



---

# Simple Blood Bank management System

---



## Submitted By

Md.Masudur Rahman Nirob	ID: 22235103496
Humyra Jahan Tanha	ID: 22235103453
Humayra Nasrin	ID: 22235103455
Neamul Hasan Bulbul	ID: 22235103457
Umme Fawzia	ID: 22235103468

**Supervisor By:** Ahmed Shafkat (Lecturer)

Submitted in partial fulfillment of the requirements of the degree of  
Bachelor of Science in  
Computer Science and Engineering



Department of Computer Science and Engineering  
Bangladesh University of Business and Technology

DECEMBER 1, 2023

## **Contents**

<b>Declaration</b> .....	<b>iv</b>
<b>Approval</b> .....	<b>v</b>
<b>Dedication</b> .....	<b>vi</b>
<b>Acknowledgement</b> .....	<b>vii</b>
<b>Abstract</b> .....	<b>viii</b>
<b>1 Introduction</b> .....	<b>ix</b>
1.1 Introduction .....	<b>ix</b>
1.2 Problem Statement:.....	<b>ix</b>
1.3 Problem Background: .....	<b>x</b>
1.4 Project Objective:.....	<b>x</b>
1.5 Motivation .....	<b>xi</b>
1.6 Project Contribution .....	<b>xi</b>
1.7 Project Report Organization .....	<b>xii</b>
1.8 Hardware Requirements for the Present Project.....	<b>xii</b>
1.9 Software Requirements for the Present Project.....	<b>xii</b>
1.10 Summary .....	<b>xiii</b>
<b>2 Literature Review</b> .....	<b>xiii</b>
2.1 Introduction .....	<b>xiii</b>
2.2 Related work.....	<b>xiii</b>
2.3 Problem Analysis .....	<b>xv</b>
2.4 Summary.....	<b>xv</b>
<b>3 Proposed Model</b> .....	<b>xvi</b>
3.1 Introduction .....	<b>xvi</b>
3.2 Feasibility Analysis.....	<b>xvi</b>
3.3 Requirement Analysis.....	<b>xvii</b>

3.4 Summary.....	<b>xvii</b>
<b>4 Implementation and Testing.....</b>	<b>xviii</b>
4.1 Introduction .....	<b>xviii</b>
4.2 Hardware Requirements.....	<b>xviii</b>
4.3 Software Requirements.....	<b>xviii</b>
4.4 Summary.....	<b>xix</b>
<b>5 Conclusion.....</b>	<b>xx</b>
5.1 Conclusion .....	<b>xx</b>
5.2 Limitation and Future Works.....	<b>xx</b>
5.2.1 Limitation.....	<b>xx</b>
5.2.2 Future Works.....	<b>xxi</b>
<b>Appendix.....</b>	<b>xxii</b>

## Declaration

We do hereby declare that the project works presented here entitled, “Simple Blood Bank Management System” are the results of our own works. We further declare that the project has been compiled and written by us and no part of this project has been submitted elsewhere for the requirements of any degree, award or diploma or any other purposes except for this project. The materials that are obtained from other sources are duly acknowledged in this project.

Signature of Developers:

---

Md.Masudur Rahman Nirob ID: 22235103496

---

Humyra Jahan Tanha ID: 22235103453

---

Humayra Nasrin ID: 22235103455

---

Neamul Hasan Bulbul ID: 22235103457

---

Umme Fawzia ID: 22235103468

# Approval

I do hereby declare that the project works presented here entitled as Simple Blood Bank Management System are the outcomes of the original works carried out by Md.Masudur Rahman Nirob, Humyra Jahan Tanha, Humayra Nasrin, Neamul Hasan Bulbul and Umme Fawzia under my supervision. I further declare that no part of this project has been submitted elsewhere for the requirements of any degree, award or diploma or any other purposes except for this project. I further certify that this project is applicable for SDP 1 and I also sure that the dissertation meets the requirements and standard for the degree of Computer Science and Engineering.

---

Convener  
Khan Md. Hasib  
Assistant Professor  
Department of Computer Science Engineering  
Bangladesh University of Business and Technology

---

Board Member  
Ashfia Jannat Keya  
Lecturer  
Department of Computer Science Engineering  
Bangladesh University of Business and Technology

---

Supervisor  
Ahmed Shafkat  
Lecturer  
Department of Computer Science Engineering  
Bangladesh University of Business and Technology

## Dedication

We would like to dedicate this project to our loving parents & Blood Donors . . .

## Acknowledgement

We are deeply thankful to Bangladesh University of Business and Technology (BUBT) for providing us such a wonderful environment to peruse our project. We would like to express our sincere gratitude to Ahmed Shafkat Lecturer, CSE, and BUBT. We have completed our project with his help. We found the project area, topic, and problem with his suggestions. He guided us with our study, and supplied us many articles and academic resources in this area. He is patient and responsible. When we had questions and needed his help, he would always find time to meet and discuss with us no matter how busy he was. We also want to give thanks to our CSE department. Our department provide us logistic supports to complete our project with smoothly. We would also like to acknowledge our team members for supporting each other and be grateful to our university for providing this opportunity for us.

# Abstract

Humans have traditionally relied on handwritten methods to document various aspects of life. In the field of medical care, meticulous record keeping for donors is crucial. To streamline this process, we have designed a user-friendly data-entry software specifically tailored for storing comprehensive information about blood donors in an efficient and straightforward manner. Data is stored in binary format to accommodate vast amounts of information, as hospitals are required to maintain records for hundreds of patients.

The Blood Bank Management System developed in C++ provides a user-friendly interface for managing donor information within a university setting. The system is designed to maintain a database of student donors, including their personal details, donation history, and contact information. The program allows users to perform various operations such as adding new donors, viewing the donor list, searching for donors based on blood type, updating donor information, and deleting donor records.



# **1 Introduction**

## **1.1 Introduction**

In today's day of modernization, we as humans have steered away from using handwritten mechanisms for keeping record about anything and everything. Since blood donation is of vital importance in our lives, we have developed a data-entry software for storing information about blood donor in a simple yet effective manner. The data is stored in binary form to be able to hold a high amount of data since blood bank have to keep records of hundreds of patients. Simple Blood Bank Management System (SBBMS) aids in the registration of complete blood donor data. It records retains the blood donor's including their name, id, age, blood type, contact number, etc. It minimizes the need to gather these details during each software visit. The primary goal of the software is to serve as a fully automated Hospital blood bank front desk management system—swift, user-friendly, and cost-effective. This is intended to decrease manual donor registrations, thereby reducing the likelihood of errors in donor records.

## **1.2 Problem Statement:**

In the medical field blood donation stands as one of the most critical services in the contemporary world. Consequently, The problem statement for a simple blood bank management system involves addressing inefficiencies in manual processes, ensuring accurate blood inventory tracking, timely donor management, and facilitating seamless blood distribution to meet urgent medical requirements. The project aims to enhance overall efficiency, reduce errors, and improve accessibility in blood bank operations

## **1.3 Problem Background:**

In the demanding field of healthcare, where the responsibility of caring for countless patients is already a formidable task, the added burden of manual data management only compounds the challenges faced by healthcare workers. Individuals requiring blood face diverse challenges depending on their unique situations and the specific context in which they need blood. To address these complexities and streamline the blood donation process, our innovative app offers a solution. Blood donors can effortlessly register for blood donation and to meet the need blood through the software, introducing a more convenient and efficient method. This not only eases the workload for healthcare workers but also enhances overall productivity by eliminating the tedious manual management of hundreds of patients' information. Recognizing the dire need to reduce such barriers in their workflow, our app stands as a valuable tool in improving efficiency and ensuring optimal patient care.

#### **1.4 Project Objective:**

The objective of a Blood Bank Management System is to streamline and enhance the efficiency of blood donation processes. This system aims to provide a centralized platform for managing blood inventory, donor records, and blood transfusion requests. By implementing this system, the project seeks to automate and organize the entire blood bank workflow, ensuring quick and accurate access to critical information. The objectives are:

- To make a valid informative portal for the blood bank.
- To allow the probable recipients to make search.
- To provide an efficient donor and blood stock management functions for blood bank.
- To improve the efficiency of blood stock management by alerting the blood bank staffs.
- To provide synchronized and centralized donor and blood stock database.
- To help raise awareness in the community about blood donation.

Ultimately, the Blood Bank Management System aspires to contribute to public health by fostering a more efficient, transparent, and responsive blood donation and distribution network. The project report will delve into the technical aspects, benefits, and potential impact on healthcare services.

### **1.5 Motivation:**

The motivation behind a blood management system is to ensure the safety and maintaining all the process of blood resources. This includes: Patient Safety, Resource Conservation, Cost Efficiency, Regulatory Compliance, Improved Patient Outcomes, Data and Analytics.

Overall, a blood management system aims to improve patient care, reduce waste, and promote responsible use of blood products.

The blood bank management system serves as a vital project by enhancing the efficiency of blood donation processes. It motivates the project by ensuring seamless donor registration, inventory tracking, and swift response during emergencies. This system plays a crucial role in saving lives by facilitating timely access to blood units, reducing wastage, and providing a user-friendly interface for both donors and healthcare professionals. The motivation lies in addressing critical gaps in blood supply management, promoting transparency, and ultimately contributing to a robust healthcare infrastructure, underscoring the significance of technology in optimizing blood bank operations for the greater good.

### **1.6 Project Contribution**

The overall contribution of the proposed software includes:

- Adding details about one or more patients
- Viewing the entire record about the patients
- Searching up a patient using only one of his/her details
- Editing patient information

- Deleting one or more detail about any patient
- Storing the data in binary form to hold a large amount of data

## **1.7 Project Report Organization**

An overview of the steps of the project report is organized as follows:

Chapter 2 presents the literature review on the Simple Hospital Management System.

Chapter 3 contains the analysis of the requirements, feasibility and methodology of our proposed software in detail

Chapter 4 provides a brief discussion of the implementation, tests, and evaluations to estimate our software. Lastly,

Chapter 5 is a review of our project work, including conclusions as well as discussions about the objectives for future work.

## **1.8 Hardware Requirements for the Present Project**

- PROCESSOR: Intel dual Core i3
- RAM: 1 GB
- HARD DISK/SSD: 80 GB

## **1.9 Software Requirements for the Present Project**

- OPERATING SYSTEM: Windows 7/XP/8/10 Home, Education, Pro/11 Home, Education, Pro
- FRONT END: C++ Language

## **1.10 Summary**

This chapter comprises of a broad overview of what problems we are specifically targeting to solve, what the purpose of our project is, along with the motivation for the output of the software. This section also represents the overall steps on which we carried out our entire project.

## 2 Literature Review

### 2.1 Introduction

Now at this age of modern technology, everyone from every sector is trying to get their work done with the help of technology supply-chain management, robotics, vehicle registration, national identity card management and more with the help of different programming languages. The project Simple Blood Bank Management System is also one of such projects. It makes work in hospitals easier and faster for both blood donors and the hospital management authority. Hospitals are very modern nowadays because of the features used in this type of management programs.

### 2.2 Related work

"Blood Care Hospital," situated in Houston, Texas, with a daily capacity to accommodate 500 blood donors, employs an advanced Blood Bank Management System to streamline operations.

The system incorporates several key features:

- **Donor Registration:** Enables efficient registration and tracking of blood donors, maintaining a comprehensive database of donor information.
- **Inventory Management:** Provides real-time monitoring of blood inventory, ensuring accurate records of blood types, quantities available, and expiration dates.
- **Transfusion Tracking:** Facilitates the management of blood transfusion requests, ensuring timely and accurate responses to medical needs.
- **Reports and Analytics:** Generates detailed reports and analytics on blood utilization, donor demographics, and inventory status, aiding in strategic decision-making.

- **Security and Compliance:** Adheres to healthcare regulations, ensuring the confidentiality and security of sensitive donor and patient information.
  - **Communication and Alerts:** Facilitates communication with donors and healthcare institutions, sending alerts for critical events such as low blood stock levels or upcoming donation events.
  - **User-Friendly Interface:** Offers an intuitive interface for healthcare professionals to navigate and manage the system efficiently.
  - **Integration Capabilities:** Integrates seamlessly with other healthcare systems, promoting data flow across different departments and ensuring a unified approach.
  - **Mobile Accessibility:** Provides a mobile application, allowing healthcare professionals to access and manage blood bank data remotely.
- 
- **The Challenge:** With a significant number of donors traveling from different regions, there was a need for an online platform. This challenge was addressed by creating a user-friendly website, enabling donors to schedule appointments in advance, enhancing the overall efficiency of blood donation processes.
- 
- **The Solution:** In response to the challenge posed by a significant number of donors traveling from different regions, Blood Care Hospital implemented a comprehensive online platform as part of its Blood Bank Management System. The solution involved the development of a user-friendly website that allows prospective blood donors to schedule appointments in advance. This proactive approach ensures a smoother and more organized blood donation process, minimizing wait times for donors and optimizing the overall efficiency of the blood bank. The online platform not only caters to the needs of local donors but also accommodates those arriving from different

cities or regions, providing a convenient and accessible means to contribute to the critical cause of blood donation.

### **2.3 Problem Analysis**

By researching deeper in the project, there was a problem, which came out of it. To ensure a protected use of software, a method of using or implying password is important. This feature is currently unavailable in it. By researching deeper in the project, there was some problem, which came out of it. To ensure a protected use of software, a method of using or implying password is important, monitoring blood storage, blood expire date etc. This feature is currently unavailable in it.

### **2.4 Summary**

The software is built for making the work environment easier and faster at any hospital. Modern technologies were used to build it and it works well having no error. Some problem was found which could be resolved by taking more time on building or including a function, which will give it protection.

### 3 Proposed Model

#### 3.1 Introduction

This chapter represents the proposed model and illustrates the feasibility analysis, requirement analysis as well as methodology where we will discuss how we developed the project, which is a simple blood bank management system that is used to make the blood bank management easier and efficient. Additionally, this chapter represents the procedure of how we developed the project with the programming language C++ with the compatible hardware. We will also highlight technical economical aspects of it

#### 3.2 Feasibility Analysis

Feasibility analysis is the method of concluding the fallibility of a system. This study is essential to open new concepts that could effectively improve a project's scope. So, it's best to make these decisions in advance. Now, in this part, whether the system is feasible for development or not will be discussed. This study also includes the availability of resource codes, evaluation of cost, how this system can benefit an organization, and how the system can be maintained after development. There are two types of feasibility to measure this analysis: technical and economical.

**Technical Feasibility:** In the technical feasibility assessment, the discussion revolves around the project's completion by evaluating the adequacy of technical knowledge and the compatibility of hardware and software. The chosen tools include the programming language C++ and Codeblocks as the integrated development environment (IDE). The recommended hardware configuration specifies a minimum of 2 GB RAM, 2 GB hard disk space, and a cache of at least 100 MB. The conclusion affirms that with these selected software and hardware components, the proposed system can be efficiently developed.



### **3.3 Economic Feasibility:**

The economic feasibility analysis focuses on the cost and potential financial gains of the project. The cost of developing the system is nearly zero as no additional purchases for hardware or software are required. The project utilizes the free-to-use programming language C++. The specified hardware requirements are minimal. The system aims to benefit the Blood Bank management by facilitating financial data collection and analysis, particularly in storing donor information. With low development and maintenance costs, the project is deemed economically efficient for the hospital authority.

### **3.4 Requirement Analysis**

Requirement analysis is a critical phase following requirement elicitation, involving a thorough examination and refinement of collected requirements. This process ensures consistency and clarity, often employing graphical representations of the entire system. The goal is to enhance overall project understandability. Customer interactions play a crucial role in clarifying points of confusion and prioritizing requirements based on importance. The iterative nature of this analysis aims to produce a more robust and well-defined system.

### **3.5 Summary**

In the section delves into technical and economic feasibility, along with requirement analysis for the project. Emphasizing a cost-effective approach, it details the use of a single programming language and minimal hardware requirements, providing a comprehensive overview of crucial project aspects.

## **4 Implementation and Testing**

### **4.1 Introduction**

For optimal functionality, computer software relies on specific hardware components and software resources. These prerequisites, termed system requirements, serve as guidelines rather than strict rules. Software typically outlines two sets of requirements: minimum and recommended. As newer software versions demand increased processing power and resources, system requirements tend to escalate. Analysts argue that this trend significantly influences upgrades to existing computer systems more than technological advancements do.

### **4.2 Hardware Requirements**

The primary set of requirements outlined by operating systems or software applications pertains to physical computer resources, commonly referred to as hardware. This list is often accompanied by a hardware compatibility list (HCL), particularly with operating systems. An HCL provides a catalog of tested compatible, and sometimes incompatible, hardware devices for a specific operating system or application. The subsequent sub-sections delve into different facets of hardware requirements.(Hardware Requirements for the Present Project 1.8)

### **4.3 Software Requirements**

Software Requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or prerequisites are generally not included in the software installation package and need to be installed separately before the software is installed. (Software Requirements for the Present Project 1.9)

## 4.4 Summary

Implementing a Blood Bank Management System (BBMS) brings automation and optimization to the intricate processes within healthcare institutions. Similar to hospital management software, a BBMS reduces manual efforts, alleviating stress on staff and enhancing overall productivity and coordination. The system provides a lasting competitive advantage by offering a more flexible allocation of human resources.

In the medical field, the significance of digital systems is underscored by the enhanced security they provide. Unlike manual systems prone to data leaks, a comprehensive BBMS ensures the safety of information through robust access control mechanisms, minimizing the risk of unauthorized access or information theft. This heightened security has contributed to the widespread adoption of BBMS in recent years.

Developing a BBMS involves a meticulous understanding of the processes it aims to establish, the functions it will support, and a clear definition of its users. Requirements are systematically outlined in a Software Requirements Specification (SRS) document, providing a shared vision between stakeholders and vendors. Accurate estimation, crucial for system development, is based on the specifications laid out in the SRS. Whether developed by a specialized company or a turnkey vendor, the BBMS becomes an integral tool in managing blood-related information efficiently and securely within the healthcare ecosystem.

## **5. Conclusion:**

### **5.1 Conclusion:**

In conclusion, the development of the Blood Bank Management System represents a significant achievement in the healthcare sector. This project aimed to streamline the operations of blood banks, making the process of blood donation, inventory management, and blood distribution more efficient, reliable, and secure. Through the development of this system, several important achievements and outcomes have been realized.

### **5.2 Limitation and Future work:**

#### **5.2.1 Limitation:**

Some limitations of a simple blood bank management system for a project report could include:

1. Scalability: The system may struggle to handle a large volume of data or a significant increase in users over time.
2. Security Concerns: Inadequate security measures might make the system vulnerable to unauthorized access or data breaches.
3. Lack of Advanced Features: A simple system may lack advanced features like real-time inventory tracking, donor-patient matching algorithms, or integration with other healthcare systems.
4. User Interface Design: If the user interface is not intuitive or user-friendly, it could impact the system's usability.
5. Limited Reporting and Analytics: The system might lack robust reporting and analytical tools, hindering data-driven decision-making.
6. Dependency on Manual Data Entry: If the system heavily relies on manual data entry, there's an increased risk of errors and inefficiencies.

Consider addressing these limitations in your project report for a comprehensive analysis.

### **5.2.2 Future Works:**

There are several avenues for enhancing the "Simple Blood Bank Management System":

1. Improved User Experience: Enhancing the system's interface for a more user-friendly experience.
2. Mobile App Development: Creating a mobile application for on-the-go access and updates.
3. Geolocation Integration: Adding location services for locating nearby blood donation centers and donors.
4. Donor Reminders: Implementing reminders to encourage regular blood donations.
5. Health Tracking: Enabling donors to monitor their health and receive health recommendations.
6. AI and Predictive Analytics: Utilizing AI and data analysis to forecast blood demand more accurately.

The project has great potential for growth and continued positive impact on the community it serves.

## Appendix

### Source Code:

```
#include <iostream>
#include <string.h>
#include <fstream>
#include <ctype.h>
#include <time.h>
#include <conio.h>
using namespace std;
struct Donor          // Structure declaration
{
    string name,department,email,bloodType,date; // Member (string variable)
    int age;          // Member (int variable)
    long long int id;    // Member (long long int variable)
    long long int number;    // Member (long long int variable)
    long long int num;    // Member (long long int variable)
};
const int MAX_DONORS = 100;
Donor donors[MAX_DONORS];
int numdonors = 0;
void savedonerdata() //save data for doner info in file.function
{
    ofstream file("D:\\Cse Project\\SDP 1\\donors.csv"); //add doner info in file
    if (file.is_open())
    {
        for (int i = 0; i < numdonors; i++)
        {
            file << donors[i].name << "\\t" << donors[i].id<<"\\t"<< donors[i].department<<"\\t"<<
donors[i].bloodType << "\\t" << donors[i].age <<"\\t"<< donors[i].date<<"\\t"<< donors[i].num<<"\\t"<<
donors[i].number<<"\\t" << donors[i].email<<endl;
        }
    }
}
```

```

        file.close();
    } else {
        cout << "Unable to open the file for saving." << endl;
    }
}

void DonorsdataFile() //show doner info from file. function
{
    ifstream file("D:\\Cse Project\\SDP 1\\donors.csv"); // show doner info
    if (file.is_open())
    {
        numdonors = 0;

        while (file >> donors[numdonors].name >> donors[numdonors].id >>
donors[numdonors].department >> donors[numdonors].bloodType >> donors[numdonors].age>>
donors[numdonors].date>> donors[numdonors].num>> donors[numdonors].number>>
donors[numdonors].email)
        {
            numdonors++;
        }

        file.close();
    } else {
        cout << "Unable to open the file.Doner List Is Empty" << endl;
    }
}

// Add this function for input validation
long long int getNumericInput() {
    long long int input;
    while (true) {
        cin >> input;
        if (cin.fail()) {
            cout << "Invalid input. Please enter number." << endl;
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
        } else {
            break;
        }
    }
}

```





```

newdonor.num = getNumericInput();
// cin>>newdonor.num;
cout<<"Enter Your Phone No: ";    //input user phone number
newdonor.number = getNumericInput();
// cin>>newdonor.number;
cout<<"Enter Your Email: ";    //input user email
cin>>newdonor.email;
donors[numdonors] = newdonor;
numdonors++;
cout << "\t\t\t Donor added successfully!" << endl; //after add doner to show this
savedonerdata(); // Save donors' data to the file after adding a new donor
}else{
    cout << "\t\t\t The blood bank is full. Cannot add more donors." << endl; //after don't add doner
}
}

void displayDonors() //display doner info in this function
{
    cout << "::Donor List::" << endl;    //display all donor info
    cout << "-----" << endl;
    for (int i = 0; i < numdonors; i++)
    {
        cout << "Donor " << i + 1 << ":" << endl;
        cout << "Name: " << donors[i].name << endl;
        cout<<"Student ID: "<<donors[i].id<<endl;
        cout << "Blood Type: " << donors[i].bloodType << endl;
        cout << "Age: " << donors[i].age << endl;
        cout<<"Last Date Of Donation: "<<donors[i].date<<endl;
        cout<<"Number Of Donation: "<<donors[i].num<<endl;
        cout<<"::-Contact Info:-:"<<endl;
        cout<<"Phone: "<<donors[i].number<<endl;
        cout<<"Email: "<<donors[i].email<<endl;
        cout << "-----" << endl;
    }
}

void searchDonors() //search doner info fanction
{

```



```

        cout<<"\t\t\t: ";
        cin >> updateinfo;
for (int i = 0; i < numdonors; i++)
{
    if (donors[i].id == updateinfo)    //upadet doner info
    {
        cout<<"Donor Name:"<<donors[i].name<<endl;
        cout << "Enter the new Last Date Of Donation : ";//<< updateinfo << ": ";
        //donors[i].date = getNumericInput();
        cin>>donors[i].date;
        cout << "Enter the new age for " << updateinfo << ": ";
        donors[i].age = getNumericInput();
        cout << "Enter the new Number Of Donation " << updateinfo << ": ";
        donors[i].num = getNumericInput();
        cout << "\t\t\t Donor information updated!" << endl;
        savedonerdata();
        return;
    } }
    cout << "\t\t\t Donor not found. Could not update." << endl;
}

void deleteDonor()            // delete doner function
{
    long long int deleteprofile;
    int j;
    cout<<"\n\t\t===== ";
    cout<<"\n\t\t|  Enter The Id of The Donor You Want to Delete  |"<<endl;
    cout<<"\t\t===== ";
    cout<<endl;
    cout<<"\t\t\t: ";
    cin >> deleteprofile;
    for (int i = 0; i < numdonors; i++) {
        if (donors[i].id == deleteprofile)            // delete doners info
        {
            for (j = i; j < numdonors - 1; j++)

```

```

    {
        donors[j] = donors[j + 1];
    }
    numdonors--;
    cout << "\t\t" << deleteprofile << " Has been deleted from the list." << endl;
    savedonerdata();
    return;
} }

cout << "\t\t Donor not found. Could not delete." << endl;
}

void delay(int time){
    clock_t goal= time+clock();
    while(goal>clock());
}

void Welcome()
{
    // Setting console color
    system("COLOR 00 ");
    // Blood Bank Animation
    char load[1][100] = {"Tomar Rokto Nimu ...."};
    for (int i = 0; i < 2; i++) {
        cout << "\n\n\n\n\n\n\n\n\n\n\n\n\n\n\t\t\t\t\t ";
        for (int j = 0; load[i][j] != '\0'; j++) {
            delay(100);
            cout << load[i][j];
        }
    }
    // Changing console color
    system("COLOR FC");
    delay(50);
    system("cls");
    // Displaying a border
    cout << "\n\t\t\t\t*";
    for (int str = 1; str < 54; str++) {
        delay(50);
    }
}

```

```

    cout << "*";
}

// Displaying project name
char prjct[300] = {"Simple Blood Bank Managment System "};
cout << "\n\n\t\t\t";
for (int i = 0; prjct[i] != '\0'; i++) {
    cout << " ";
    delay(100);
    cout << prjct[i];
}
cout << "\n";
delay(250);
cout << "\n\t\t\t\t\t";
for (int str = 1; str < 54; str++) {
    delay(50);
    cout << "*";
}
cout << "\n";
// Creating a line
for (int i = 0; i < 120; i++)
    cout << "_";
delay(100);
// Displaying additional text
cout << "\n\n\n\n\t\t\t\t\t Made by- ";
delay(350);
char name[2][30] = {"Team Wizard\n\n\n"};
for (int i = 0; i < 1; i++) {
    cout << "\n\t\t\t\t\t ";
    for (int j = 0; name[i][j] != '\0'; j++) {
        delay(50);
        cout << name[i][j];
    }
}
delay(500);
// Creating another line

```

```
for (int i = 0; i < 120; i++)

    cout << "_ ";

delay(100);

// Displaying a message to press any key to continue

cout << "\n\n\t\t Press any key to continue";

delay(350);

cout << ".";

delay(350);

cout << ".";

delay(350);

cout << ".";

delay(350);

cout << ".";

getch();

} int main() {

    Welcome();

    system("COLOR FC");

    DonorsdataFile();

    char choice;

    while (true) {

        try {

            cout << "\n\t\t=====";

            cout << "\n\t\t| BUBT Student Blood Bank Management System |" << endl;

            cout << "\t\t=====";

            cout << "\n\t\t \t \t ::Enter Your Choice::" << endl;

            cout << "\n\t\t\t\t To Add New Donor Data \t(1)\n\t\t\t\t To View List Of Donor \t(2)\n\t\t\t\t To Search Donor \t(3)" << endl;

            cout << "\t\t\t\t To Update Donor Data \t(4)\n\t\t\t\t To Delete Donor Data \t(5)\n\t\t\t\t To Log Out \t(6)" << endl;

            cout << "\t\t\t\t:";

            cin >> choice;

            if (!isdigit(choice)) {

                throw 1;

            }

            switch (choice) {
```

```

    case '1':
        system("cls");
        addDonors();
        break;
    case '2':
        system("cls");
        displayDonors();
        break;
    case '3':
        system("cls");
        searchDonors();
        break;
    case '4':
        system("cls");
        updateDonor();
        break;
    case '5':
        system("cls");
        deleteDonor();
        break;
    case '6':
        cout << "\n\t\t\t Thank You For Visiting Us!" << endl;
        return 0;
    default:
        system("cls");
        cout << "\t\t\t Invalid choice. Please try again." << endl;
} }

catch (int error) {
    system("cls");
    cout << "\t\t\t Invalid input. Please enter a numeric choice." << endl;
    cin.clear();
    cin.ignore(numeric_limits<streamsize>::max(), '\n');
} }

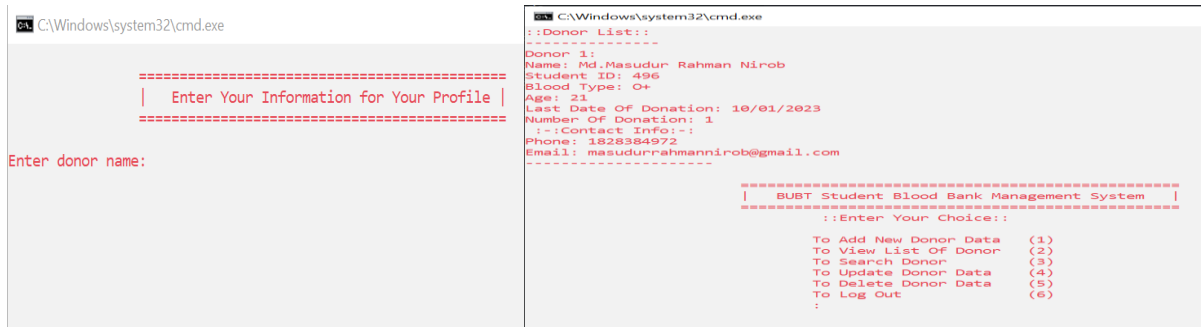
return 0; }:
```

## Output:



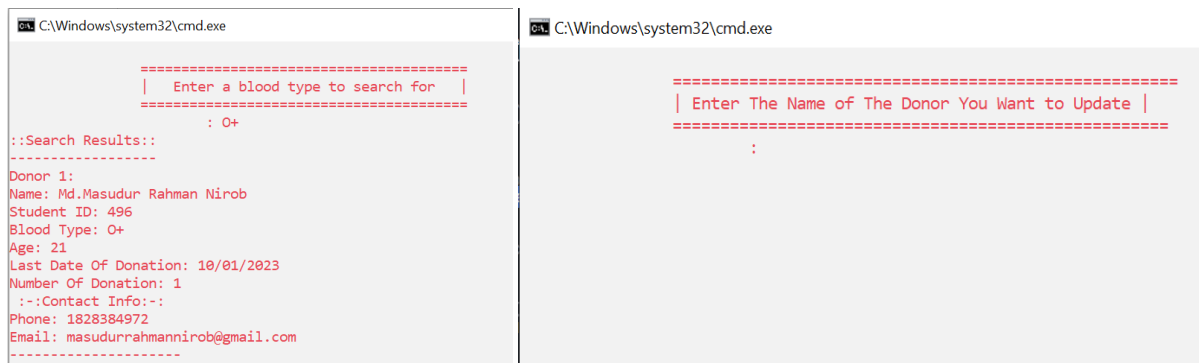
Loading Screen

Font page



If you Enter 1 For add new donors

If you Enter 2 For view list of donor



If you Enter 3 for Search Donor

If you Enter 4 for Update Donor



If you Enter 5 for Delete Donor

If you Enter 6 for Log Out