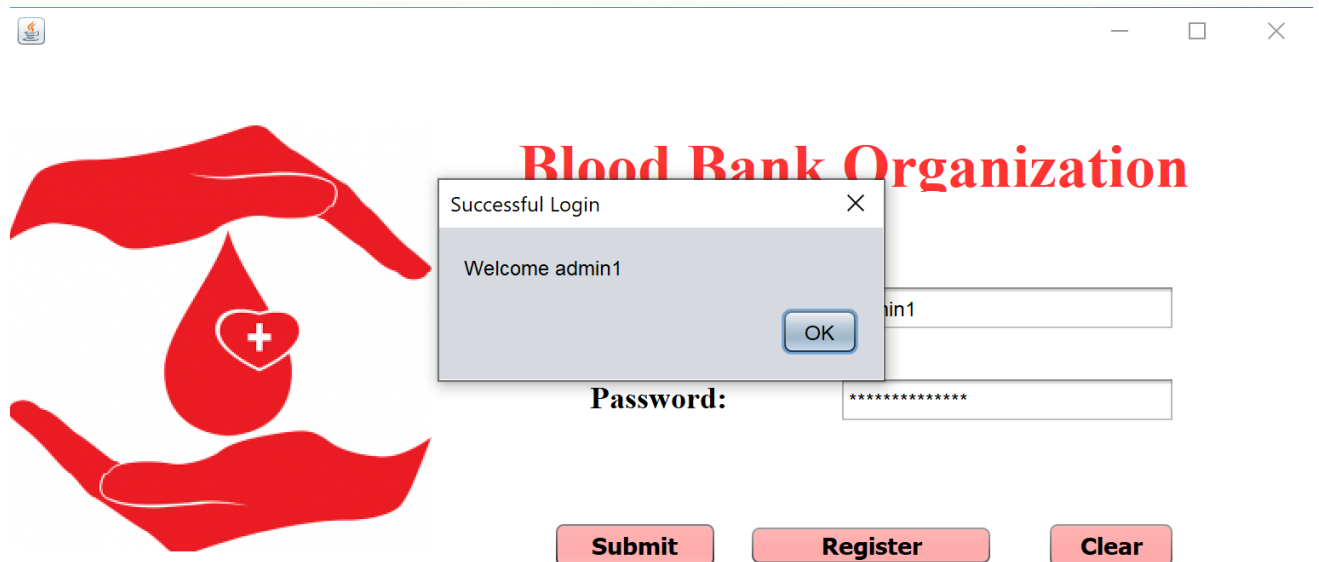


Figure 4.3: SUCCESSFUL LOGIN MESSAGE



Figure 4.4: MENU DISPLAY [HOME PAGE]

The screenshot shows a web application window titled "Donor Form:". The background image is a close-up of a person's arm with a blood donation needle and tube. The form contains the following fields and controls:

- Donor ID:** A text input field.
- Donor Name:** A text input field.
- Contact No.:** A text input field.
- Blood Group:** A dropdown menu with "O+ve" selected.
- Address:** A long text input field.
- Age:** A text input field.
- Gender:** A dropdown menu with "Male" selected.
- Branch ID:** A text input field.
- Buttons:** "Check Eligibility", "View", "Clear", "Delete", and "Back".

Figure 4.5: DONOR FORM

This screenshot shows the same "Donor Form:" window with data entered into the fields. A message dialog box is overlaid on the form, indicating a status check.

Form Data:

- Donor ID:** 1009
- Donor Name:** Srinivas
- Contact No.:** 9870659
- Blood Group:** B-ve
- Address:** Hosakere
- Age:** 50
- Gender:** Male
- Branch ID:** 1003

Message Dialog Box:

- Title:** Message
- Icon:** Information icon (i)
- Text:** Checking Eligibility
- Buttons:** OK

Figure 4.6: FILLING DETAILS AND CHECKING FOR ELIGIBILITY

Eligibility Check Form

Donor ID: Branch ID:

Any Infection to the donor? ☐ Yes ☐ No

Any Virus infection to the donor? ☐ Yes ☐ No

Does the donor have any transmittable diseases? ☐ Yes ☐ No

Last Donated Date:

Check

Figure 4.7: ELIGIBILTY FORM

Eligibility Check Form

Donor ID: Branch ID:

Any Infection to the donor? ☐ Yes ☒ No

Any Virus i... ☐ Yes ☒ No

Does the donor have a... ☐ Yes ☒ No

Last Donated Date:

Check

Message

Eligible to donate

Figure 4.8: FILLING THE ELIGIBILTY FORM

Donor Form:

Donor ID: Donor Name:

Contact No.: Blood Group:

Address:

Age: Gender:

Branch ID:

Message X

New Donor details inserted!

Figure 4.9: SUCCESSFUL INSERTION MESSAGE

Patient Form:

Patient ID: Patient Name:

Phone No.: Gender:

Address:

Hospital ID: Blood Group:

Figure 4.10: PATIENT FORM

Patient Form:

Patient ID: 1006 Patient Name: Ahalya

Phone No.: 7865404891

Address: RajajiNagar, B

Hospital ID: 5003 Blood Group: A+ve

Message: insersion successful

Buttons: Submit, View, Clear, Delete, Back

Figure 4.11: INSERTION OF PATIENT FORM

Patient Details:

Blood Group: O+ve

Search

patid	patname	phno	gender	address	bgrp	Hospital_ID
1003	Namita	9684451379	Female	Banashankari, Bangalore	O+ve	5003

Back

Figure 4.12: DISPLAYING PATIENTS DETAILS



Figure 4.13: DISPLAYING BRANCH DETAILS

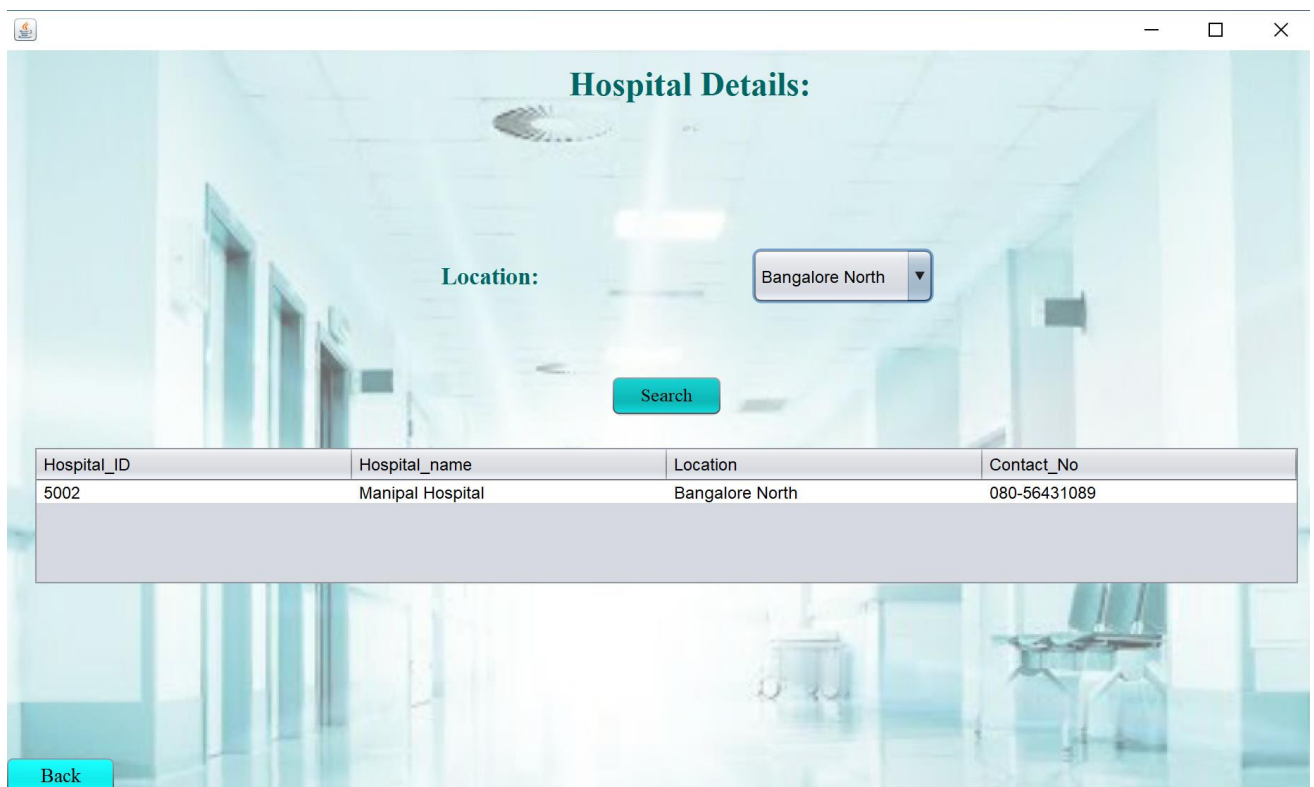


Figure 4.14: DISPLAYING HOSPITAL DETAILS BY LOCATION

[illegible]

Figure 4.15: AVAILABILITY FORM

Availability

Branch ID:

Hospital ID:

BloodGroup	Quantity	branch_id	hospital_id
O+ve	2L	1001	5001
O-ve	750ml	1001	5001
A+ve	1.75L	1001	5001
A-ve	500mL	1001	5001
B+ve	1.25L	1001	5001
B-ve	450mL	1001	5001
AB+ve	1L	1001	5001
AB-ve	300mL	1001	5001

Figure 4.16: CHECKING AVAILABILITY BY BRANCH ID

The screenshot shows a web application window titled "Availability Form:". The background is a blurred image of blood bags labeled "B" and "AB". The form contains the following fields and controls:

- Branch ID:** A text input field.
- Hospital ID:** A text input field.
- Blood Group:** A dropdown menu currently showing "O+ve".
- Quantity:** A text input field.
- Submit:** A button.
- Back:** A button.

Figure 4.17: UPDATION OF AVAILABILITY OF BLOOD GROUP

This screenshot shows the same "Availability Form:" interface as Figure 4.17, but with a modal message box displayed in the center. The message box has a title bar "Message" and a close button (X). It contains an information icon (i) and the text "Updation successful". An "OK" button is at the bottom right of the message box. In the background, the form fields are partially visible and populated with the following values:

- Branch ID:** 1004
- Hospital ID:** 5004
- Blood Group:** B+ve
- Quantity:** 1L

The "Submit" and "Back" buttons remain at the bottom of the form.

Figure 4.18: SUCCESSFUL UPDATION MESSAGE

QUERIES

1.Displaying hospital by location

```
select * from hospital where Location="Banglore north";
```

2.Displaying patient details by blood group

```
select * from patient where blood_group="O+ve";
```

3. Checking availability in respective branch and hospital

```
select * from availability where branch_id=1001 and
hospital_id=5001;
```

4. Updation of blood availability

update Availabilty set Quantity="2L" where BloodGroup="O+ve"
and Branch_id=1001 and Hospital_id=5001;

5. Checking eligibility of donor

The screenshot displays the Apache NetBeans IDE interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The toolbar shows various icons for file operations and development tools. The project explorer on the left shows a project named 'BloodBankProj' with a package 'com.mysql.jdbc' and a class 'Eligibility.java'. The main editor window shows the source code of 'Eligibility.java'. The code is as follows:

```
177 try
178 {
179     Class.forName("com.mysql.jdbc.Driver");
180     Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/bloodbank", "root", "root123");
181     String sql="insert into donates values(?, ?, ?, ?, ?, ?)";
182     PreparedStatement pstmt=con.prepareStatement(sql);
183     String sql1="delete from donor where donid=?";
184     PreparedStatement pstmt1=con.prepareStatement(sql1);
185     pstmt.setString(1, donid.getText());
186     pstmt.setString(1, donid.getText());
187     pstmt.setString(2, branchid.getText());
188     String dod;
189     dod="" + LastDonateDate.getDate();
190     Date dod1;
191     dod1=LastDonateDate.getDate();
192     int monthd=dod1.getMonth();
193     System.out.println(monthd);
194     pstmt.setString(6, dod);
195     System.out.println("date" + dod);
196     /*int monthd=dod.getMonth();
197     pstmt.setString(6, (java.sql.Date) dod);*/
198
199     Date d=new Date();
200     int month = d.getMonth();
201
202     System.out.println("month" + month);
203     String infection="";
204     String virus="";
205     String TransmittableDisease="";
206     if (InfectNo.isSelected())
207     {
```

```

237         else if (VirusYes.isSelected())
238         {
239             virus=VirusYes.getText();
240             JOptionPane.showMessageDialog(null, "Not eligible to donate");
241             pstml.executeUpdate();
242         }
243
244     }
245     else if (InfecYes.isSelected())
246     {
247         infection=InfecYes.getText();
248         JOptionPane.showMessageDialog(null, "Not eligible to donate");
249         pstml.executeUpdate();
250     }
251
252     pstm.setString(3,infection);
253     pstm.setString(4,virus);
254     pstm.setString(5,TransmittableDisease);
255
256
257     pstm.executeUpdate();
258     JOptionPane.showMessageDialog(null, "New Donor details inserted!");
259     con.close();
260
261 }
262 catch (Exception e)
263 {
264     JOptionPane.showMessageDialog(null, e);
265 }
266
267 }

```

```

204 String virus="";
205 String TransmittableDisease="";
206 if (InfecNo.isSelected())
207 {
208     infection=InfecNo.getText();
209     if (VirusNo.isSelected())
210     {
211         virus=VirusNo.getText();
212         if (TDNo.isSelected())
213         {
214             TransmittableDisease=TDNo.getText();
215             if (monthd-month<=8)
216             {
217                 JOptionPane.showMessageDialog(null, "Eligible to donate");
218                 new DonorForm().setVisible(true);
219                 dispose();
220             }
221             else if (monthd-month>8)
222             {
223                 JOptionPane.showMessageDialog(null, "Not eligible to donate");
224                 pstml.executeUpdate();
225             }
226         }
227     }
228     else if (TDYes.isSelected())
229     {
230         TransmittableDisease=TDYes.getText();
231         JOptionPane.showMessageDialog(null, "Not eligible to donate");
232         pstml.executeUpdate();
233     }
234 }

```

Figure 4.19-4.21: Eligibility query

Hospital Details:

Location: Bangalore North

Search

Hospital_ID	Hospital_name	Location	Contact_No
5002	Manipal Hospital	Bangalore North	080-56431089

Back

Figure 4.22: QUERY 1

Patient Details:

Blood Group: O+ve

Search

patid	patname	phno	gender	address	bgrp	Hospital_ID
1003	Namita	9684451379	Female	Banashankari, Bangalore	O+ve	5003

Back

Figure 4.23: QUERY 2

Availability

Branch ID: Hospital ID:

BloodGroup	Quantity	branch_id	hospital_id
O+ve	2L	1001	5001
O-ve	750ml	1001	5001
A+ve	1.75L	1001	5001
A-ve	500mL	1001	5001
B+ve	1.25L	1001	5001
B-ve	450mL	1001	5001
AB+ve	1L	1001	5001
AB-ve	300mL	1001	5001

Figure 4.24: QUERY 3

Availability Form:

Branch ID: Blood Group: Quantity:

Message

Update successful

Figure 4.25: QUERY 4

Eligibility Check Form

Donor ID: Branch ID:

Any ☐ Yes ☒ No

Any Virus i ☐ Yes ☒ No

Does the donor have a ☐ Yes ☒ No

Last Donated Date:

Message

Eligible to donate

Figure 4.26: QUERY 5

CONCLUSION

We have successfully implemented the BLOOD BANK ORGANISATION DATABASE which helps in managing the data used to perform the various tasks in the blood bank organisation.

View tables are used to display all the components of different entities that user needs. One can just select the buttons and modify the data as per requirements.

We have successfully used various functionalities of JAVA and SQL and created the fully functional database system

Blood bank organisation Database has to do with making appropriate effort to stop the rising problem of all manual blood bank operation in order to enhance the operation of such supermarket.

In this project, the software or system that can be used to aid all blood bank organization that is still operating manually have been successfully developed. The software can be implementing in all types of blood bank organization.

Features

1. A password system that will be embedded into login page to increase the Security of the system.
2. A good Printing module should be included.
3. A data required for different operations are accessible to the admin.
4. Quick and easy saving and loading of database file.

References

Net Beans 8.2

<https://docs.oracle.com/netbeans/nb82/netbeans/docs.htm>

JDBC Driver for MySQL (Connector/J)

<https://dev.mysql.com/downloads/connector/j/5.1.html>

MySQL Database

<https://www.mysql.com/downloads/>

[loads/](#)

