JSS SCIENCE AND TECHNOLOGY UNIVERSITY

(Formerly Sri Jayachamarajendra College of Engineering)

JSS TI Campus, Manasagangothri, Mysore – 570006.

CS57L: Database Laboratory

Project Report on “Blood Bank Management System”

Submitted To :

Dr. Manimala S,

Professor, Dept. Of Computer Science,

JSS Science and Technology University,

Mysore – 570006.

Submitted By :

❏	**Vikash Kumar**	: USN – **01JST18CS169**	Roll – **52**
❏	**Vachan Patil**	: USN – **01JST18CS159**	Roll – **47**
❏	**V Harshith**	: USN – **01JST18CS175**	Roll – **52**
❏	**Venkatasudheer S**	: USN – **01JST18CS164**	Roll – **50**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Contents :

1. Acknowledgement

We would like to express our special thanks of gratitude to our teacher Dr.Manimala S, who gave us the golden opportunity to do this wonderful project on the topic " Blood Bank Management System", which also helped us in doing a lot of research and we learnt a great deal by doing them.

We also would like to thank our friends and family, whose guidance was always a support to us. We also enjoyed working as a team and are looking forward to implementing it successfully.

2. Abstract

Our project is a web based Blood Bank Management System. It will help the donors to register and donate blood at the blood bank and manage the entire internal database through one portal. It also helps the medical center to collect blood from the blood bank. For doing this project we have used Django Framework based on Python 3, Bootstrap, HTML, CSS and SQLite on Windows Operating System. The Website is hosted on PythonAnywhere.

The goal of this report is to provide clear specifications and a clear concept of the implementation of the system. This project explores a new dimension to make it more interesting and challenging.

3. Problem Definition

The percentage of people donating blood is increasing day by day due to awareness to donate blood for those needed. The blood received has to be managed thoroughly so that there will be no negative effect to the blood receiver once they receive blood. To handle these internal operations of a Blood Bank we have made a Web based application.

4. Aim

The main aim of developing this system is to provide blood to the people who are in need of blood. The number of people who are in need of blood are increasing in large numbers day by day. Using this system users can search blood groups available in the blood bank and he can also get the contact number of the donor who has the same blood group he needs. In order to help people who are in need of blood, this Online Blood Bank management system can be used effectively for getting the details of available blood groups and users can also get the contact number of the blood donors having the same blood group and within the same city. So if the blood group is not available in the blood bank, the user can request the donor to donate the blood to him and save someone's life.

5. Objective

This blood donation system is an online website so it is easily available to everyone. When a person wants to donate blood he has to register to the system. Donor registration is very easy, to get registered to the system he has to fill up a registration form. Donors have to give information like blood group, contact details etc.

For hospitals to place orders first that hospital has to register to the system. Then the hospital can search blood groups available which are needed. They check it online using our blood bank management website. If a blood group is not available in a blood bank they can also get contact numbers of the persons who have the same blood group they need. And they can request the person to donate blood for saving someone's life.

5. Scope

I. This project covers Web based Blood Bank Management System within a Blood Bank
II. It also covers Donor registration and hospital management which make use of the BLood Bank
III. It provides secure access for admin to the database and users to a limited set of information.
IV. This project is to be built on one databases - Admin database and he has the full power over the complete infrastructure
V. For the users the website works without any delay of login as thought to be important in the case of emergencies where every second is important for the survival of the person

6. System Requirements Engineering

6.1 Functional Requirements

This subsection contains the requirement for the loan section of the banking system. These requirements are gathered on real world banking scenarios. Features form there, they are refined into use case diagrams and sequence diagrams to best capture the functional requirements of the system.

i. User

- Get to know the availability of blood in the database
- Register as a Donor or Hospital
- Place order for the blood required

ii. Admin

- Maintain the entire system and infrastructure
- Check details of table and related data
- Filtering and sorting of details
- Maintaining other users on the database

6.2 Non-functional Requirements

This subsection contains requirements that specify criteria that can be used to judge the operation of a system, rather than specific behaviors. They serve as constraints or restrictions on the design of the system across the different backlogs.

i. **Performance**

- The product shall be based on the web and has to be run on a web server.
- The product shall take initial load time depending on the system hardware.
- The performance shall depend upon hardware components of the client/customer.

ii. **Security**

- The system's back-end servers shall never display a user's password or any other sensitive credentials.
- The system's back-end servers shall only be accessible to authenticated administrators.
- The system's back-end databases shall be encrypted and within the company's perimeter if possible.

iii. **Reliability**

- The reliability of the overall program depends on the reliability of separate components.
- The main pillar of reliability of the system is the backup of the database which is continuously maintained and updated to reflect the most recent changes.

iv. **Maintainability**

- A relational database is used for maintaining the database and the application server takes care of the site.
- In case of a failure, a re-initialization of the program will be done.
- Also, the software design is done with modularity in mind so that maintainability can be done efficiently.

6.3 Security Requirements

- Admin or User id and password protection
- Sign off button
- Failed login attempts
- Check for validity of data of Donor or Hospital while registration attempts

6.4 System Requirements

6.4.1 Software Requirements

This project is developed mainly using open source technologies like Python, Django, Linux based OS, etc.

- Front End: HTML, CSS, Bootstrap, JQuery
- Back End: Django Framework
- Middleware: Django Framework
- Database: Sqlite

6.4.2 Hardware Requirements:

These are the minimum requirements to run the web app smoothly.

- Minimum RAM: 256 MB
- Hard Disk: 10 GB
- Processor: Intel Core i3 3rd Gen or higher
- Operating System: Windows 10/ Linux

7. System Design and Implementation

7.1 Architectural Design

Architectural design elements give us an overall view of the software.It involves identifying the major components of the system and the communication between these components. The architectural design element is usually depicted as a set of interconnected subsystems, often derived from analysis packages within the requirements model. Architectural Style: Our software is based on data centered architecture.
Data centered software architecture is characterized by a centralized data store that is shared by all surrounding components. All the modules or components access this repository to store,retrieve or manipulate data as a result of which any changes made to the database will be updated and this updated version is available to all modules. Main purpose of data centered architecture is to achieve integrality of data. Data-centered architecture consists of different components that communicate through shared data repositories. The components access a shared data structure and are relatively independent, in that, they interact only through the data store.
We are using sqlite in the backend which acts as the data repository for our bank management system. Different modules like customers and accounts interact with this database to insert , update and delete data in the database. The data is consistent and integrity of data is also maintained. We are making use of the Django web framework. It acts as middleware, sqlite for backend and html, css for frontend design.

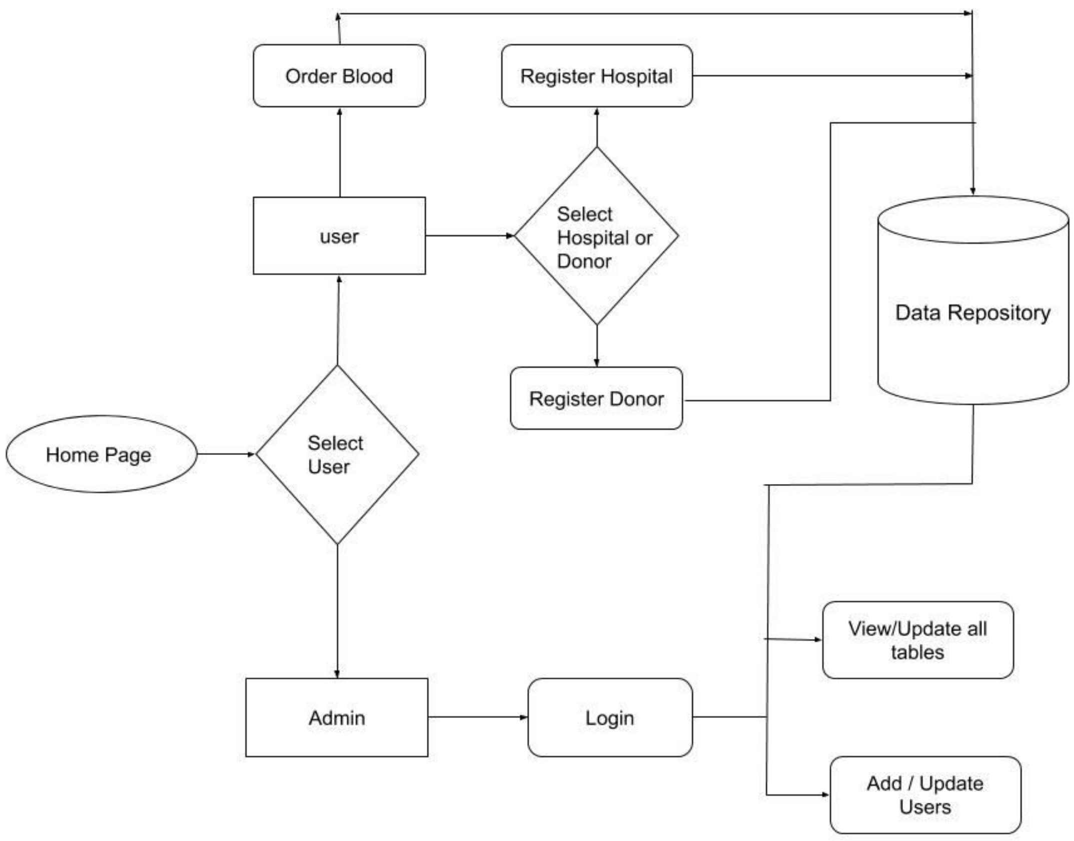


Figure: System Architecture of the WebApp

7.2 Data Flow Diagrams

A data-flow diagram(DFD) is a way of representing the flow of a data of a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself.
There are different levels of DFD's. The description of the system indicating only an overview of the system is represented using DFD level-0. It is also known as context diagram. It's designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as a single bubble with input and output data indicated by incoming/outgoing arrows.

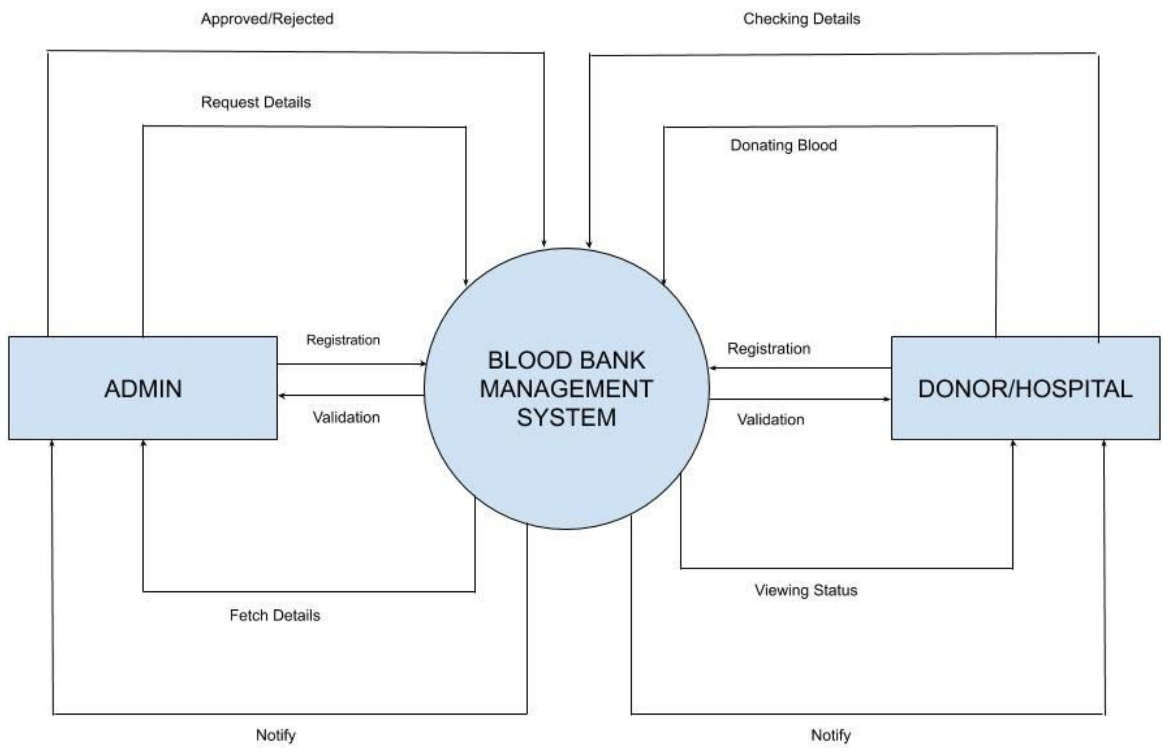


Figure: Level 0 Data Flow Diagram

7.3 E-R DIAGRAM

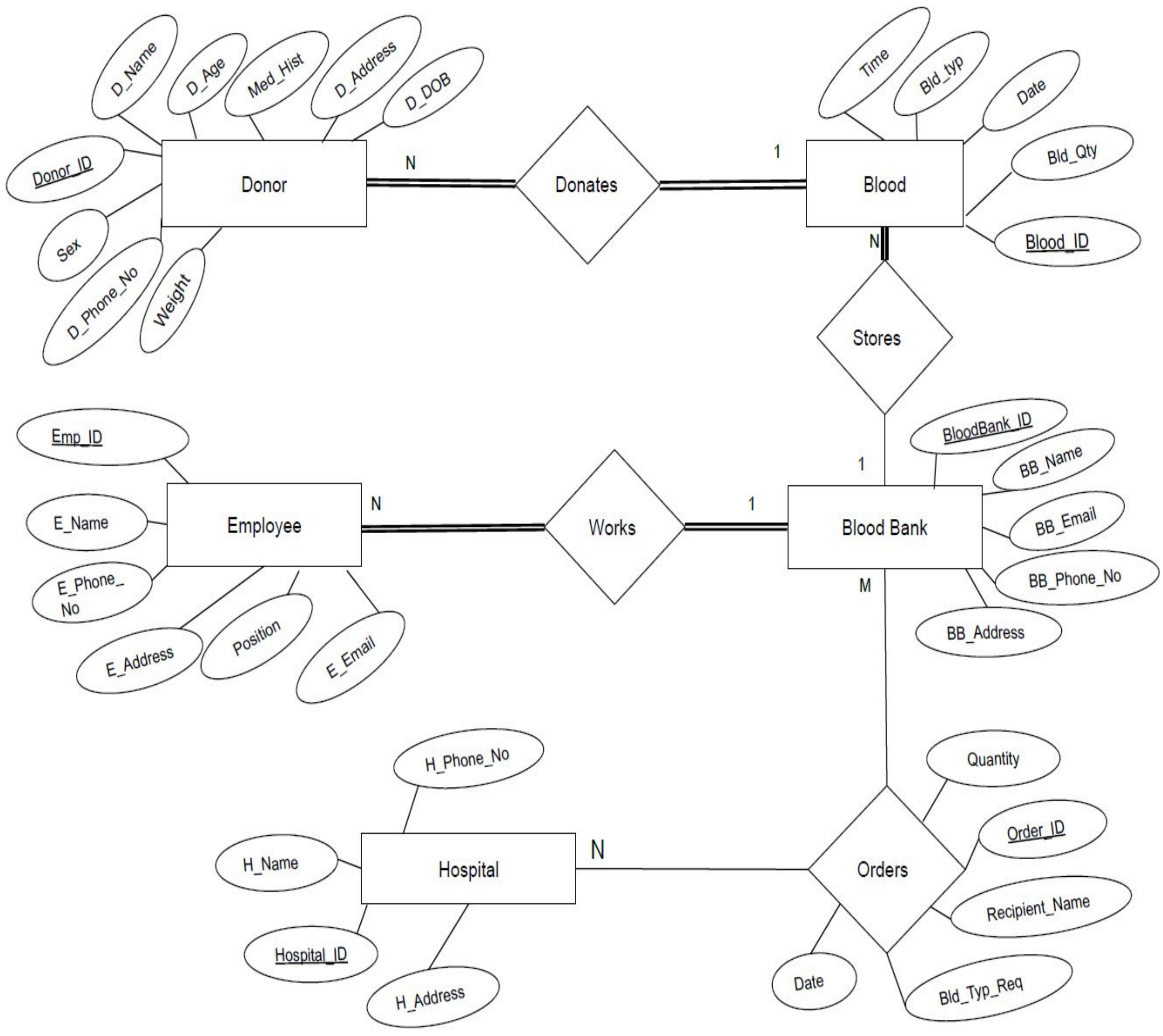
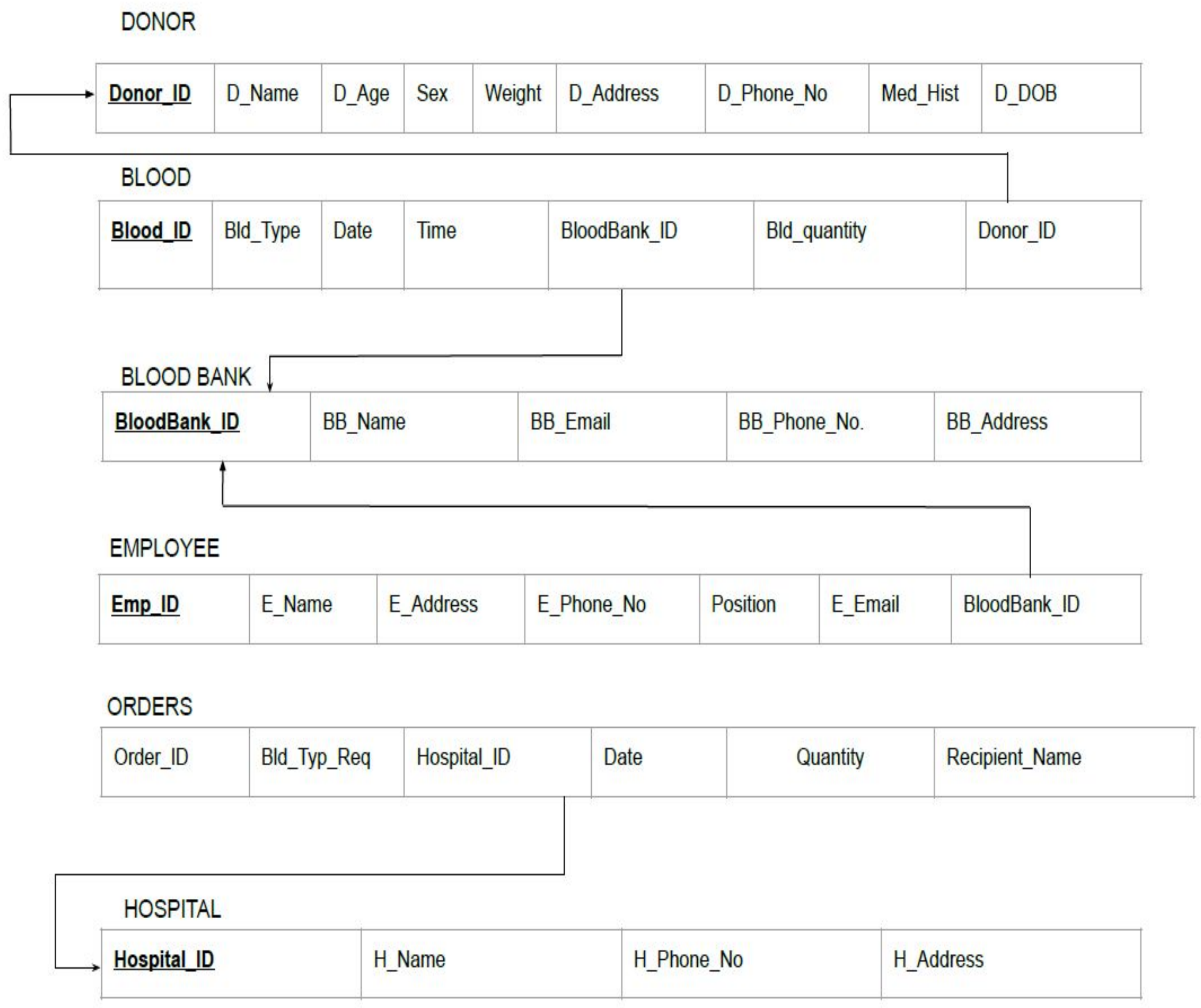


Figure: Entity Relationship Diagram of the Database

7.4 SCHEMA DIAGRAM



7.5 Tables & Attributes

7.5.1 Donor

SL.NO	FIELD	DATA TYPE	CONSTRAINTS	DESCRIPTION
1.	Donor_ID	varchar(20)	Primary Key	Donor ID
2.	D_Name	varchar(30)	Not Null	Donor Name
3.	D_Age	int	Not Null	Donor Age
4.	Sex	varchar(10)	Not Null	Donor Sex
5.	Weight	int	Not Null	Donor Weight
6.	D_Address	varchar(30)	Not Null	Donor Address
7.	D_DOB	date	Not Null	Donor Birth Date
8.	D_Phone_No	varchar(20)	Not Null	Donor Phone No.
9.	Mid_Hist	varchar(30)	Null	Donor Medical History (Any past medical report)

Functional Dependencies:-
R:- Donor(Donor_ID, D_Name, D_Age, Sex,Weight,D_Address, D_DOB,
D_Phone_No, Mid_Hist)
FD:- Donor_ID->{ D_Name, D_Age, Sex,Weight,D_Address, D_DOB,
D_Phone_No, Mid_Hist}

The above FD is in BCNF form.

7.5.2 Blood

SL.NO	FIELD	DATA TYPE	CONSTRAINTS	DESCRIPTION
1.	Blood_ID	varchar(10)	Primary Key	Blood ID
2.	Bld_Type	varchar(10)	Not Null	Blood Type
3.	Date	date	Not Null	Date on which blood is donated
4.	Time	time	Not Null	Time on which blood is donated
5.	Donor_ID	varchar(10)	Foreign Key	Donor ID
6.	Bld_Qty	int	Not Null	Blood Quantity
7.	BloodBank_ID	varchar(10)	Foreign key	BloodBank_ID

Functional Dependencies:-
R:- Blood(Blood_ID, Bld_Type, Date, Time, Donor_ID, Bld_Qty,
BloodBank_ID)

FD:- Blood_ID->{ Bld_Type, Date, Time, Donor_ID, Bld_Qty,
BloodBank_ID}

The above FD is in BCNF form.

7.5.3 BloodBank

SL.NO	FIELD	DATA TYPE	CONSTRAINTS	DESCRIPTION
1.	BloodBank_ID	varchar(20)	Primary Key	Blood Bank ID
2.	BB_Name	varchar(30)	Not Null	Blood Bank Name
3.	BB_Phone_No	varchar(15)	Not Null	Blood Bank Phone No.
4.	BB_Address	varchar(40)	Not Null	Blood Bank Address
5.	BB_Email	varchar(40)	Null	Email Address of Blood Bank

Functional Dependencies:-
R:-Blood bank(BloodBank_ID, BB_Name, BB_Phone_No,
BB_Address, BB_Email)

FD:-BloodBank_ID-> {BB_Name, BB_Phone_No,
BB_Address, BB_Email}

The above FD is in BCNF form.

7.5.4 Hospital

SL.NO	FIELD	DATA TYPE	CONSTRAINT	DESCRIPTION
1.	Hospital_ID	varchar(20)	Primary Key	Hospital ID
2.	H_Name	varchar(30)	Not Null	Hospital Name
3.	H_Phone_No	varchar(15)	Not Null	Hospital Phone No
4.	H_Address	varchar(30)	Null	Hospital Address

Functional Dependencies:-
R:-Hospital(Hospital_ID, H_Name, H_Phone_No, H_Address)
FD:-Hospital_ID->{H_Name, H_Phone_No, H_Address}

The above FD is in BCNF form.

7.5.5 Employee

SL.NO	FIELD	DATA TYPE	CONSTRAINT	DESCRIPTION
1.	Emp_ID	varchar(20)	Primary Key	Employee ID
2.	E_Name	varchar(20)	Not Null	Employee Name
3.	E_Phone_No	varchar(15)	Not Null	Employee Phone No.
4.	E_Address	varchar(40)	Not Null	Employee Address
5.	E_Email	varchar(40)	Null	Email Address of Employee
6.	Position	varchar(20)	Not Null	Role of Employee in Blood Bank
7.	BloodBank_ID	varchar(20)	Foreign Key	Blood Bank ID

Functional Dependencies:-

R:- Employee (Emp_ID, E_Name, E_Phone_No, E_Address, E_Email,
Position, BloodBank_ID)

FD:-Emp_ID->{ E_Name, E_Phone_No, E_Address, E_Email,
Position, BloodBank_ID}

The above FD is in BCNF form.

7.5.6 Orders

SL.NO	FIELD	DATA TYPE	CONSTRAINT	DESCRIPTION
1.	Order_ID	varchar(20)	Primary Key	Order ID
2.	Hospital_ID	varchar(20)	Foreign Key	Hospital ID
3.	Bld_Typ_Req	varchar(20)	Not Null	Blood Type required by Recipient in Hospital
4.	Quantity	int	Not Null	Blood Quantity required
5.	Recipient_name	varchar(30)	Not Null	Receiver Name
6.	Date	date	Not Null	Date on which blood is received

Functional Dependencies:-

R:- Orders (Order_ID, Hospital_ID, Bld_Typ_Req, Quantity, Recipient_name, Date)
FD:-Order_ID-> { Hospital_ID, Bld_Typ_Req, Quantity, Recipient_name, Date}

The above FD is in BCNF form.

7.6 Database Normalization:-

The above Database Has Normalization up to BCNF.

First Normal Form (1NF):-

- A relation is in 1NF if it contains an Atomic Value.
- It states that an attribute of a table cannot hold multiple values. It must hold only single-valued attributes. In other words, First Normal Form disallows the use of Multi-valued Attributes, Composite Attributes and their combinations.
- If a table has multi-valued attributes, the multiple values are added to other rows to make every attribute single-valued.

Second Normal Form (2NF):-

- To be in 2NF, a relation must first be in 1NF.
- In the Second Normal Form, all Non-Key attributes are fully functional dependents of the Primary Key.
- If 2NF is violated by an attribute, the table is decomposed and that attribute is put in the new table with a relation to the Original table.

Third Normal Form (3NF):-

- A Relation is in 3NF if it is in 2NF and does not contain any transitive partial dependency.
- 3NF is used to reduce data duplication and achieve data integrity.
- If there is no transitive dependency for non-prime attributes, then the relation must be in third normal form.

Boyce-Codd Normal Form (BCNF):-

- Relation is in bcnf if it's 3NF.
- A relation schema R is in BCNF if whenever an FD X -> A holds in R, then X is a superkey of R.
- A relation NOT in BCNF should be decomposed so as to meet this property, while possibly forgoing the preservation of all functional dependencies in the decomposed relations.

Therefore, as per the requirements our database is in BCNF.

8. Modules of Blood Bank Management System

Blood Bank Management system is to provide services for the people who are in need of blood by getting help from the donors who are interested in donating blood. There are five main modules in this system.

- Admin
- Donor Registration
- Hospital Registration
- Order
- Availability

Admin:-

Admin can manage all the relations in the database. He can add or remove or modify any information from the database system. The database has following relations:-

- Bloodbanks
- Bloods
- Donors
- Employees
- Hospitals
- Orders

Donor Registration:-

In this module, people who are interested in donating blood get registered in the website and give his overall details related to him, i.e. he fills in a registration form by giving the total details such as name, age, sex ,weight, address, DOB, phone no., Medical History etc.

Hospital Registration:-

In this module, any hospital can get registered in the website and give details about the hospital, i.e. they fill in a registration form by giving total details such as hospital name, hospital phone number and address. Then the hospital can place orders for the blood group available. This can be checked online using our blood bank management website

Order:-

In this module , hospitals or any person in need from a particular hospital can place an order for Blood type request and quantity . and they get registered in the website by giving details about

- Hospital id
- Date
- Quantity
- Blood type request
- Recipient name

Availability:-

In this module, it shows the database relation of blood type and stock available .It will get updated at each transaction. The present availability of the blood and its type can be checked from this page.

9. Unit Testing and Result Analysis

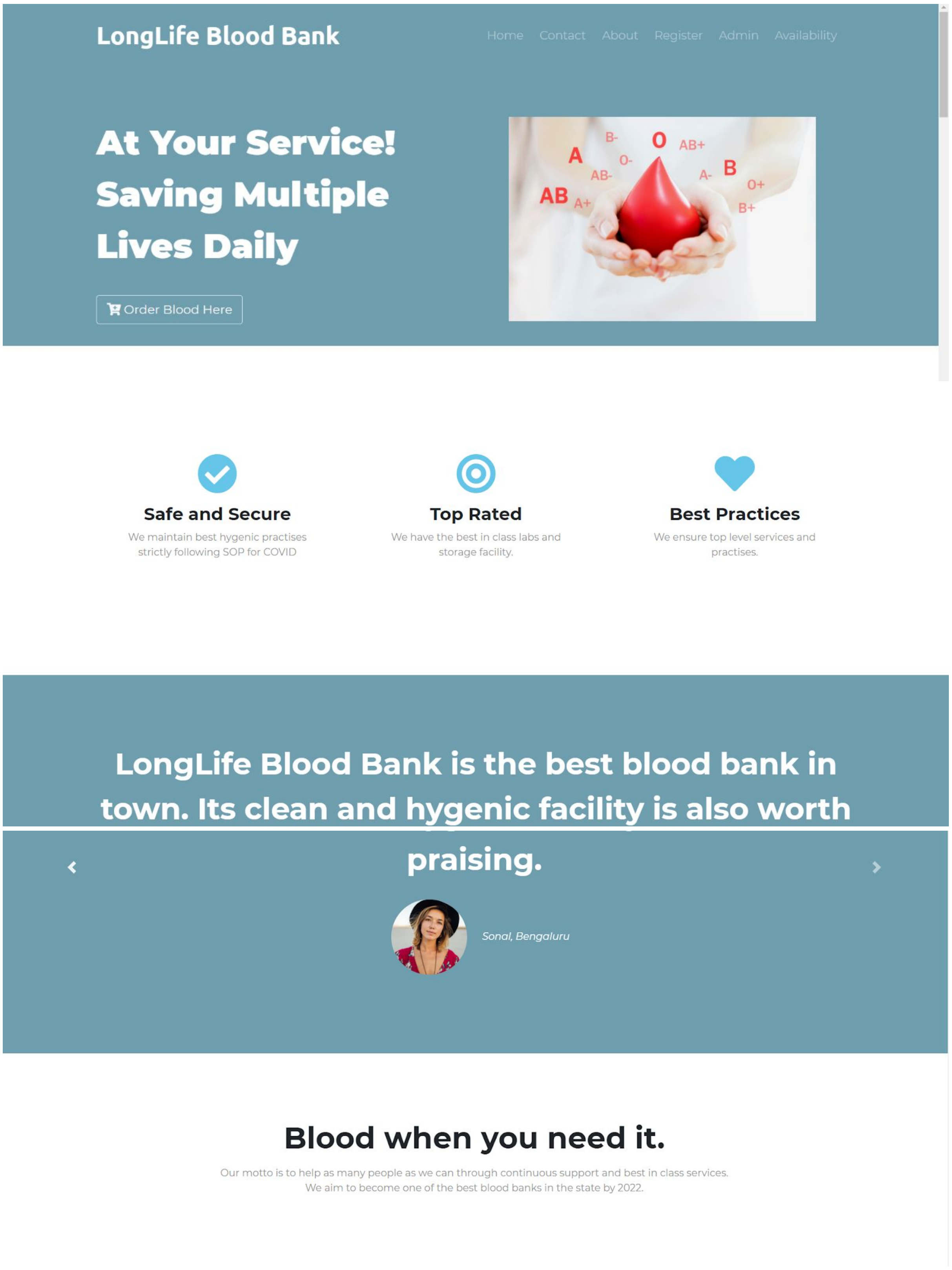
Unit testing focuses verification effort on the smallest unit of software design: the software component or module. Using the component-level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and the errors those tests uncover is limited by constrained scope established for unit testing. The unit test focuses on the internal processing logic and data structures within the boundaries of a component. In this section of the document, we unveil different tests conducted on each module and the behaviour in response.

Test No.	**Feature**	**Input**	**Expected Output**	**Actual Output**	**Remarks**
1.	Successful Login of admin	Enter correct ID and Password	Admin logs to their profile	Admin logs to their profile	pass
2.	Unsuccessful Login of admin	Incorrect ID and Password	Admin does not logs to their profile	Admin does not logs to their profile	pass
3.	View all relations after admin login	Click on the respective table	All relations are displayed with details	All relations are displayed with details	pass

4.	Approve order	Details of the hospital and requirement including blood quantity	Accept if blood quantity is present and update database else reject	Accept if blood quantity is present and update database else reject	pass
5.	Reject order	Order quantity of a particular blood type is not available or data entered is invalid	Reject order and display error message	Reject order and display error message	pass
6.	Accept Donor Registration	User inputs all the details required	Database updated if all data is valid	Database updated if all data is valid	pass
7.	Reject Donor Registration	User inputs invalid Data in the form	No changes happen and error message displayed	No changes happen and error message displayed	pass
8.	Update Availability Table	After every order update table	Table updated on availability page	Table updated on availability page	pass

10. Graphical Representation of the Website

10.1 Home Page



LongLife Blood Bank

Home Contact About Register Admin Availability

At Your Service!
Saving Multiple
Lives Daily

Order Blood Here

Safe and Secure
We maintain best hygenic practises strictly following SOP for COVID

Top Rated
We have the best in class labs and storage facility.

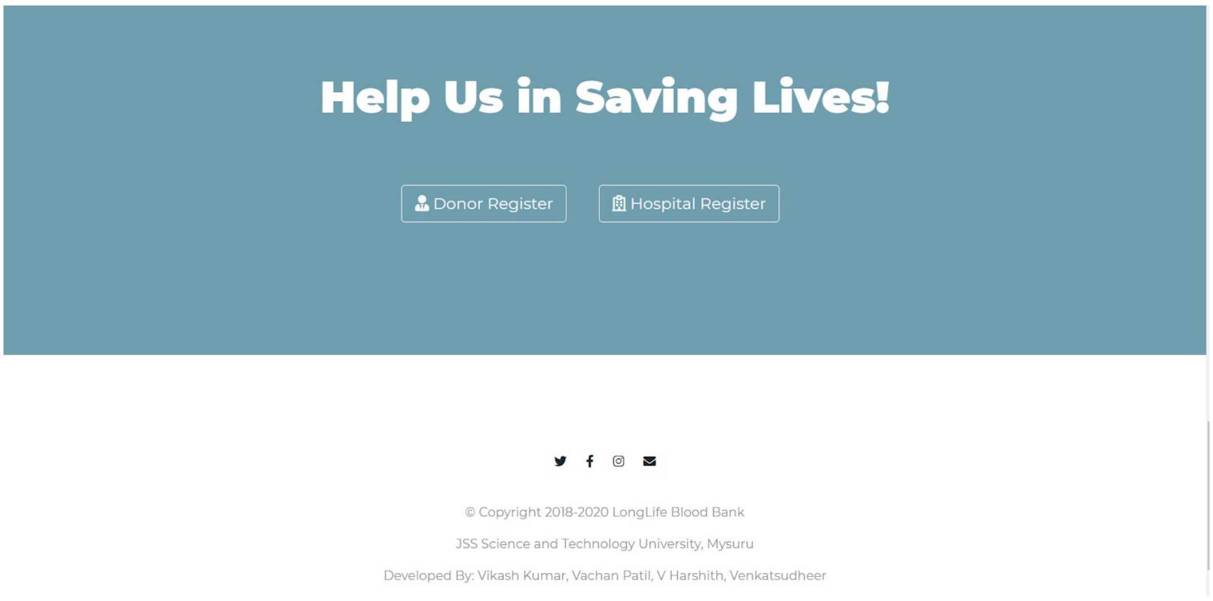
Best Practices
We ensure top level services and practises.

LongLife Blood Bank is the best blood bank in town. Its clean and hygenic facility is also worth praising.

Sonal, Bengaluru

Blood when you need it.

Our motto is to help as many people as we can through continuous support and best in class services.
We aim to become one of the best blood banks in the state by 2022.

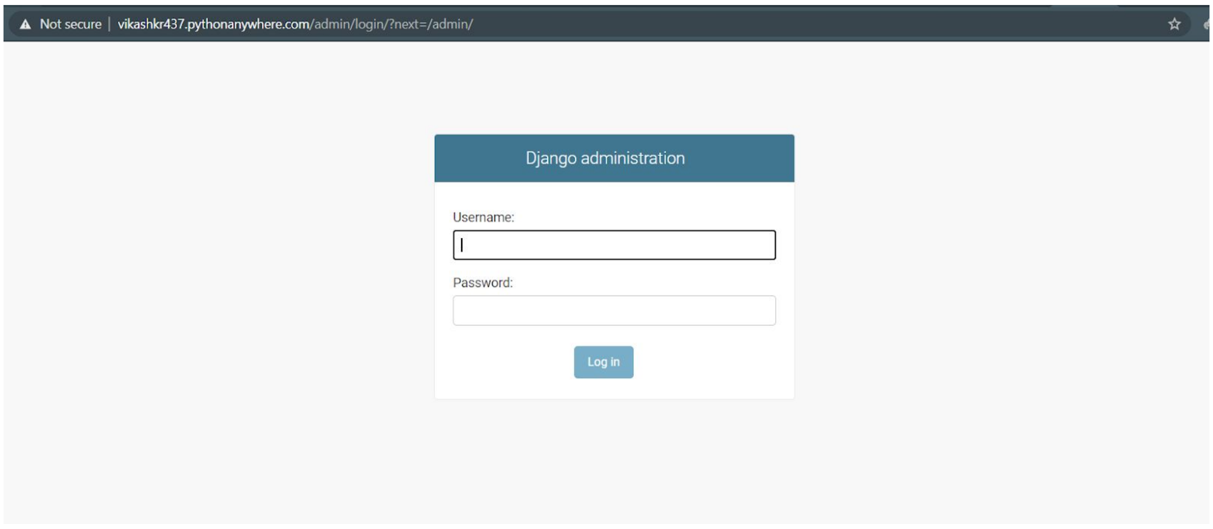


© Copyright 2018-2020 LongLife Blood Bank

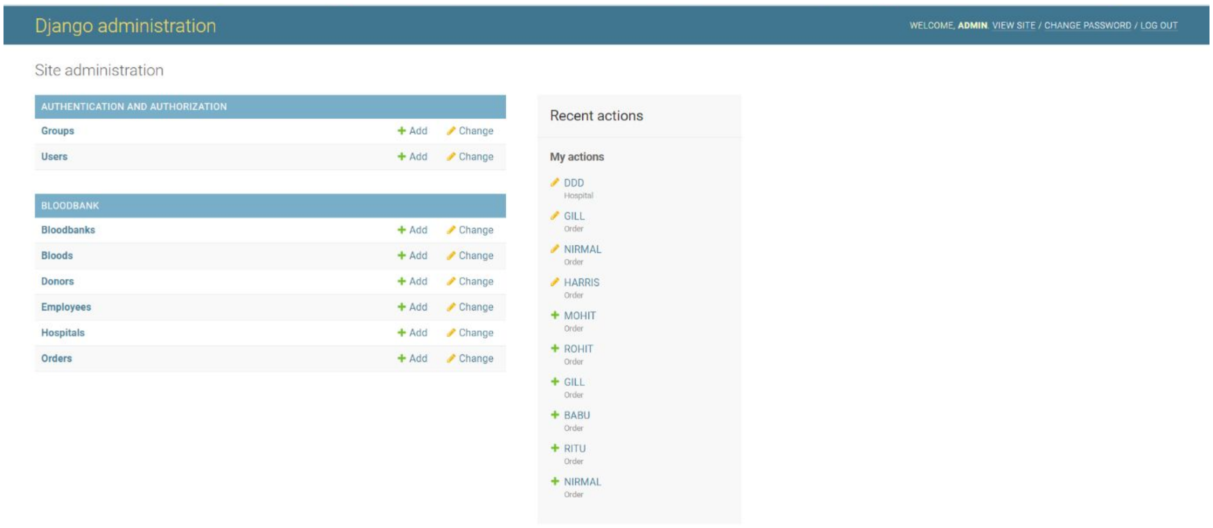
JSS Science and Technology University, Mysuru

Developed By: Vikash Kumar, Vachan Patil, V Harshith, Venkatsudheer

10.2 Admin login page:


Not secure | vikashkr437.pythonanywhere.com/admin/login/?next=/admin/
Django administration
Username:
Password:
Log in

After Login :



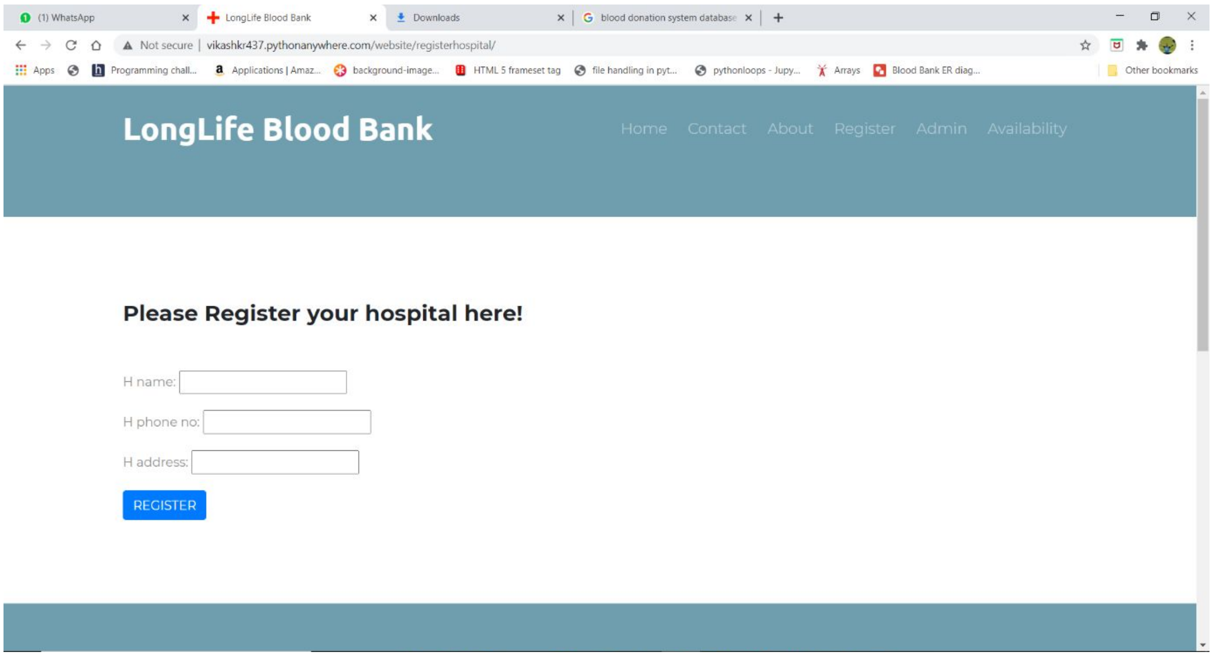
AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	Change
Users	+ Add	Change

BLOODBANK		
Bloodbanks	+ Add	Change
Bloods	+ Add	Change
Donors	+ Add	Change
Employees	+ Add	Change
Hospitals	+ Add	Change
Orders	+ Add	Change

My actions

- DDD Hospital
- GILL Order
- NIRMAL Order
- HARRIS Order
- MOHIT Order
- ROHIT Order
- GILL Order
- BABU Order
- RITU Order
- NIRMAL Order

10.3 Hospital Registration



(1) WhatsApp
LongLife Blood Bank
Downloads
blood donation system database
Not secure | vikashkr437.pythonanywhere.com/website/registerhospital/
LongLife Blood Bank
Home Contact About Register Admin Availability

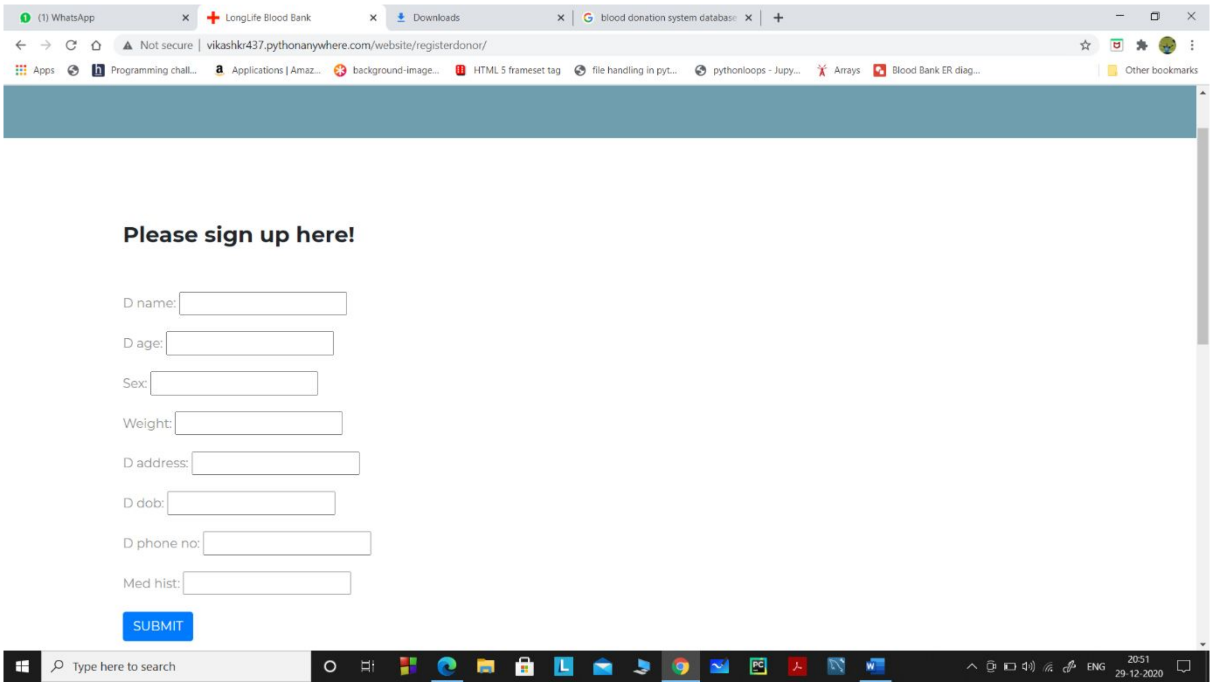
Please Register your hospital here!

H name:

H phone no:

H address:

10.4 Donor Registration Page



(1) WhatsApp
LongLife Blood Bank
Downloads
blood donation system database
Not secure | vikashkr437.pythonanywhere.com/website/registerdonor/

Please sign up here!

D name:

D age:

Sex:

Weight:

D address:

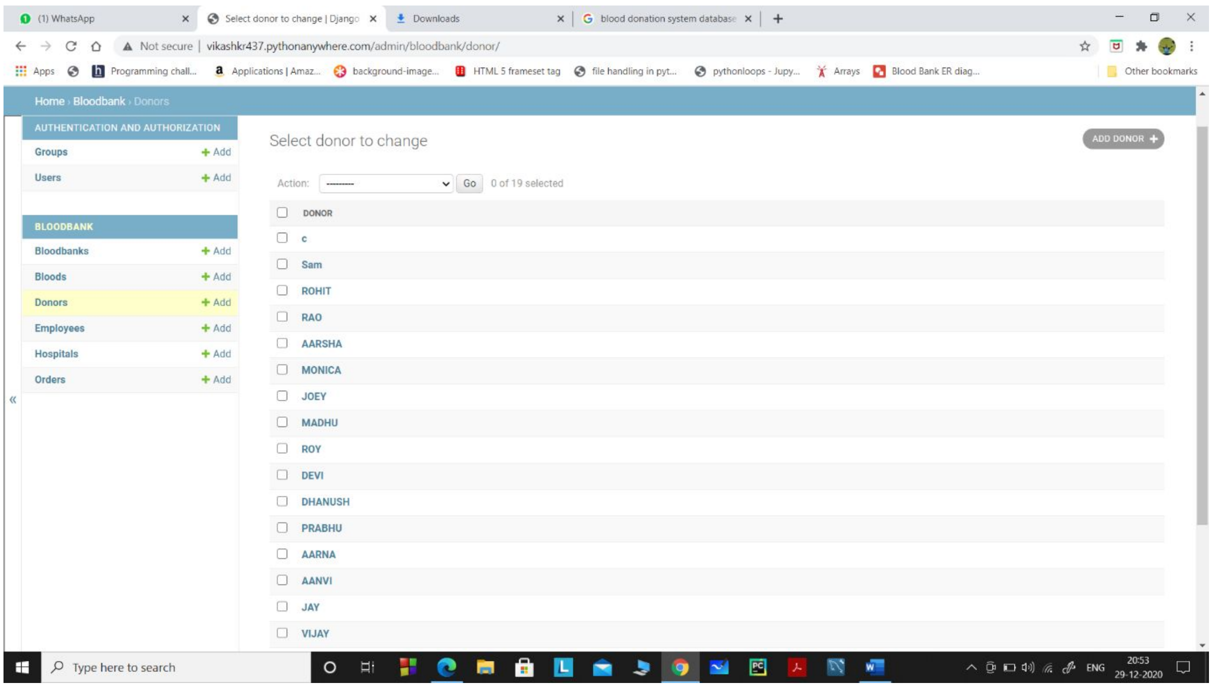
D dob:

D phone no:

Med hist:

SUBMIT
Type here to search
20:51
29-12-2020

10.5 Donor database(after admin login)



vikashkr437.pythonanywhere.com/admin/bloodbank/donor/

Home › Bloodbank › Donors

AUTHENTICATION AND AUTHORIZATION

- Groups + Add
- Users + Add

BLOODBANK

- Bloodbanks + Add
- Bloods + Add
- Donors + Add
- Employees + Add
- Hospitals + Add
- Orders + Add

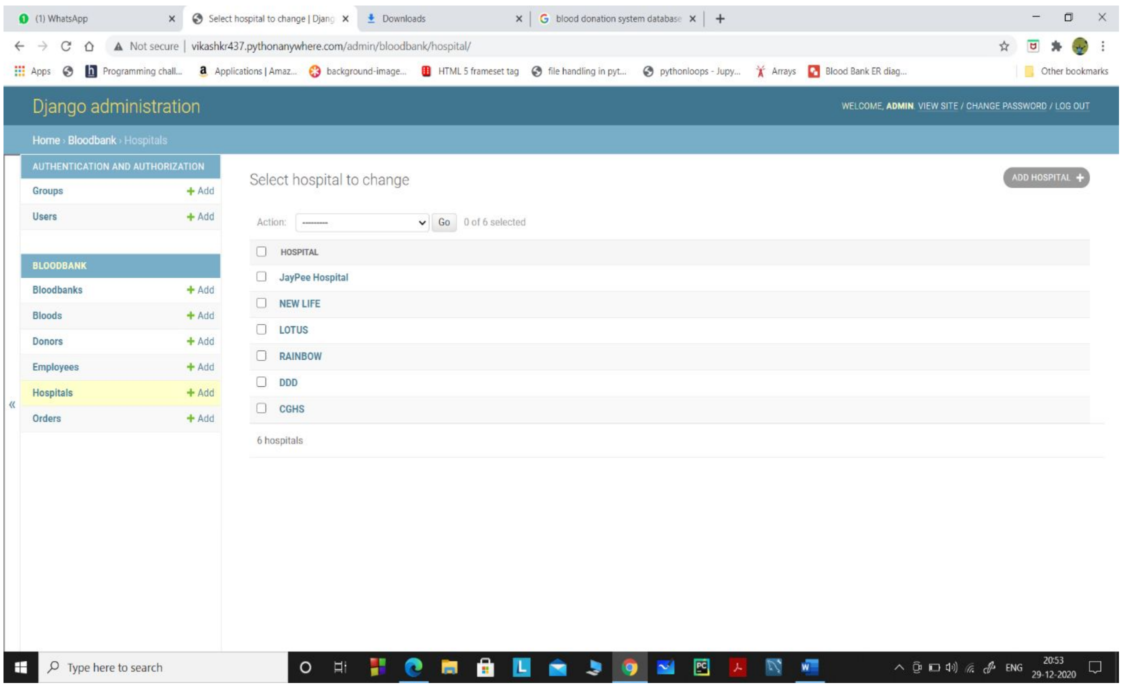
Select donor to change

ADD DONOR +

Action: --------- Go 0 of 19 selected

DONOR
c
Sam
ROHIT
RAO
AARSHA
MONICA
JOEY
MADHU
ROY
DEVI
DHANUSH
PRABHU
AARNA
AANVI
JAY
VIJAY

10.6 Hospital database (after admin login)



vikashkr437.pythonanywhere.com/admin/bloodbank/hospital/

Django administration

WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Bloodbank › Hospitals

AUTHENTICATION AND AUTHORIZATION

- Groups + Add
- Users + Add

BLOODBANK

- Bloodbanks + Add
- Bloods + Add
- Donors + Add
- Employees + Add
- Hospitals + Add
- Orders + Add

Select hospital to change

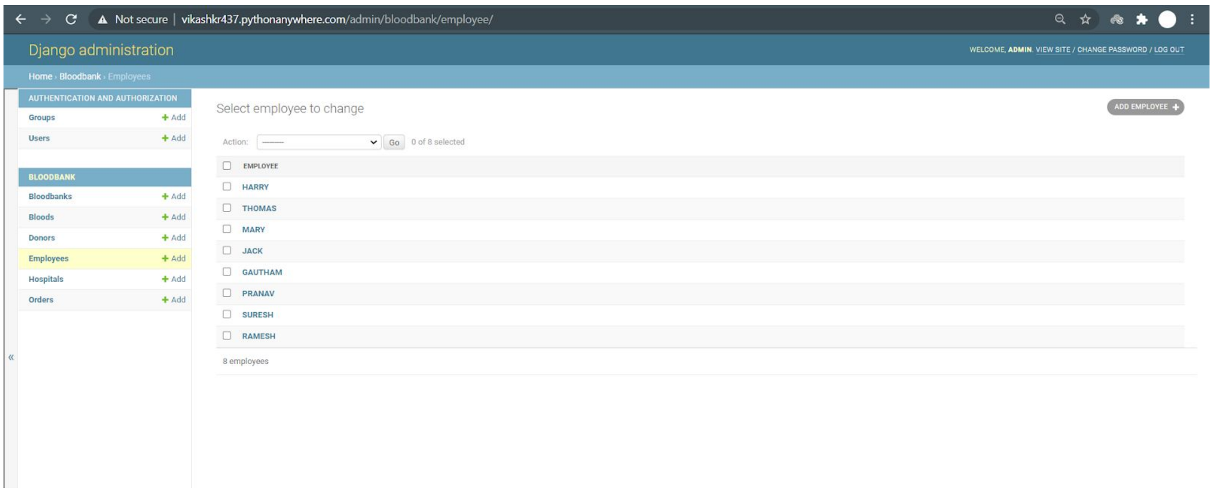
ADD HOSPITAL +

Action: --------- Go 0 of 6 selected

HOSPITAL
JayPee Hospital
NEW LIFE
LOTUS
RAINBOW
DDD
CGHS

6 hospitals

10.7 Employee database (after admin login)



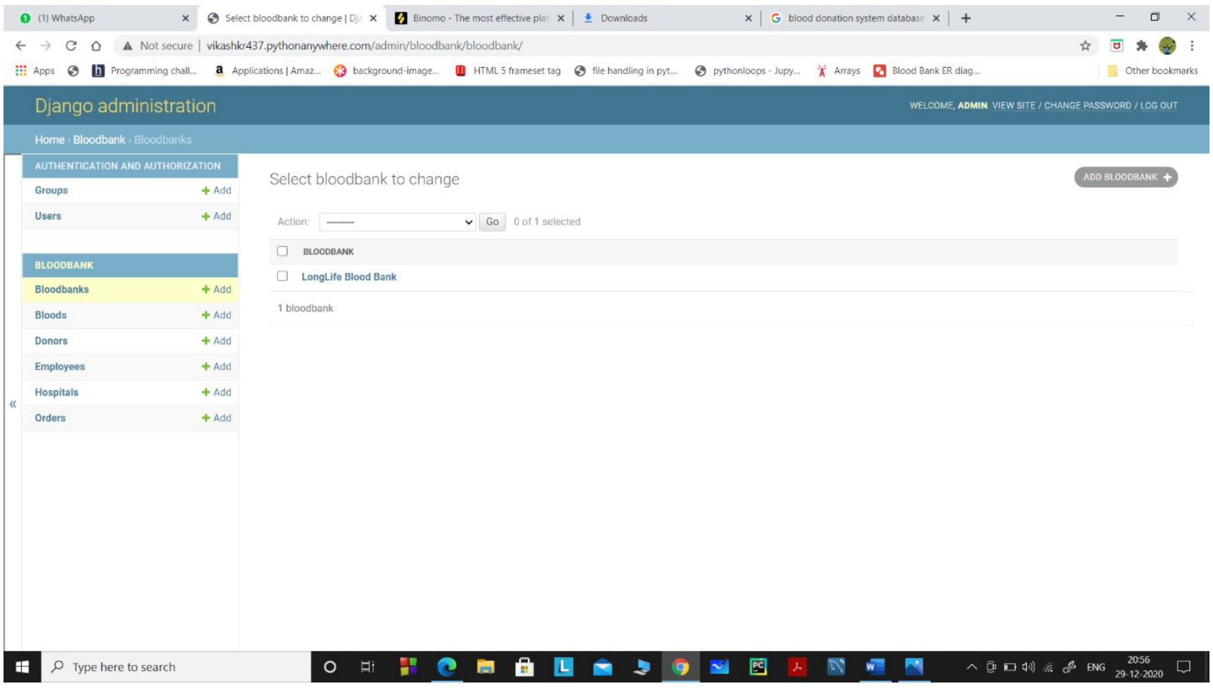
Select employee to change

Action: ——— Go 0 of 8 selected

EMPLOYEE
HARRY
THOMAS
MARY
JACK
GAUTHAM
PRANAV
SURESH
RAMESH

8 employees

10.8 Blood bank(after admin login)



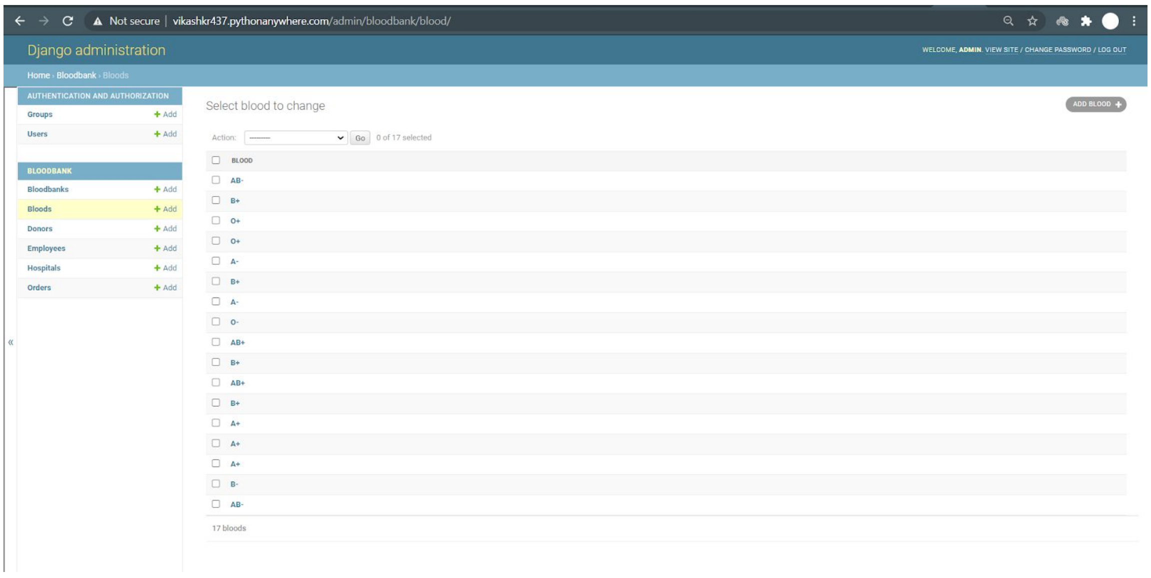
Select bloodbank to change

Action: --------- Go 0 of 1 selected

BLOODBANK
LongLife Blood Bank

1 bloodbank

10.9 Blood Database (after Admin login)



Not secure | vikashkr437.pythonanywhere.com/admin/bloodbank/blood/

Django administration

WELCOME, **ADMIN**. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Bloodbank › Bloods

AUTHENTICATION AND AUTHORIZATION	
Groups	+ Add
Users	+ Add

BLOODBANK	
Bloodbanks	+ Add
Bloods	+ Add
Donors	+ Add
Employees	+ Add
Hospitals	+ Add
Orders	+ Add

Select blood to change

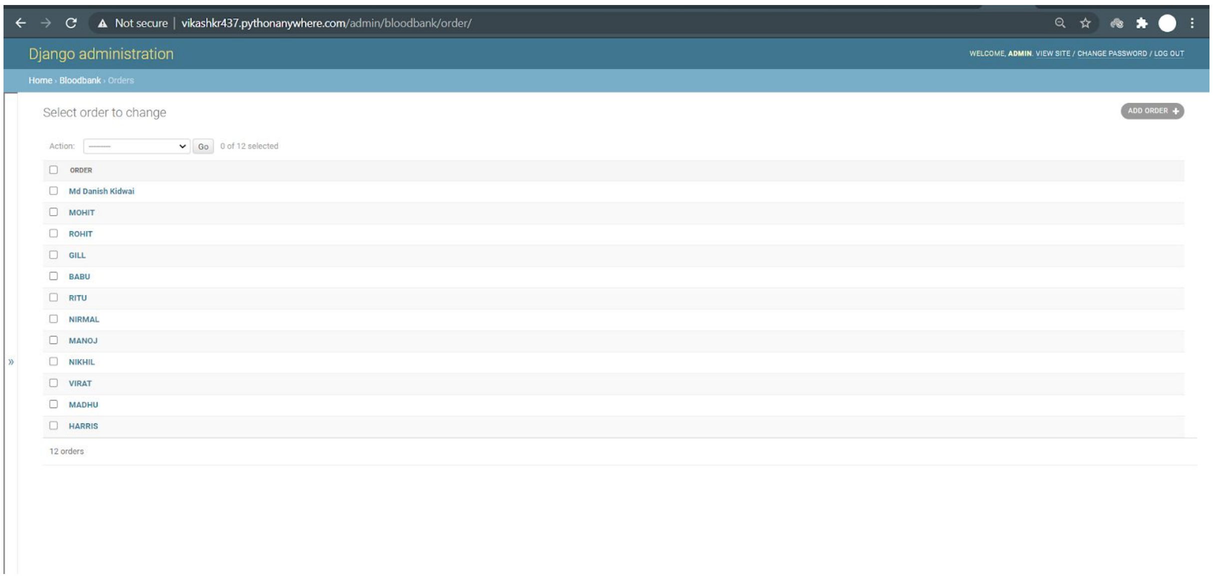
ADD BLOOD +

Action: --------- Go 0 of 17 selected

BLOOD
AB-
B+
O+
O+
A-
B+
A-
O-
AB+
B+
AB+
B+
A+
A+
A+
B-
AB-

17 bloods

10.10 Orders Database(after admin login)



Not secure | vikashkr437.pythonanywhere.com/admin/bloodbank/order/

Django administration

WELCOME, **ADMIN**. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Bloodbank › Orders

Select order to change

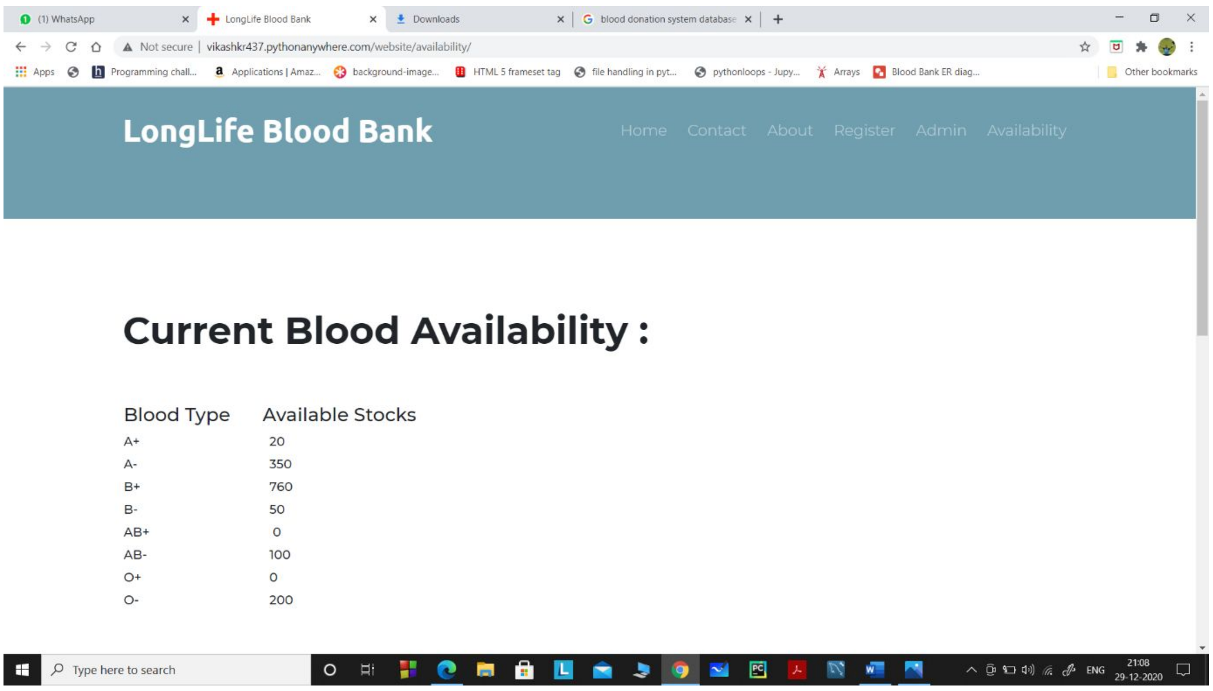
ADD ORDER +

Action: --------- Go 0 of 12 selected

ORDER
Md Danish Kidwai
MOHIT
ROHIT
GILL
BABU
RITU
NIRMAL
MANOJ
NIKHIL
VIRAT
MADHU
HARRIS

12 orders

10.11 Current blood availability:



Current Blood Availability :

Blood Type	Available Stocks
A+	20
A-	350
B+	760
B-	50
AB+	0
AB-	100
O+	0
O-	200

11. Result analysis

All the functional and non-functional requirements were taken into consideration and a web-page was created and viewed in localhost. The database was normalized to BCNF form. The webpage was created using HTML, CSS and Bootstrap at front end, MySql at backend and Flask as a middleware that was used to connect to the backend. Insert, delete and modify options have been provided through the user Interface itself and easy navigation from one screen to another has been provided. Aggregate functions such as group by, order by has also been used. A trigger for checking the medical history and option to delete it has also been given. All the requirements have been met and tested. Various test cases were developed to check the functionality of our web-page and all the testcases were provided with appropriate output.

12. Conclusion

The project has greatly enhanced the way in which a blood bank can work through a web site. Such apps are the need of the hour as the entire world is at the highest pace for internet adoption and here in India too it will prove to be very beneficial and worth trying as it minimises errors, makes operations transparent, gives an easy to use interactive interface along with a great UI. This should really help in making people aware of the importance of blood and blood banks.While doing the project we got to know a lot about the internal workings of the blood bank.

This project gave us a great opportunity for a web-based project to design, code, measure and execute. This has helped to implement the different software engineering and database management principles Concepts such as data integrity and continuity. This has also helped me find out more about PythonAnywhere, Django, HTML, CSS and SQLite and Personal Web Server.

13. References

I. Fundamentals Of Database Systems-7th edition by Ramez Elmasri Shamkant B.Navathe
II. https://docs.djangoproject.com/en/3.1/
III. https://www.djangoproject.com/start/
IV. https://www.w3schools.com/
V. https://www.overleaf.com/project
VI. https://tex.stackexchange.com