

BLOOD BANK MANAGEMENT SYSTEM

A Report Submitted

In partial fulfillment of the requirements

for the degree of

Master of Computer Application

Group No. 16

Submitted by

Nikshep Paliwal	(21736)
Ravi Kant	(21752)
Pankaj Kumar	(21738)
Priyanshu Subberwal	(21744)

Under the Guidance of

Mrs. Mekhla Rai
Miss. Neetika Gond



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
KAMLA NEHRU INSTITUTE OF TECHNOLOGY, SULTANPUR – 228118(U.P.)
(An autonomous Government Engineering Institute under 2f & 12B of UGC Act)

Affiliated to

**KAMLA NEHRU INSTITUTE OF TECHNOLOGY, SULTANPUR –
228118(U.P.)**

**(An autonomous Government Engineering Institute under 2f & 12B of
UGC Act)**

Affiliated to

Dr. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY



CERTIFICATE

This is to certify that the project “**Blood Bank Management System**” is being submitted by **Nikshap Paliwal (21736), Pankaj Kumar (21738), Ravi Kant (21752), Priyanshu Subberwal (21744)**, is partial fulfillment for the degree of Master of Computer Applications, original work carried out by himself. The matter in this project is genuine work done by the student and has been submitted to this institute or any other institute for the fulfillment of the requirement of any course of the study. The project is developed by the students of **KAMLA NEHRU INSTITUTE OF TECHNOLOGY, SULTANPUR – 228118(U.P.) (An autonomous Government Engineering Institute under 2f & 12B of UGC Act) Affiliated to Dr. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY** is record of their own work carried out under my supervision.

Mrs. Mekhla Rai

Miss Neetika Gond

(Mini-project in-charge)

Department of Computer Science

KNIT, Sultanpur

Dr. Awadhesh Kumar

(Head of Department)

Department of Computer Science

KNIT, Sultanpur

Acknowledgement

The success of this project can be attributed to the help of the number of people. Firstly, we would like to thank out supervisor Mrs. Mekhla Rai for their support and technical advice during the course of the project. We are also thankful to Miss. Neetika Gond for his valuable suggestion.

I thank the people who were a part of this project in numerous ways, people who gave their unending support right from the stage the project idea was conceived.

The four things that go on to make a successful endeavor are dedication, hard work, patience and correct guidance. I would like to thank our principle who has always been the source of inspiration.

We are thankful to other lab supervisors, our batch mate and juniors who helped us a lot implementation phase of our project.

A warms thanks also goes to who had provided support during the challenges and understanding of our many hours spent in college.

Last but not the least I would like to thank all our friends and family for their support, and all others who have contributed to the completion of this project directly or indirectly.

Nikshap Paliwal (21736)

Ravi Kant (21752)

Pankaj Kumar (21738)

Priyanshu Subberwal (21744)

ABSTRACT

Blood transfusion safety is a relevant and significant public health issue in India. Since most blood banks are still in paper-based system, various disadvantages are experienced by various stakeholders, which endanger the lives of patients and deter the healthcare system. As such, the researchers aimed to design, develop, and implement an online blood bank management system (OBBMS). This web-based application allows hospitals in India to make inventories of their blood bags online, subsequently, allowing each hospital to check the availability of blood bags anytime. The researchers designed and administered a questionnaire that assess the perceptions of various stakeholders in both manual-based and OBBMS. Based on the findings and results, it was found out that these stakeholders perceived online blood bank management system is much better than the manual system. Therefore, with the use of online blood bank management system, blood transfusion process is safe and secured. Threats on improper blood donor documentation, or misplaced records will be totally eradicated. Also, processes involving recording about blood donors, blood bag collection, storage, and inventory will be systematized and organized, hence, improving the healthcare management for blood banks.

CONTENTS	PAGENO.
Declaration	01
Certificate	02
Acknowledgement	03
Abstract	04
1: Introduction	
1.1 Problem Statement	07
1.2 Scope	07
1.3 Limitation	07
2: Objective	08
3: System Requirement	
3.1 Software	08
3.2 hardware	08
4: Theoretical Background	
4.1 HTML5	09
4.2 CSS3	09
4.3 Java Script	09
4.4 NodeJS	10
4.5 MongoDB	10
5: Used model	11
6: Diagram	
6.1 0 level diagram	13
6.2 1 level diagram	14
6.3 E R Diagram	15
6.4 Flow Chart	16
6.5 Use Case Diagram	17
7: Design and Development	18
8: Reference	22

TABLE OF FIGURES

Serial No.	<u>Figure Name</u>	Page No.
01	Fig.5(a) Evolutionary Model	11
02	Fig.-6.1 Level 0 DFD	13
03	Fig.6.2 Level 1 DFD	14
04	Fig.-6.3 ER Diagram	15
05	Fig.-6.4 Flow Chart	16
06	Fig.-6.5 Use Case Diagram	17
07	Fig.-7.1 Home Page	18
08	Fig.-7.2 Blood Bank Login Page	18
09	Fig.-7.3 Blood Bank Registration Form	19
10	Fig.-7.4 Donor Login Page	19
11	Fig.-7.5 Donor Registration Page	20
12	<u>Fig.-7.6</u> Query Running	21
13	Fig.-7.7 Database Stored	21

1. INTRODUCTION

1.1 PROBLEM STATEMENT

Despite advances in technology, nowadays, most blood bank systems are running in manual system. As such, there is a prevalent problem in the availability of needed blood types. For instance, when a person needs a certain type of blood and this type is not available in the hospital, family members send messages through social media to those who can donate to them and this process takes longer than the life of the patient to the most dangerous. In addition, it seems that there is lack of proper documentation about blood donors and its medical history. This may lead to blood bag contamination and may affect the blood transfusion safety. Generally, this study aims to determine how the use of online bank management system enhance blood transfusion safety. Subsequently, this study seeks to answer the following specific problems:

1. What is the level of perception among blood bank's stakeholders on manual-based system?
2. What is the level of perception among blood bank's stakeholders on online blood bank management system?
3. H0: Is there no significant difference in the level of perception among stakeholders between manual-based and online-based blood bank system?

H1: Is there a significant difference in the level of perception among stakeholders between manual-based and online-based blood bank system?

1.2 SCOPE

This research study covers the three basic operations of blood banks, namely: donor registration, monitoring of blood bags or products' inventories, and monitoring of blood bags or products' issuance. Also, due to time-constraint, respondents will be from hospitals from North Region of India, though the research study talks about blood banks in the India. In addition, the study considers three possible users of the system, namely: administrator, donor, and blood receptionists.

1.3LIMITATION

This research study does not cover the actual blood collection activity, and actual blood transfusion operation. Blood donors and patients or recipients of blood donation are not system users, their registration or information will be encoded by the blood bank receptionists.

2.Objective

This applied research aims to design, develop and implement online blood bank management system. This web-based application provides:

- To ensure hospital to have good supply or inventories of blood bags.
- To check the availability of blood bags anytime.
- To manage the information of its blood donor.
- Function to check if the person donate blood for the last 3 months.
- To allow good documentation about the donor and its blood donation activities.
- Support fast searching to find match blood bags for the right person.

3. SYSTEM REQUIREMENT

3.1 The Software used:

- Language used in **front-end**: HTML5, CSS3 and Java Script.
- Language used in **back-end**: NodeJS, ExpressJS and MongoDB.
- MS-word for documentation.
- Chrome Browser to execute the website processing.

3.2 The Hardware used:

While developing the project, the used hardware is:

- Operating system: windows 7 and higher version.
- Ram: 4GB.
- Processor: 1Ghz or faster processor.

4: THEORATICAL BACKGROUND

4.1 HTML:

HTML (Hyper Text Markup Language) is a markup language that tells web browsers how to structure the web pages you visit. It can be as complicated or as simple as the web developer wants it to be. HTML consists of a series of elements, which you use to enclose, wrap, or markup different parts of content to make it appear or act in a certain way. Web browsers can read HTML files and render them into visible or audible web pages. Browsers do not display the HTML tags and scripts, but use them to interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language, rather than a programming language.

The enclosing tags can make content into a hyperlink to connect to another page, italicize words, and so on. For example, consider the following line of text:

My dog is very loyal.

If we wanted the text to stand by itself, we could specify that it is a paragraph by enclosing it in a paragraph (<p>) element.

<p>My dog is very loyal. </p>

4.2 CSS:

Cascading Style Sheets (CSS) is a style sheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

CSS is among the core languages of the open web and is standardized across Web browsers according to W3C specifications. Previously, the development of various parts of CSS specification was done synchronously, which allowed the versioning of the latest recommendations. You might have heard about CSS1, CSS2.1, or even CSS3. There will never be a CSS3 or a CSS4; rather, everything is now CSS without a version number.

After CSS 2.1, the scope of the specification increased significantly and the progress on different CSS modules started to differ so much, that it became more effective to develop and release recommendations separately per module. Instead of versioning the CSS specification, W3C now periodically takes a snapshot of the latest stable state of the CSS specification and individual modules progress. CSS modules now have version numbers, or levels, such as CSS Color Module Level 5.

4.3 JavaScript:

JavaScript (often shortened to JS) is a lightweight, interpreted, object-oriented language with first-class functions, and is best known as the scripting language for Web pages, but it's used in many non-browser environments as well. It is a prototype-based, multi-paradigm scripting language that is dynamic, and supports object-oriented, imperative, and functional programming styles.

JavaScript runs on the client side of the web, which can be used to design / program how the web pages behave on the occurrence of an event. JavaScript is an easy to learn and also powerful scripting language, widely used for controlling web page behavior.

Contrary to popular misconception, JavaScript is not "Interpreted Java". In a nutshell, JavaScript is a dynamic scripting language supporting prototype-based object construction. The basic syntax is intentionally similar to both Java and C++ to reduce the number of new concepts required to learn the language. Language constructs, such as if statements, for and while loops, and switch and try ... catch blocks function the same as in these languages (or nearly so).

JavaScript can function as both a procedural and an object-oriented language. Objects are created programmatically in JavaScript, by attaching methods and properties to otherwise empty objects at run time, as opposed to the syntactic class definitions common in compiled languages like C++ and Java. Once an object has been constructed it can be used as a blueprint (or prototype) for creating similar objects.

JavaScript's dynamic capabilities include runtime object construction, variable parameter lists, function variables, dynamic script creation (via eval), object introspection (via for ... in), and source code recovery (JavaScript programs can decompile function bodies back into their source text).

4.4 NodeJS:

Node (or more formally Node.js) is an open-source, cross-platform runtime environment that allows developers to create all kinds of server-side tools and applications in JavaScript. The runtime is intended for use outside of a browser context (i.e., running directly on a computer or server OS). As such, the environment omits browser-specific JavaScript APIs and adds support for more traditional OS APIs including HTTP and file system libraries.

From a web server development perspective Node has a number of benefits:

Great performance! Node was designed to optimize throughput and scalability in web applications and is a good solution for many common web-development problems (e.g., real-time web applications).

Code is written in "plain old JavaScript", which means that less time is spent dealing with "context shift" between languages when you're writing both client-side and server-side code.

JavaScript is a relatively new programming language and benefits from improvements in language design when compared to other traditional web-server languages (e.g., Python, PHP, etc.) Many other new and popular languages compile/convert into JavaScript so you can also use TypeScript, CoffeeScript, ClojureScript, Scala, LiveScript, etc.

The node package manager (npm) provides access to hundreds of thousands of reusable packages. It also has best-in-class dependency resolution and can also be used to automate most of the build toolchain.

Node.js is portable. It is available on Microsoft Windows, macOS, Linux, Solaris, FreeBSD, OpenBSD, WebOS, and NonStop OS. Furthermore, it is well-supported by many web hosting providers, that often provide specific infrastructure and documentation for hosting Node sites.

It has a very active third-party ecosystem and developer community, with lots of people who are willing to help.

4.5 MongoDB:

MongoDB is a type of NoSQL database that stores documents and collections. The documents are equivalent to records, and the collections are what we know as tables in SQL. So, MongoDB is a database

that runs on a server. We store application data there like our users, products, and etc. The MongoDB enforces no data schema or relations. So, the individual records can be structured in the way we want.

MongoDB is easily connected to node and express, and that is something great about MongoDB. We can easily add it to a node express backend, not directly to angular.

Overall, MongoDB is a powerful database that we can easily integrate into our node express environment.

We will use MongoDB here because the MEAN Stack is really popular, but we could theoretically swap it for any other database, e.g., MySQL.

5: MODEL USED

Evolutionary model is a combination of Iterative and Incremental model of software development life cycle. Delivering your system in a big bang release, delivering it in incremental process over time is the action done in this model. Some initial requirements and architecture envisioning need to be done. It is better for software products that have their feature sets redefined during development because of user feedback and other factors. The Evolutionary development model divides the development cycle into smaller, incremental waterfall models in which users are able to get access to the product at the end of each cycle. Feedback is provided by the users on the product for the planning stage of the next cycle and the development team responds, often by changing the product, plan or process. Therefore, the software product evolves with time. All the models have the disadvantage that the duration of time from start of the project to the delivery time of a solution is very high. Evolutionary model solves this problem in a different approach.

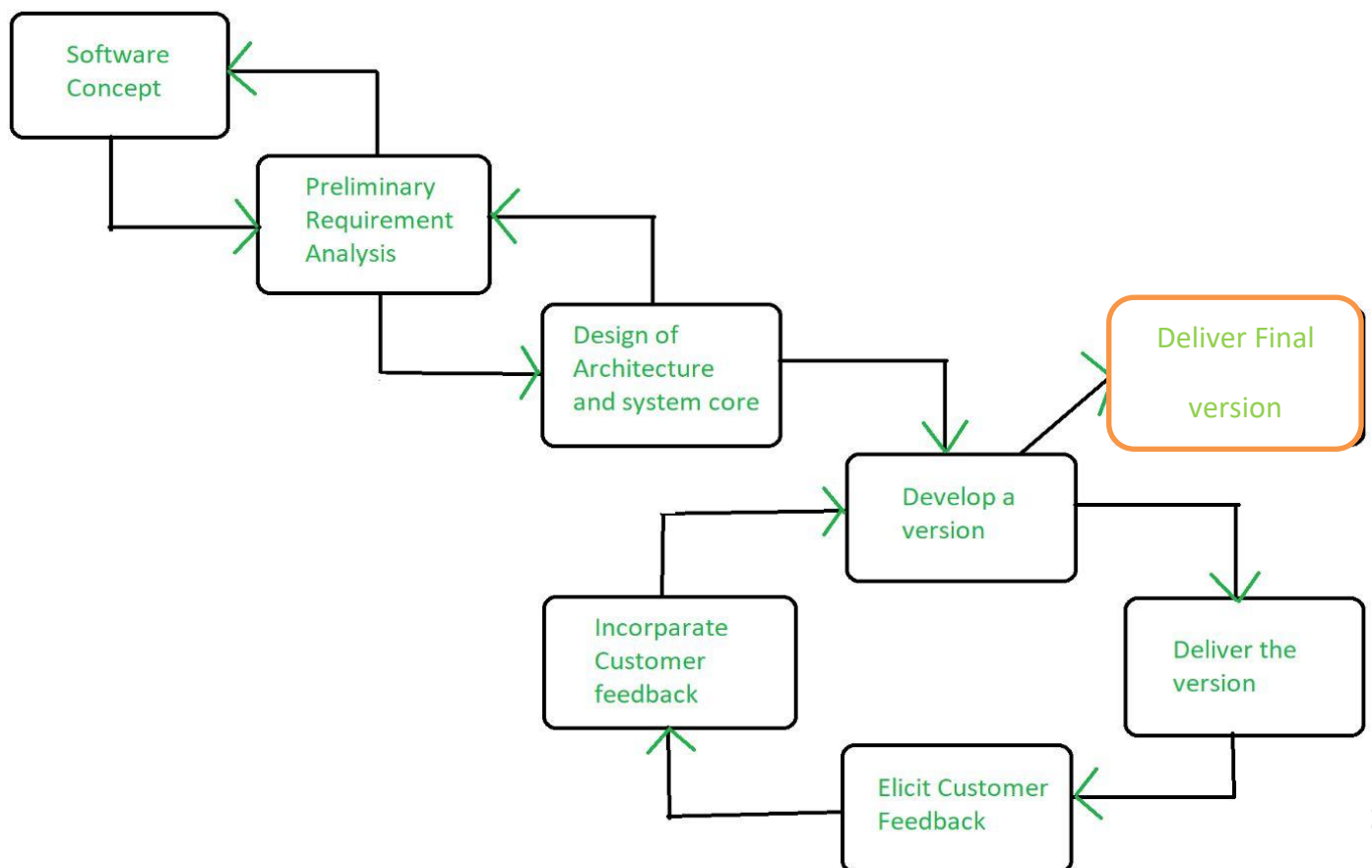


Fig.5(a) Evolutionary Model

Evolutionary model suggests breaking down of work into smaller chunks, prioritizing them and then delivering those chunks to the customer one by one. The number of chunks is huge and is the number of deliveries made to the customer. The main advantage is that the customer's confidence increases as he constantly gets quantifiable goods or services from the beginning of the project to verify and validate his requirements. The model allows for changing requirements as well as all work is broken down into maintainable work chunks. **Application of Evolutionary Model:**

1. It is used in large projects where you can easily find modules for incremental implementation. Evolutionary model is commonly used when the customer wants to start using the core features instead of waiting for the full software.
2. Evolutionary model is also used in object-oriented software development because the system can be easily portioned into units in terms of objects.

Necessary conditions for implementing this model: -

- Customer needs are clear and been explained in deep to the developer team.
- There might be small changes required in separate parts but not a major change.
- As it requires time, so there must be some time left for the market constraints.
- Risk is high and continuous targets to achieve and report to customer repeatedly.
- It is used when working on a technology is new and requires time to learn.

Advantages:

- In evolutionary model, a user gets a chance to experiment partially developed system.
- It reduces the error because the core modules get tested thoroughly.

Disadvantages:

- Sometimes it is hard to divide the problem into several versions that would be acceptable to the customer which can be incrementally implemented and delivered.

6: DIAGRAMS

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination.

Below is the DFD for our project.

6.1 LEVEL 0 DFD/CONTEXT DIAGRAM:

DFD Level 0 is also called a Context Diagram. It's a basic overview of the whole system or process being analyzed or modeled. It's designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities.

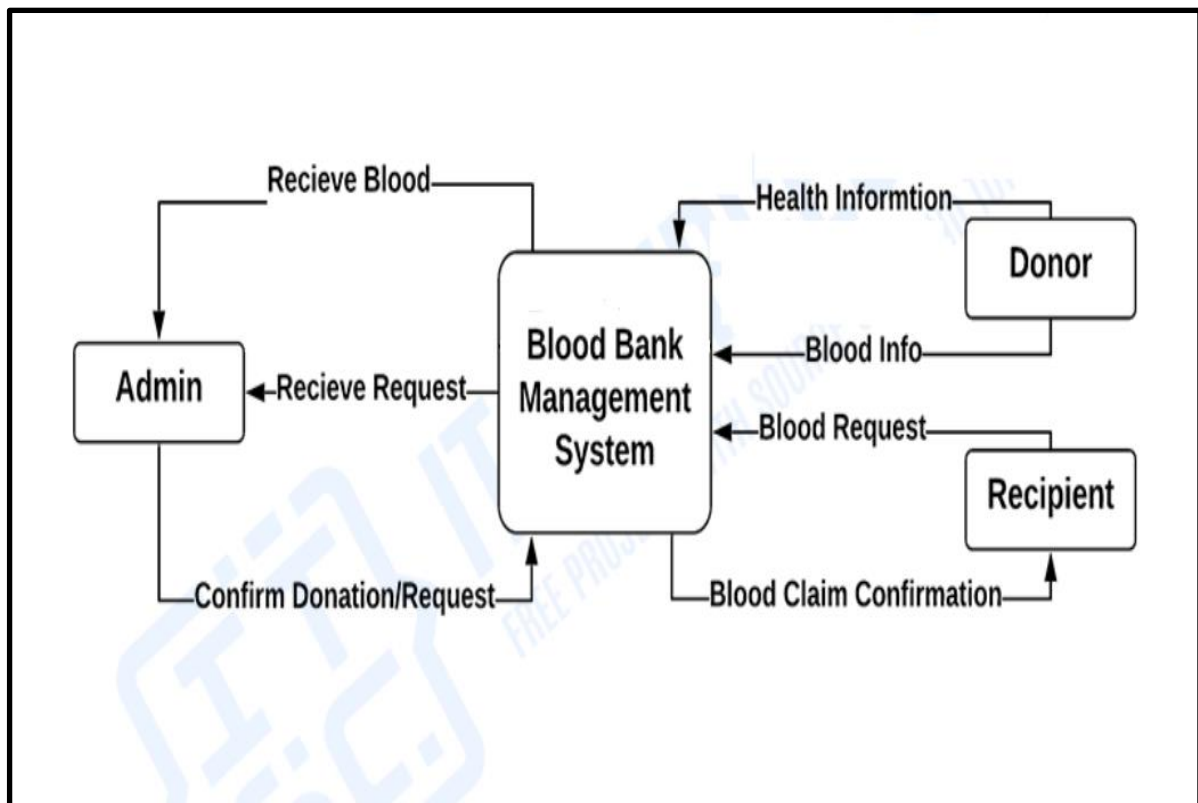


Fig.-6.1 Level 0 DFD

6.2 DFD level 1:

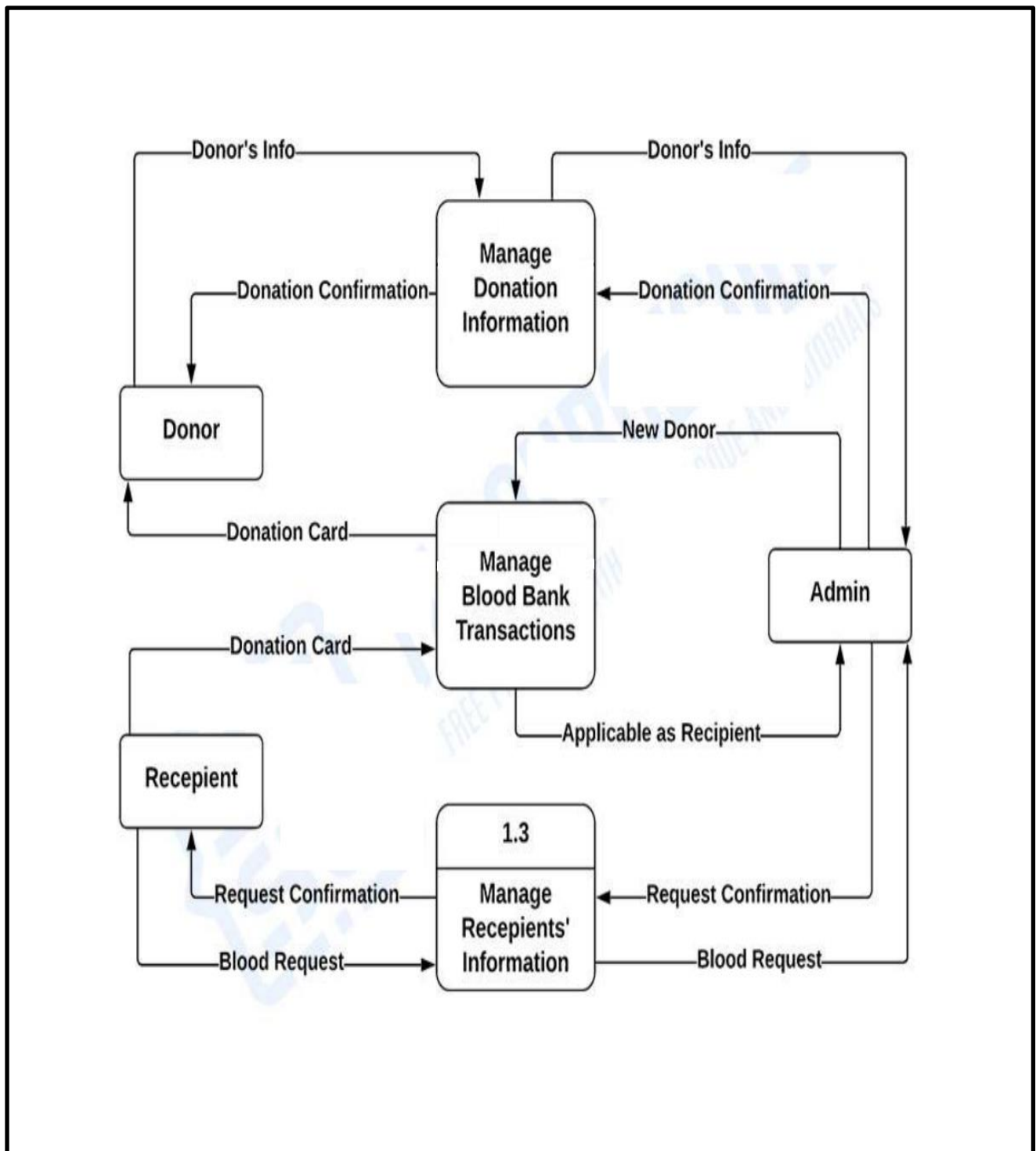


Fig.-6.2 Level 1 DFD

6.3: ER diagram

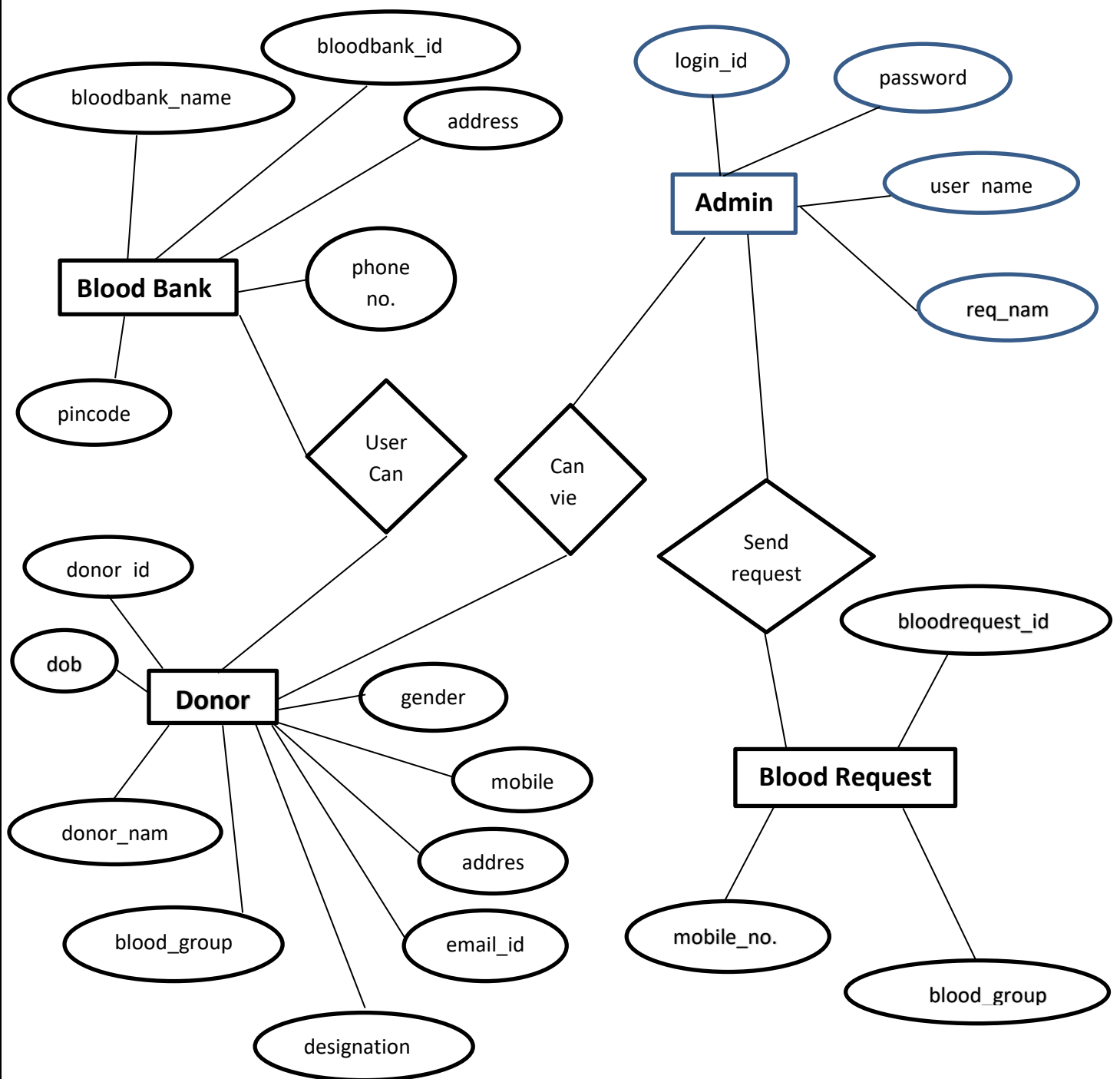


Fig.-6.3 ER Diagram

6.4: Flow chart

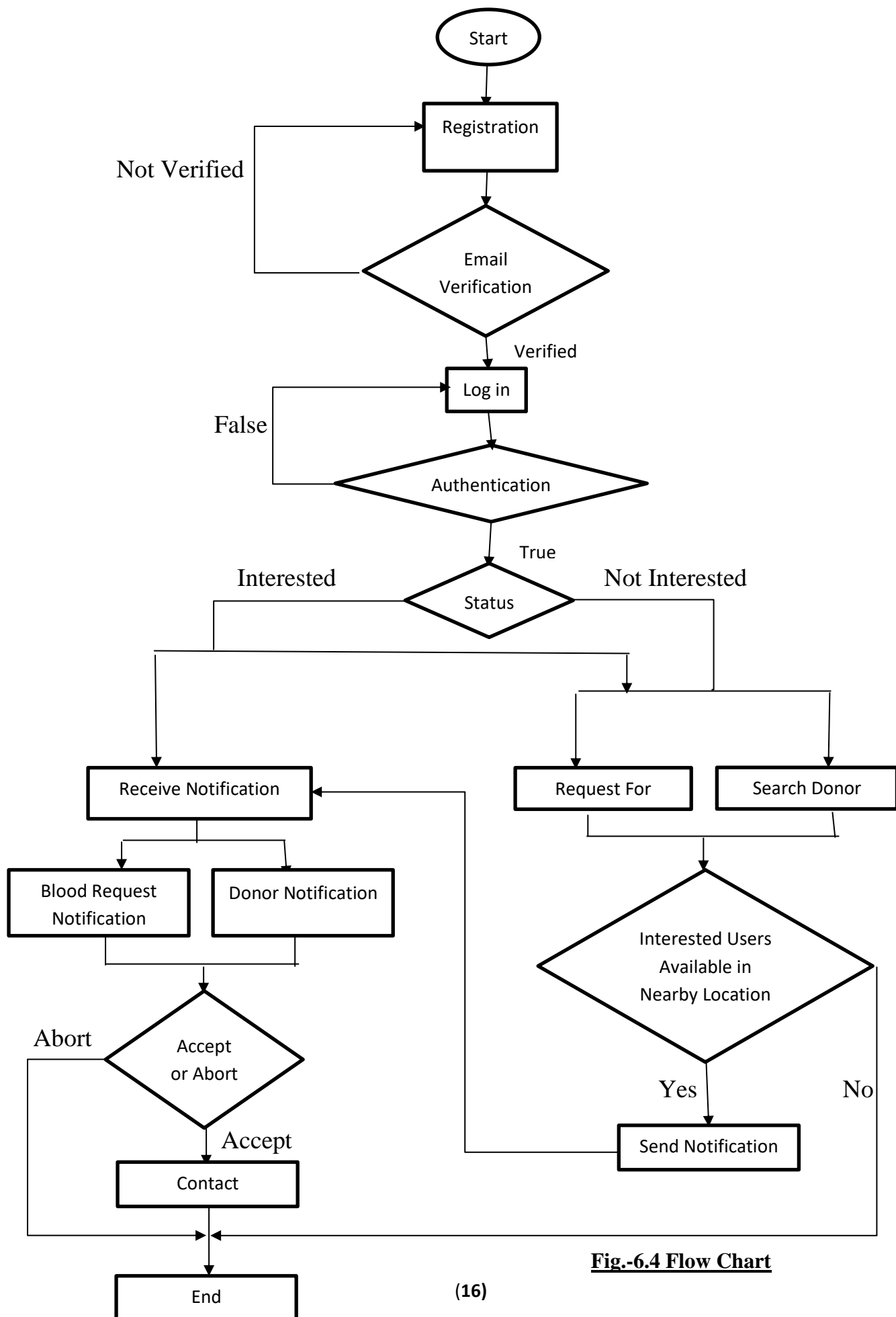


Fig.-6.4 Flow Chart

6.5 Use Case Diagram:

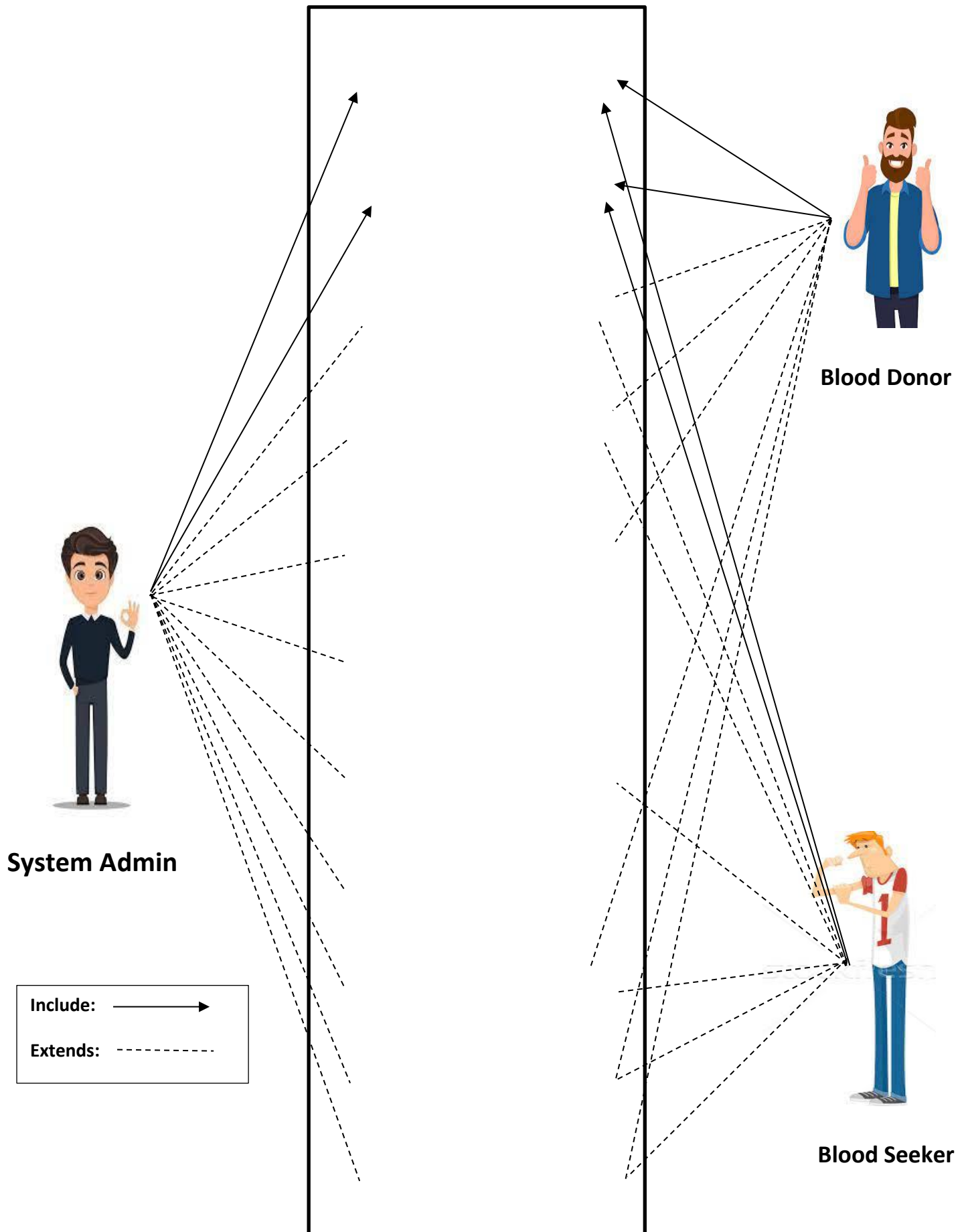


Fig.-6.5 Use Case Diagram

7. Design and development

7.1 Home Page:

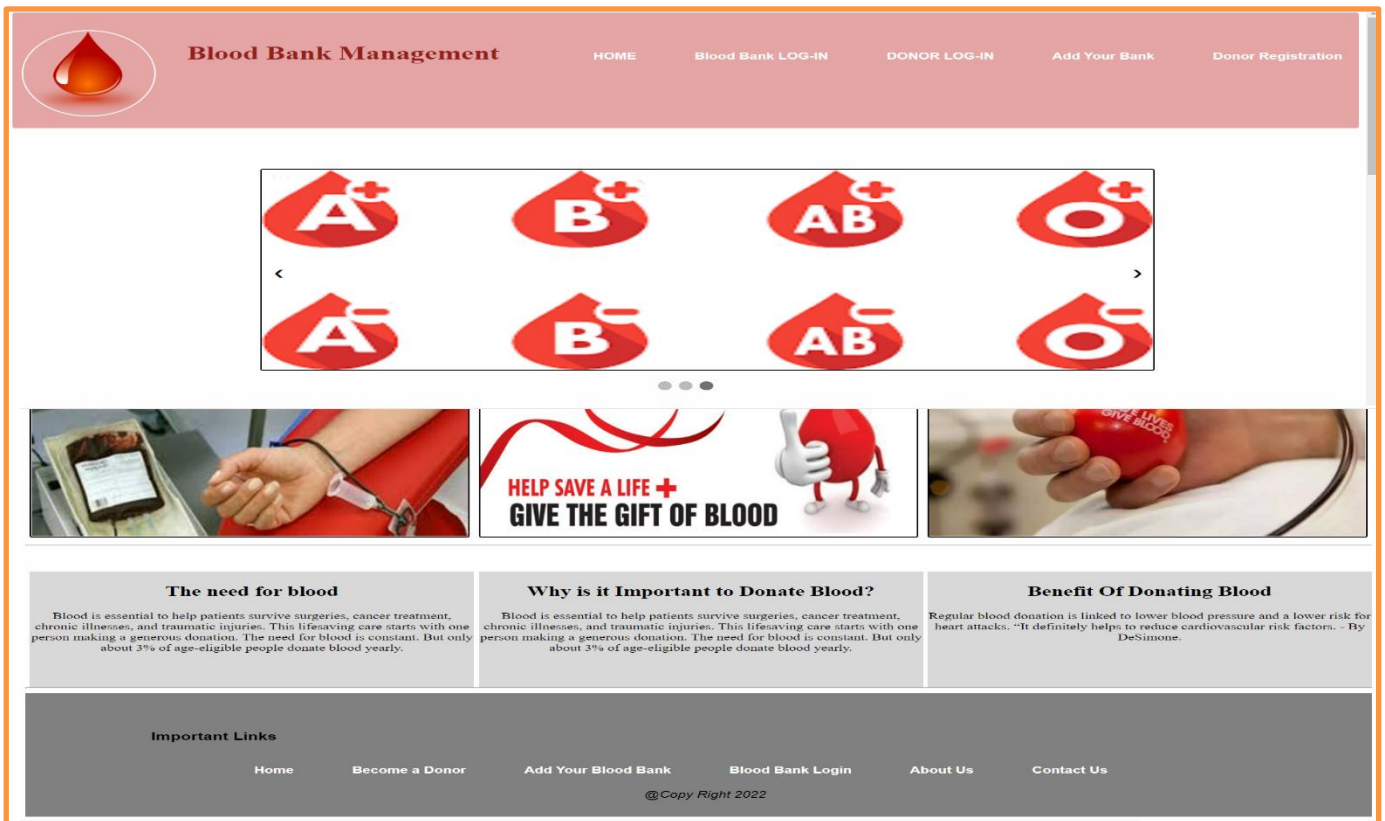


Fig.-7.1 Home Page

7.2 Blood Bank Login Page:

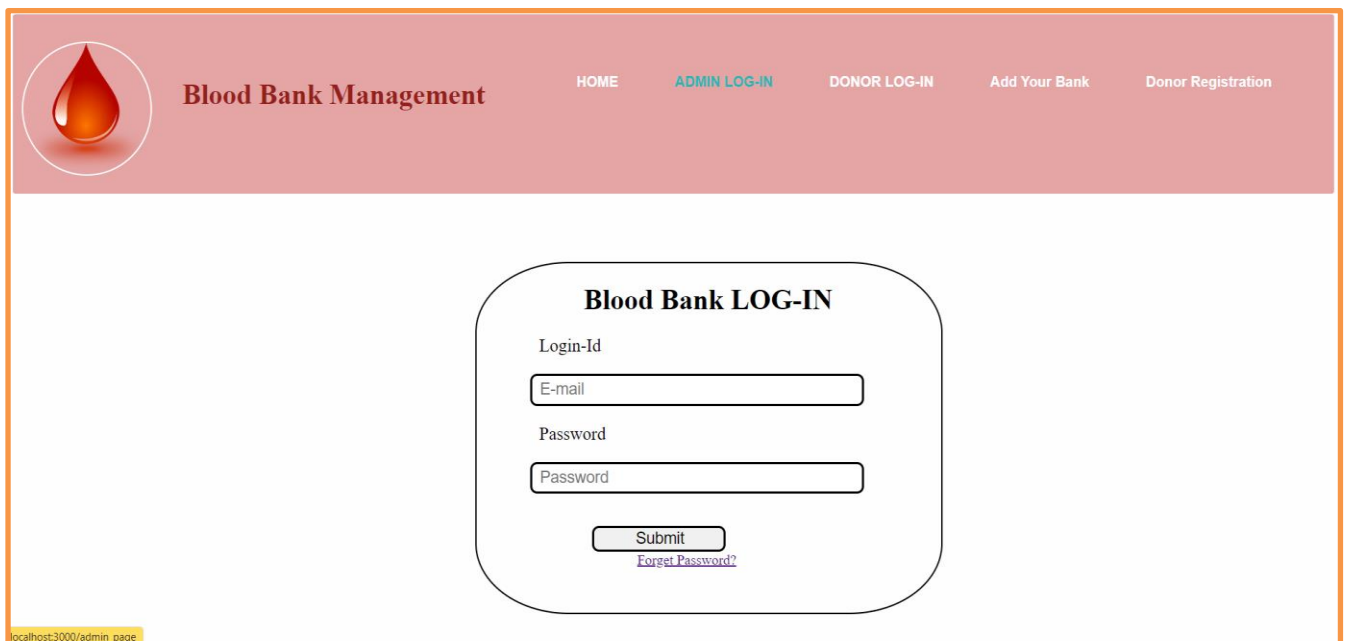


Fig.-7.2 Blood Bank Login Page

7.3 Blood Bank Registration Form:

Blood Bank Registration

Enter Blood Bank Name

KNI

Address

KNIT SULTANPUR

SULTANPUR

Enter your Pin Code

228118

Contact

1234567890

Email

ABC@GMAIL.COM

Create Password

.....

Registration

Fig.-7.3 Blood Bank Registration Form

7.4 Donor Login Page:

DONOR LOG-IN

Login-Id

E-mail

Password

Password

Submit

Forget Password?

Compatible Blood Type Donors		
Blood Type	Donate Blood To	Receive Blood From
A+	A+ AB+	A+ A- O+ O-
O+	O+ A+ B+ AB+	O+ O-
B+	B+ AB+	B+ B- O+ O-
AB+	AB+	Everyone
A-	A+ A- AB+ AB-	A- O-
O-	Everyone	O-
B-	B+ B- AB+ AB-	B- O-
AB-	AB+ AB-	AB- A- B- O-

Fig.-7.4 Donor Login Page

7.5 Donor Registration Page:

Donor Registration

Enter Your First Name:

Enter Your Middle Name:

Enter Your Last Name:

Date of Birth

Address

Enter your Pin Code

Contact

Gender

Blood Group

Email

Create Password

Registration

Fig.-7.5 Donor Registration Page

Fig.-7.5 Donor Registration Page

7.6 Query Run Successfully :

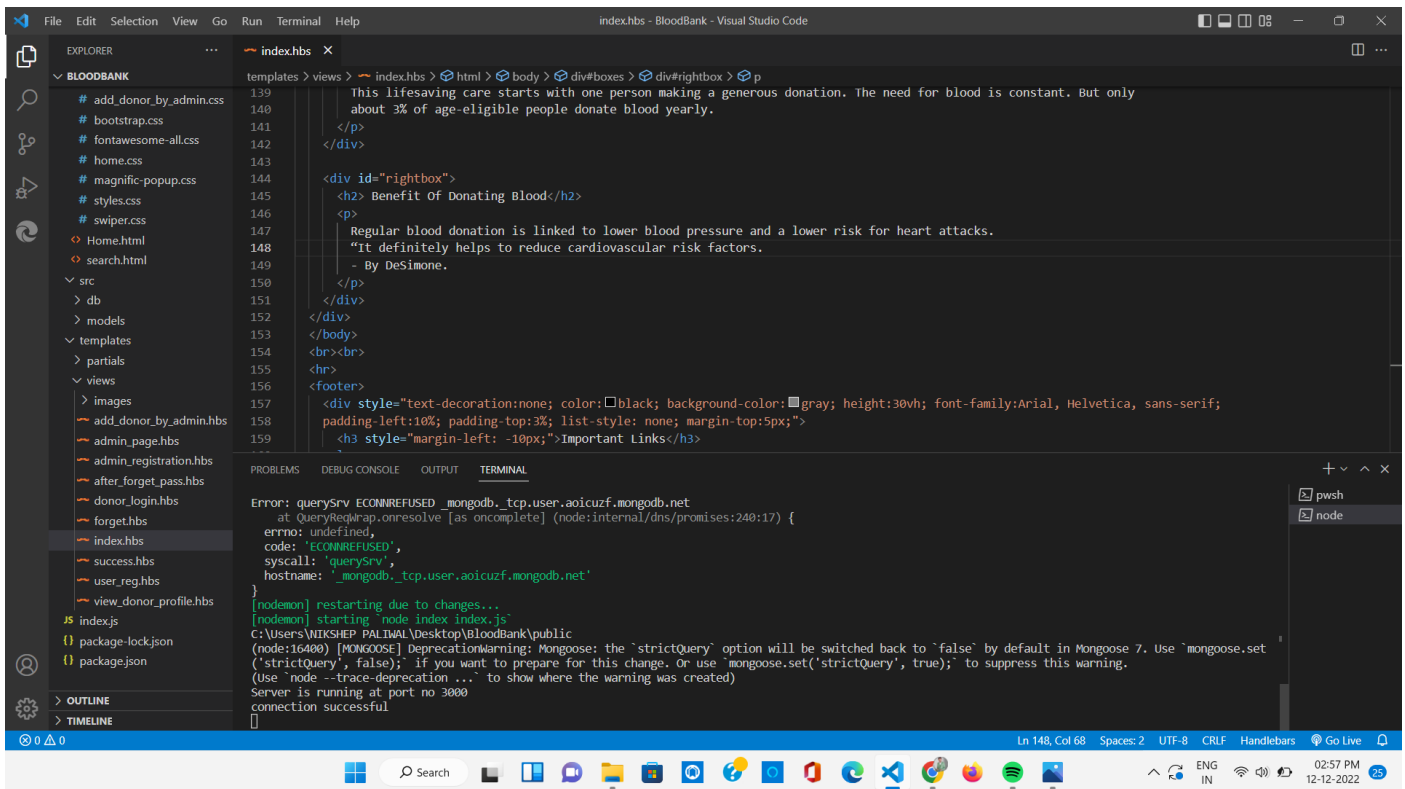


Fig.-7.6 Query Running

7.7 Database Stored Successfully:

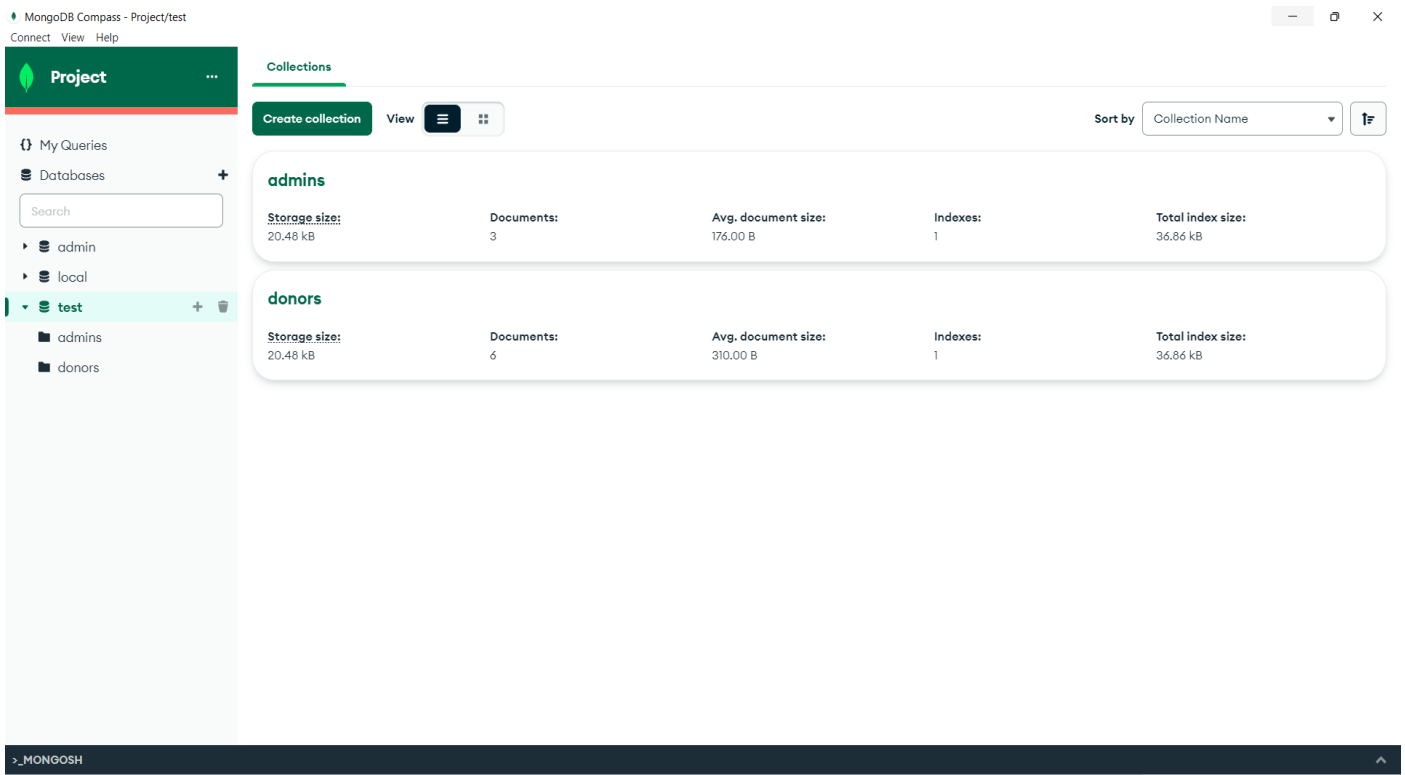
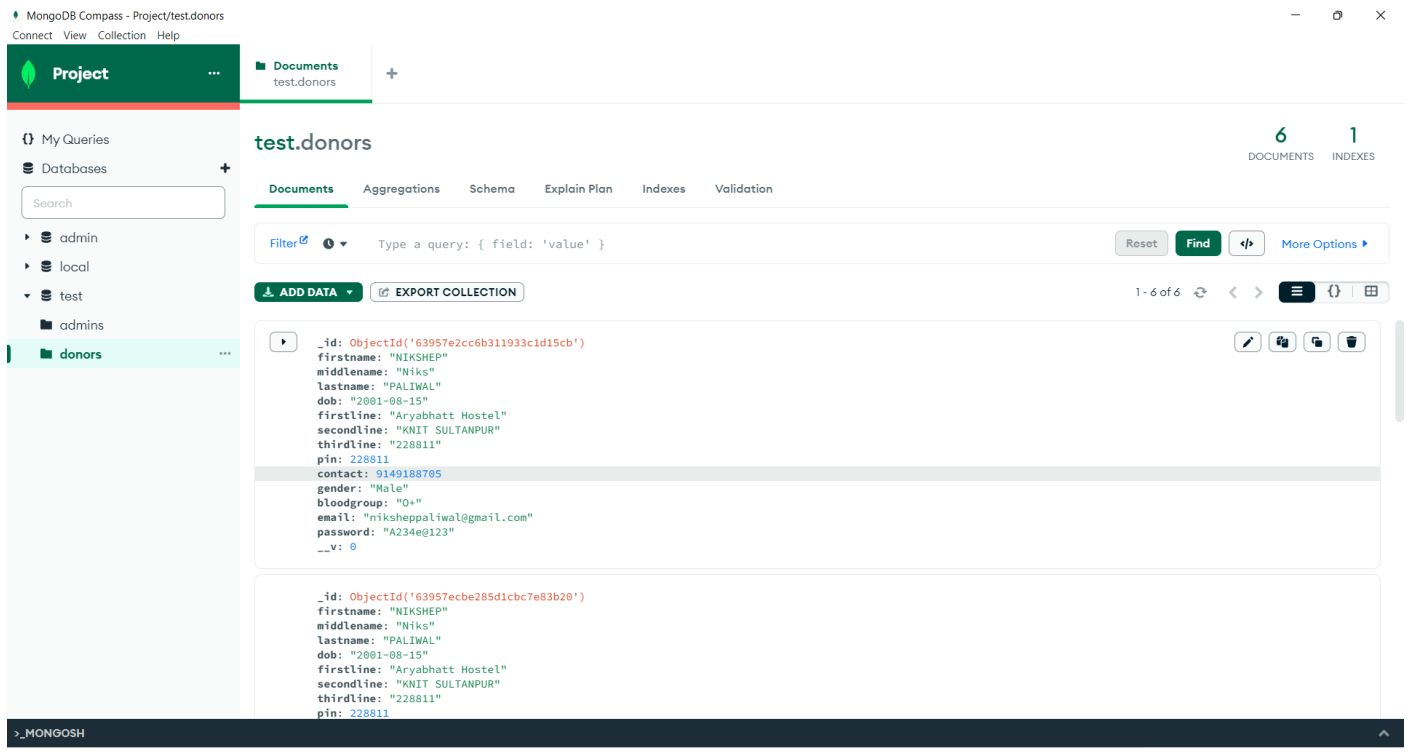


Fig.-7.7 Database Stored Successfully



8: References

- <https://developer.mozilla.org/en-US/docs/Web/HTML>
- <https://developer.mozilla.org/en-US/docs/Web/CSS>
- <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- <https://www.w3schools.com/html/>
- <https://www.w3schools.com/css/>
- <https://www.w3schools.com/javascript/>
- https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs