# E-BLOOD BANK MONITORING HOSPITALS

### A PROJECT REPORT

***Submitted by***

**Prashant Singh (20BCS1653) Aryan Jangir (20BCS1671) Chirag Bither (20BCS1838) Prateek Singh (20BCS1583) Gurwinder Singh (20BCS7634)**

***in partial fulfillment for the award of the degree of***

## BACHELOR OF ENGINEERING

### IN

**COMPUTER SCIENCE ENGINEERING**



### Chandigarh University

November, 2023



## BONAFIDE CERTIFICATE

Certified that this summer term project **“E-Blood Bank Monitoring Hospitals”** is the bonafide work of “**Prashant Singh (20BCS1653), Aryan Jangir (20BCS1671), Chirag Bither (20BCS1838), Prateek Singh (20BCS1583), Gurwinder Singh (20BCS7634)”** who carried out the project work under my supervision.

**SIGNATURE SIGNATURE**

Er Gaurav Mehta Dr. Navpreet Kaur Walia

**SUPERVISOR HEAD OF THE DEPARTMENT**

**COMPUTER SCIENCE AND ENGINEERING**

Submitted for the project viva-voce examination held on 28-11-2023

**INTERNAL EXAMINER EXTERNAL EXAMINER**



## ACKNOWLEDGEMENT

On this great occasion of accomplishment of our project “**E-Blood Bank Monitoring Hospitals”**, I would like to sincerely express our gratitude to **Er Gaurav Mehta**, who has supported through the completion of this project. Whenever we faced any trouble, theywere there for us.

### Finally, as one of the team members, I would like to appreciate all my group members for their support and coordination. I hope we will achieve more in our future endeavors.

TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **Sr. No** | **Chapters** | **Page No** |
| 1. | Abstract | **8-10** |
| 2. | Chapter-1 Introduction |  |
|  | 1.1Introduction | **11** |
|  | 1.2 Problem statement | **11** |
|  | 1.3 Project scope | **12** |
| 3. | Chapter-2 Literature survey |  |
|  | 2.1 Literature survey | **13-14** |
|  | 2.2 Relevance of Hospital Information | **15** |
|  | 2.3 Healthcare Worker’s Training and Efficiency | **15-16** |
|  | 2.4 Problem Definition | **16-17** |
|  | 2.5 Objective | **17-18** |
| 4. | Chapter-3 Design flow/Process |  |

|  |  |  |
| --- | --- | --- |
|  | 3.1 Evaluation and Selection Specific Features | **19-21** |
|  | 3.2 Design Constraints | **22** |
|  | 3.3 Analysis and Feature Finalization Subject to Constraints | **23-25** |
|  | 3.4 Design Flow | **25-26** |
| 5. | Chapter-4 Results analysis and validation |  |
|  | 4.1 Implementation of Solution | **27-33** |
|  | 4.2 Design Flow | **34-37** |
| 6. | Chapter-5 Conclusion and future work |  |
|  | 5.1 Conclusion | **38** |
|  | 5.2 Future Plan | **39-40** |
| 7. | References | **41-42** |
| 8. | Appendix | **43-67** |
| 9. | User Manual | **68-70** |
|  | Front Page | **1** |
|  | Bonafide Certificate | **2** |
|  | Acknowledgment | **3** |
|  | Table of content | **4-5** |
|  | List of figures | **6-7** |

# LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **SR. No.** | **Figure Name** | **Page No.** |
| 1. | Figure 1: 0-Level Data Flow Diagram | **34** |
| 2. | Figure 2: 1-Level Data Flow Diagram | **35** |
| 3. | Figure 3: 2-Level Data Flow Diagram | **35** |
| 4. | Figure 4: ER Diagram | **34** |
| 5. | Figure 5: Gantt Chart | **12** |
| 6. | Figure 6: Home Page | **23** |
| 7. | Figure 7: About us-1 | **24** |
|  | Figure 8: About us-2 | **28** |
| 8. | Figure 9: What we do-1 | **30** |
|  | Figure 10: What we do-2 | **31** |
|  | Figure 11: What we do-3 | **32** |

|  |  |  |
| --- | --- | --- |
| 9. | Figure 12: How to help-1 | **29** |
|  | Figure 13: How to help-2 | **30** |
| 10. | Figure 14: Success Stories-1 | **30** |
|  | Figure 15: Success Stories-2 | **31** |
| 11. | Figure 16: Plan your visit | **31** |
| 12. | Figure 17: Contact | **32** |
|  | Figure 18: Contact | **32** |
| 13. | Figure 19: How to Access the Application | **68** |
|  | Figure 20: How to do User Registration | **68** |
|  | Figure 21: How to Login | **69** |
|  | Figure 22: How Dashboard Navigate | **69** |

## सारांश

* यह सॉफ्टवेयर परियोजना अस्पताल का प्रबंधन करने के लिए विकसित की गई है। सॉफ्टवेयर परियोजना का उद्देश्य हमारे अस्पताल में रोगियों के लिए सुविधा का प्रबंधन करने के लिए "ई-ब्लड बैंक निगरानी अस्पतालों" विकसित करना है। इसमें हम रोगी की संबंधित रक्त समस्या का विवरण जानने में सक्षम हैं। हम पायथन और माई एसक्यूएल की कनेक्टिविटी की अवधारणा का उपयोग करने में सक्षम हैं। एक ब्लड बैंक निगरानी अस्पतालों में मुख्य रूप से एक अजगर और माई एसक्यूएल कनेक्टिविटी होती है और कृत्रिम बुद्धिमत्ता की एक अवधारणा भी देखी जाती है। रक्त बैंक प्रबंधन प्रणाली तक आसान पहुंच और इंटरफ़ेस के लिए एक आवेदन कार्यक्रम रक्त बैंक प्रबंधन प्रणाली के साथ जुड़ा हुआ है।
* एकप्लिकेशन प्रोग्राम या फ्रंट-एंड का उपयोग करके, हम सभी जानकारी को उचित तरीके से संग्रहीत, पुनर्प्राप्त और प्रबंधित कर सकते हैं। यह सॉफ्टवेयर, डिजाइन और काम करने में सरल होने के कारण, उपयोगकर्ताओं को अधिक प्रशिक्षण की आवश्यकता नहीं है, और अस्पताल में स्वचालित करने के लिए एक शक्तिशाली उपकरण के रूप में इस्तेमाल किया जा सकता है। निस्संदेह यह परियोजना रोगी की मदद करेगी जिन्हें रक्त की आवश्यकता है। इस परियोजना की मदद से हम विभिन्न चरणों में अस्पताल और रोगियों की मदद करने में सक्षम हैं। यह परियोजना ब्लड बैंक प्रबंधन प्रणाली के बोझ को कम करेगी और उन रोगियों की जांच करेगी जो रक्त से संबंधित समस्या का सामना कर रहे हैं। और रोगियों के डेटा को संग्रहीत करने के लिए। इस प्रणाली को विकसित करने का मुख्य उद्देश्य उन लोगों को रक्त प्रदान करना है जिन्हें रक्त की आवश्यकता होती है। रक्त की आवश्यकता वाले व्यक्तियों की संख्या दिन-प्रतिदिन बड़ी संख्या में बढ़ रही है। इस प्रणाली का उपयोग करके उपयोगकर्ता शहर में उपलब्ध रक्त समूहों की खोज कर सकता है और वह दाता का एक संपर्क नंबर भी प्राप्त कर सकता है जिसके पास उसी रक्त समूह की आवश्यकता है, प्रस्तावित प्रणाली (ई-ब्लड बैंक निगरानी अस्पतालों) रक्त बैंक प्रशासक को रक्त की मांग को पूरा करने में मदद करने के लिए डिज़ाइन की गई है। प्रस्तावित प्रणाली प्राप्तकर्ता, दाता और रक्त बैंकों के बीच की खाई को पाटने के लिए प्रक्रियात्मक दृष्टिकोण देती है। यह आवेदन सभी तीन पक्षों (यानी प्राप्तकर्ता, दाता और रक्त बैंक) के लिए एक सामान्य आधार प्रदान करेगा और प्राप्तकर्ता और / या रक्त बैंक द्वारा अनुरोधित रक्त की मांग की पूर्ति सुनिश्चित करेगा।

## ABSTRACT

* This software project is developed to manage the Hospital. The purpose of the software project is to develop the “E-Blood Bank Monitoring Hospitals” to manage the facility for patients in our hospital. In this we are able to learn the details of patient’s related blood problem. We are able to use the concept of connectivity of python and MySQL. A E-Blood Bank Monitoring Hospitals mainly consists of a python and MySQL connectivity and a concept of artificial intelligence also seen. An application program is tied with the E-Blood Bank Monitoring Hospitals for easy access and interface to the E- Blood Bank Monitoring Hospitals.
* Using application program or front-end, we can store, retrieve and manage all information in proper way. This software, being simple in design and working, does not require much of training to users, and can be used as a powerful tool for automating in Hospital. Undoubtedly this project will help the patient who are in need of blood. With the help of this project, we are able to help the hospital and patients in various stages. This project will reduce the burden of the E-Blood Bank Monitoring Hospitals and to check the patients who are facing blood related problem. And to store the data of patients. The main aim of developing this system is to provide blood to the people who need blood. The number of persons who need blood is increasing in large number day by day. Using this system user can search blood groups available in the city and he can also get a contact number of the donor who has the same blood group he needs. The proposed system (E-Blood Bank Monitoring Hospitals) is designed to help the blood bank administrator to meet the demand of blood by sending and/or serving the request for blood as and when required. The proposed system gives the procedural approach of how to bridge the gap between recipient, donor, and blood banks. This application will provide a common ground for all the three parties (i.e. recipient, donor, and blood banks) and will ensure the fulfillment of the demand for blood requested by the recipient and/or blood bank.

## ਸਾਰ

* ਇਹ ਸਾੱਫਟਵੇਅਰ ਪ੍ਰੋਜੈਕਟ ਹਸਪਤਾਲ ਦੇ ਪ੍ਰਬੰਧਨ ਲਈ ਵਿਕਸਤ ਕੀਤਾ ਗਿਆ ਹੈ। ਸਾਫਟਵੇਅਰ ਪ੍ਰੋਜੈਕਟ ਦਾ ਮਕਸਦ ਹੈ ਸਾਡੇ ਹਸਪਤਾਲ ਵਿੱਚ ਮਰੀਜ਼ਾਂ ਵਾਸਤੇ ਸੁਵਿਧਾ ਦਾ ਪ੍ਰਬੰਧਨ ਕਰਨ ਲਈ "ਈ-ਬਲੱਡ ਬੈਂਕ ਨਿਗਰਾਨੀ ਹਸਪਤਾਲ" ਦਾ ਵਿਕਾਸ ਕਰਨਾ। ਇਸ ਵਿੱਚ ਅਸੀਂ ਮਰੀਜ਼ ਨਾਲ ਸਬੰਧਿਤ ਖੂਨ ਦੀ ਸਮੱਸਿਆ ਦੇ ਵਿਸਥਾਰਾਂ ਬਾਰੇ ਜਾਣਨ ਦੇ ਯੋਗ ਹੁੰਦੇ ਹਾਂ। ਅਸੀਂ ਪਾਈਥਨ ਅਤੇ ਮਾਈ.ਐਸ.ਕਿਯੂ.ਐਲ. ਦੇ ਕੁਨੈਕਟੀਵਿਟੀ ਦੇ ਸੰਕਲਪ ਦੀ ਵਰਤੋਂ ਕਰਨ ਦੇ ਯੋਗ ਹਾਂ। ਇੱਕ ਬਲੱਡ ਬੈਂਕ ਨਿਗਰਾਨੀ ਹਸਪਤਾਲ ਵਿੱਚ ਮੁੱਖ ਤੌਰ ਤੇ ਇੱਕ ਪਾਈਥਨ ਅਤੇ ਮਾਈ ਐਸ ਕਿਉ ਐਲ ਕਨੈਕਟੀਵਿਟੀ ਹੁੰਦੀ ਹੈ ਅਤੇ ਨਕਲੀ ਬੁੱਧੀ ਦੀ ਇੱਕ ਧਾਰਣਾ ਵੀ ਵੇਖੀ ਜਾਂਦੀ ਹੈ। ਇੱਕ ਐਪਲੀਕੇਸ਼ਨ ਪ੍ਰੋਗਰਾਮ ਬਲੱਡ ਬੈਂਕ ਪ੍ਰਬੰਧਨ ਪ੍ਰਣਾਲੀ ਨਾਲ ਅਸਾਨ ਪਹੁੰਚ ਅਤੇ ਇੰਟਰਫੇਸ ਲਈ ਬਲੱਡ ਬੈਂਕ ਪ੍ਰਬੰਧਨ ਪ੍ਰਣਾਲੀ ਨਾਲ ਜੁੜਿਆ ਹੋਇਆ ਹੈ।
* ਇੱਕਐਪਲੀਕੇਸ਼ਨ ਪ੍ਰੋਗਰਾਮ ਜਾਂ ਫਰੰਟ-ਐਂਡ ਦੀ ਵਰਤੋਂ ਕਰਕੇ, ਅਸੀਂ ਸਾਰੀ ਜਾਣਕਾਰੀ ਨੂੰ ਸਹੀ ਤਰੀਕੇ ਨਾਲ ਸਟੋਰ ਕਰ ਸਕਦੇ ਹਾਂ, ਮੁੜ-ਪ੍ਰਾਪਤ ਕਰ ਸਕਦੇ ਹਾਂ ਅਤੇ ਪ੍ਰਬੰਧਿਤ ਕਰ ਸਕਦੇ ਹਾਂ। ਇਹ ਸੌਫਟਵੇਅਰ, ਡਿਜ਼ਾਈਨ ਅਤੇ ਕੰਮ ਕਰਨ ਵਿੱਚ ਸਰਲ ਹੋਣ ਕਰਕੇ, ਵਰਤੋਂਕਾਰਾਂ ਨੂੰ ਜ਼ਿਆਦਾ ਸਿਖਲਾਈ ਦੀ ਲੋੜ ਨਹੀਂ ਹੁੰਦੀ, ਅਤੇ ਇਸਨੂੰ ਹਸਪਤਾਲ ਵਿੱਚ ਸਵੈਚਲਿਤ ਕਰਨ ਲਈ ਇੱਕ ਸ਼ਕਤੀਸ਼ਾਲੀ ਔਜ਼ਾਰ ਵਜੋਂ ਵਰਤਿਆ ਜਾ ਸਕਦਾ ਹੈ। ਬਿਨਾਂ ਸ਼ੱਕ ਇਹ ਪ੍ਰੋਜੈਕਟ ਉਸ ਮਰੀਜ਼ ਦੀ ਮਦਦ ਕਰੇਗਾ ਜਿਸ ਨੂੰ ਖੂਨ ਦੀ ਲੋੜ ਹੈ। ਇਸ ਪ੍ਰੋਜੈਕਟ ਦੀ ਮਦਦ ਨਾਲ ਅਸੀਂ ਕਈ ਪੜਾਵਾਂ ਵਿੱਚ ਹਸਪਤਾਲ ਅਤੇ ਮਰੀਜ਼ਾਂ ਦੀ ਮਦਦ ਕਰਨ ਦੇ ਯੋਗ ਹਾਂ। ਇਹ ਪ੍ਰੋਜੈਕਟ ਬਲੱਡ ਬੈਂਕ ਪ੍ਰਬੰਧਨ ਪ੍ਰਣਾਲੀ ਦੇ ਬੋਝ ਨੂੰ ਘੱਟ ਕਰੇਗਾ ਅਤੇ ਉਹਨਾਂ ਮਰੀਜ਼ਾਂ ਦੀ ਜਾਂਚ ਕਰੇਗਾ ਜੋ ਖੂਨ ਨਾਲ ਸਬੰਧਿਤ ਸਮੱਸਿਆ ਦਾ ਸਾਹਮਣਾ ਕਰ ਰਹੇ ਹਨ। ਅਤੇ ਮਰੀਜ਼ਾਂ ਦੇ ਅੰਕੜਿਆਂ ਨੂੰ ਸਟੋਰ ਕਰਨ ਲਈ। ਇਸ ਪ੍ਰਣਾਲੀ ਨੂੰ ਵਿਕਸਤ ਕਰਨ ਦਾ ਮੁੱਖ ਉਦੇਸ਼ ਉਹਨਾਂ ਲੋਕਾਂ ਨੂੰ ਖੂਨ ਪ੍ਰਦਾਨ ਕਰਨਾ ਹੈ ਜਿੰਨ੍ਹਾਂ ਨੂੰ ਖੂਨ ਦੀ ਲੋੜ ਹੈ। ਜਿਨ੍ਹਾਂ ਵਿਅਕਤੀਆਂ ਨੂੰ ਖੂਨ ਦੀ ਲੋੜ ਹੈ, ਉਨ੍ਹਾਂ ਦੀ ਗਿਣਤੀ ਦਿਨ-ਬ-ਦਿਨ ਵੱਡੀ ਗਿਣਤੀ ਵਿੱਚ ਵੱਧ ਰਹੀ ਹੈ। ਇਸ ਪ੍ਰਣਾਲੀ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਵਰਤੋਂਕਾਰ ਸ਼ਹਿਰ ਵਿੱਚ ਉਪਲਬਧ ਖੂਨ ਦੇ ਗਰੁੱਪਾਂ ਦੀ ਤਲਾਸ਼ ਕਰ ਸਕਦਾ ਹੈ ਅਤੇ ਉਹ ਦਾਨੀ ਦਾ ਇੱਕ ਸੰਪਰਕ ਨੰਬਰ ਵੀ ਪ੍ਰਾਪਤ ਕਰ ਸਕਦਾ ਹੈ ਜਿਸਦਾ ਬਲੱਡ ਗਰੁੱਪ ਉਹੀ ਹੈ ਜਿਸਦੀ ਉਸਨੂੰ ਲੋੜ ਹੈ ਪ੍ਰਸਤਾਵਿਤ ਪ੍ਰਣਾਲੀ (ਈ-ਬਲੱਡ ਬੈਂਕ ਨਿਗਰਾਨੀ ਹਸਪਤਾਲ) ਨੂੰ ਲੋੜ ਪੈਣ 'ਤੇ ਖੂਨ ਵਾਸਤੇ ਬੇਨਤੀ ਭੇਜਣ ਅਤੇ/ਜਾਂ ਸਰਵ ਕਰਨ ਦੁਆਰਾ ਖੂਨ ਦੀ ਮੰਗ ਨੂੰ ਪੂਰਾ ਕਰਨ ਵਿੱਚ ਬਲੱਡ ਬੈਂਕ ਪ੍ਰਸ਼ਾਸਕ ਦੀ ਮਦਦ ਕਰਨ ਲਈ ਵਿਉਂਤਿਆ ਗਿਆ ਹੈ। ਤਜਵੀਜ਼ ਕੀਤੀ ਪ੍ਰਣਾਲੀ ਪ੍ਰਕਿਰਿਆਤਮਕ ਪਹੁੰਚ ਪ੍ਰਦਾਨ ਕਰਦੀ ਹੈ ਕਿ ਪ੍ਰਾਪਤ ਕਰਤਾ, ਦਾਨੀ, ਅਤੇ ਬਲੱਡ ਬੈਂਕਾਂ ਵਿਚਕਾਰ ਪਾੜੇ ਨੂੰ ਕਿਵੇਂ ਪੂਰਾ ਕਰਨਾ ਹੈ। ਇਹ ਅਰਜ਼ੀ ਸਾਰੀਆਂ ਤਿੰਨਾਂ ਧਿਰਾਂ (ਉਦਾਹਰਨ ਲਈ ਪ੍ਰਾਪਤ ਕਰਤਾ, ਦਾਨੀ, ਅਤੇ ਬਲੱਡ ਬੈਂਕਾਂ) ਵਾਸਤੇ ਇੱਕ ਸਾਂਝਾ ਆਧਾਰ ਪ੍ਰਦਾਨ ਕਰੇਗੀ ਅਤੇ ਪ੍ਰਾਪਤ ਕਰਤਾ ਅਤੇ/ਜਾਂ ਬਲੱਡ ਬੈਂਕ ਵੱਲੋਂ ਬੇਨਤੀ ਕੀਤੇ ਖੂਨ ਦੀ ਮੰਗ ਦੀ ਪੂਰਤੀ ਨੂੰ ਯਕੀਨੀ ਬਣਾਵੇਗੀ।

## CHAPTER I: INTRODUCTION

### INTRODUCTION:

The human body has millions of functions and is an extremely clever and complicated organization. He has been able to comprehend all of these complex functions through his studies and tests. The advancement of science and technology made medicine a crucial component of research. Medical science evolved over time into a completely new field of study. As of right now, the health sector is made up of hospitals, research and development centers, and medical schools. Therefore, the health industry strives to give the average person access to the best medical services.

### PROBLEM STATEMENT:

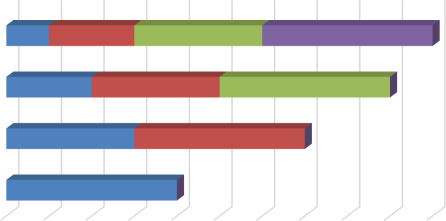
Since a hospital is connected to regular people's lives and daily activities, I made the decision to work on this project.

The manual handling of the record takes a long time and is very error prone. The goal of this project is to automate or put online routine tasks such room duties, new patient admission, patient discharge, assigning a doctor, and eventually computing the bill, among other daily tasks. I've done my best to use a structured, modular approach and a menu-oriented interface to simplify the challenging Hospital Management System procedure as much as I can. I made an attempt to create the software so that users would not encounter any problems using it and that future expansion would be simple to accomplish. The major goal of my exercise is to execute each Hospital's activity in a computerized manner rather than manually, which takes time, yet I cannot claim that my task is entirely exhaustive.

I'm certain that this software package can be easily used by those who aren't programmers, minimizing the likelihood of human mistake.

### PROJECT SCOPE:

The proposed software is the E-Blood Bank Monitoring Hospitals, which can be utilized by hospitals, clinics, dispensaries, and pathology labs. The system aims to gather patient information and store it for future use, replacing the current paper- based process. The existing procedure is slow and inefficient, resulting in delays in obtaining updated patient lists. By implementing the system, the goal is to reduce overtime costs and improve the ability to provideaccurate care to a larger number of patients. The documentation for the system includes both functional and non- functional requirements.



GANTT CHART

FINAL REPORT

PHASE-3

PHASE-2

PHASE-1

0

1

2

3

4

5

6

7

8

9

10

WEEK-1,2 WEEK-3,4 WEEK-5,6 WEEK-7,8

# CHAPTER II: LITERATURE SURVEY

### LITERATURE REVIEW:

The main goal of this literature review is to gain an understanding of how virtual assistants impact time management and independent growth among students in higher education.

1. Existing hospital management systems face significant challenges regarding operational efficiency and waiting times in various processes, departments, and personnel. To address these issues, this study proposes a framework that utilizes RFID (Radio Frequency ID) and wireless sensor technology. This system facilitates live monitoring of hospital resources, personnel, and individuals as they move through established protocols in daily activities.
2. This study focuses on Hospital Management Systems (HMS) commonly utilized in Chinese hospitals to streamline routine medical tasks, aiming to achieve greater speed and efficiency. The research highlights specific limitations of the E-Blood Bank Monitoring Hospitals (BBMH), such as the absence of a method to assess or quantify enhancements in healthcare service quality. To overcome these limitations, the study proposes the implementation of the HMS, which aims to improve service quality, identify potential cost-saving opportunities, and evaluate healthcare services. [3] The ability to evaluate services empowers hospitals to enhance customer satisfaction ratings and gain a competitive edge over hospitals lacking a similar system or performing poorly in service quality. During the initial planning, requirement gathering, and design phases of enterprise projects, significant challenges arise, often resulting in project abandonment due to high costs and uncertainties. The lack of subject-matter expertise and understanding of various hospital business functions contribute to the challenges faced in cost management during these phases.
3. Admission of patients
4. Planning of patient treatments
5. Entry of orders
6. Implementation of diagnostic and treatment procedures
7. Management of administrative documents
8. Billing processes
9. Recording clinical documentation

The process of referring patients to specialized medical institutions can be challenging due to a lack of understanding and experience with the entities involved, such as patients and clinical findings, and their responsibilities within various enterprise functions. To reduce the expenses associated with researching and analyzing the current state of healthcare systems and identifying gaps and requirements, this study aims to develop a hierarchical reference data model. This model will serve as a foundational framework for future development projects of Health Information Systems (HIS), offering flexibility for implementation based on specific needs. Major medical suppliers include pharmaceutical firms, the scientific community, and the software development community. [4] Internal factors within hospitals, such as personnel qualifications, business strategies, organizational culture, morale, and equipment availability, also impact the services provided and the delivery methods. This chapter presents a comprehensive literature review that addresses research topics relevant to the study's goals, including local, regional, and international studies on health information management systems. It also defines the components and challenges associated with implementing a health information management system. The effective utilization of information by administrators of RHIS (Routine Health Information Systems) is heavily dependent on accurate and timely data processing by software. When software fails to deliver this, it hampers decision-making by providing insufficient insights. The overall impact of RHIS on the effectiveness of the health system is yet to be fully assessed. The main focus of RHIS is on managing resources and delivering services within the healthcare system. The proposed operational definition of health system performance aims to uphold or enhance service coverage while making required adjustments or improvements to the financial and human resources associated with the provided services. This definition is deeply connected to the relationship between RHIS and health system performance to provide the services in healthcare field.

### RELEVANCE OF HOSPITAL INFORMATION SYSTEM:

In today's age, hospitals that fail to adopt hospital information systems will face difficulties in competing with other healthcare institutions due to significant advancements in medical technology and increasing patient expectations. The main motivation behind automating hospital information systems is the inefficiency of manual tasks (Meinert & Peterson, 2009:9). These systems play a critical role in enhancing the quality of care, operational efficiency, and informed decision-making. According to Ghosh (2010), hospital information systems enable convenient access to patient data centers in Kenya, streamline processes for patient registration and visit records, and have the capacity to handle information for a large number of patients. Successful implementations of electronic med record have also been witnessed in developing countries, including the Lilongwe HIS in Malawi, which is utilized for managing various clinical issues in the pediatric department of the Central Hospital; Partners in Health (PIH)-HIS in Peru; the HIV-HIS system in Haiti; Careware in Uganda; the PEPFAR project in Tanzania; and the National HIS in Zambia (Sood et al., 2008:14). Ensuring widespread acceptance among physicians necessitates addressing concerns related to confidentiality, privacy, and security when deploying these electronic medical records. Incorporating Information and Communication Technology (ICT) into the medical curriculum of developing countries can help overcome the challenge of clinicians understanding the benefits and the impact of these systems on their routines and hospital processes. Therefore, the primary objective of this study is to evaluate the effectiveness of health information system among health professionals.

### HEALTCARE WORKERS TRAINING:

The usage of PC’s/Laptops in hospitals has not been common in Kenya. More private hospitals than government hospitals have adopted computerization of hospital services. Huryk (2010) noted that a number of nurses displayed a statistically

significant correlation with their attitude towards computerization. Within KNH, nurses under the age of 40 showed a more positive attitude towards computerization than those beyond the age of 40. The main factor influencing electronic health is said to be practitioners' ICT training (Ochieng and Hosoi 2005:27; Marques et al. 2011: 91). ICT expertise is necessary to promote favorable attitudes towards electronic medical records, which will lead to a broader acceptance of these records. As a result, developed nations have added ICT training to health courses given at various academic levels in an effort to improve practitioners' ICT skills. According to Sood et al. (2008:16), advanced technologies are being used by industrialized nations to teach doctors, including robotics, virtual reality, and 3D simulations. ICT is also a part of the curriculum for medical courses. This is due to the fact that clinicians who possess ICT abilities can recognize the potential advantages of ICT in the execution and development of the numerous procedures they are involved in.

### PROBLEM DEFINITION:

Choosing exactly what to develop is the most difficult aspect of creating a software system. The establishment of the specific technical requirements is the most challenging aspect of the conceptual effort. It is difficult to define and implement good, full requirements, and many of us have failed in this endeavour. Yet, we keep moving forward.

* + - Identification and documentation of user’s needs.
      * Creation of a document that describes the external behavior and the association constraints that will satisfy those needs.
      * Analysis and validation of the requirements documents to ensure consistency, completeness, and feasibility.
    - Evolution of needs.

This project's system design demonstrates how control moves throughout the system.

Additionally, it displays the software and hardware needed for the program's execution. The primary goal of creating this system was to incorporate AI into the system. Many obstacles stand in the way of the application's potential users.

The user can search for the application he wishes to execute using this system. Additionally, there is a robust security system that can make our "E-Blood Bank Monitoring Hospitals" secure. To demonstrate how this application works, we create numerous applications.

### OBJECTIVE:

The software system is expected to do the following functionality:

* + User-friendly, (GUI) based integrated environment.
  + Keep up with all the paperwork and produce the necessary reports.
  + To offer a simple user interface for interacting with a centralized database.
  + To determine potential areas for simplification and the crucial operation procedure. The application software's potential is restricted by the following restrictions and functional boundaries, which are apparent despite the developer's best efforts.
  + No capability to create specialized reports. Only certain reports are included.
  + There is no provision for staff member entry or transfer, although this may be readily added with the addition of modules.

Regarding the project's potential future scope, it is first and foremost open to any modular expansion, meaning that more modules or functionalities may be created and integrated to meet changing user needs. The reports and any component of the software can be easily and independently updated.

Additionally, it displays the software and hardware needed for the program's execution. The primary goal of creating this system was to incorporate AI into the system. Many obstacles stand in the way of the application's potential users.

The user can search for the application he wishes to execute using this system. Additionally, there is a robust security system that can make our "E-Blood Bank Monitoring Hospitals" secure. To demonstrate how this application works, we create numerous applications.

.

# CHAPTER III: Design flow/Process

**1. Evaluation & Selection Specification/Features**

1. Objectives:
   * The major goal of this programme is to give patients a personal assistant where they may register and access information whenever and wherever they need it.
   * It can be exceedingly challenging for many people to register oneself at a hospital. Therefore, this application will assist the user in accessing the programme's data.
   * The best and most accurate applications are also available in our application, which will simplify peoples' lives.
2. Single Entity:
   * Executing the project with a small team is a challenging endeavour, but we were able to continue working by breaking it up into modules, cooperating with one another, and making sure the project is completed as a Single Entity.
3. Benefits:
   * It has several advantages and is distinctive in its own right. Because the app is a learning platform and learning is a continual process in everyone's life, the app's advantages don't seem to end.
4. Life Cycle:
   * The project life cycle started with devising out the idea for the project.
   * After successfully deciding the project, our team started to plan out how the

starting phase will be executed.

* + The important terms regarding the project were defined and we started with the design process.
  + Following the completion of the project's design, it was developed in about a month, tested for errors, and then deployed as the last stage of the development life cycle.

1. Team Spirit:
   * Our team overcame numerous challenges during the development process to produce the intended outcomes. We occasionally needed days to fix a single error, which was irritating because it stopped the development. Despite the challenges, we persisted and ultimately succeeded in finishing the job.
2. Risk and Uncertainty:

* Project development poses a lot of risk and uncertainties which need to be taken care of during the development process. Some of them include:
  + Is the project relevant as per the client’s needs?
  + Scalability of the project.
  + Technical faults and uncertainties.
  + Losses incurred for funding the project.
  + System will support the project.
* In order to manage all of these risks and come up with a solution that minimizes the aforementioned factors, we worked nonstop.

1. Directions:
   * In order to improve the hospital management system, we listened to client comments after creating a prototype for the blood bank system.
   * We built a user-friendly application following their instructions based on the feedback we obtained from customers.
2. Uniqueness:

* Our project is distinctive because of the E-Blood Bank Monitoring Hospitals used in it. It will make using the blood bank site easier for the usEr There will be different features. A version of the E-Blood Bank Monitoring Hospitals that our teamdeveloped and created will be useful to users as needed.
* Additionally, the application is appealing because of how interactive the GUI is. This project has a very strong security system. The system is extremely difficult to hack.

1. Flexibility:
   * The project has been planned such that it can always be changed to meet the client's needs. The project can be used in a variety of ways, and adjustments can be changed dynamically as needed.
2. Sub-contrasting:
   * Our team needed very little outside assistance because we were completely knowledgeable about every stage of the project development cycle. However, Er Gaurav Mehta, our supervisor, and Er Vishav Pratap Singh, our co-supervisor, were assigned to assist us in ensuring the success of this project. We also want to thank Er Swati for his assistance during the examination.

### DESIGN CONSTRAINTS

1. Time:

* We concentrated on how much time we would need to complete the project throughout the planning stage of the development life cycle to make sure we stayed within the time limitations.
* This ensured that we worked within the time constraints.

1. Cost:

* When determining the amount of money needed to complete the project, we set a maximum spending cap and made sure that even if additional resources were needed, the cost limitations would not be impacted.

1. Scope:

* As was already indicated in the project plan, our project is a little different from others' designs in that regard. As a result, the scope limitations are excluded from the debate because we selected the project quite well.

1. Quality:

* Our staff did its utmost to maintain the project's high standard. Rest assured that we can make changes as needed while still adhering to quality limits thanks to customer feedback.

1. Require Funds:

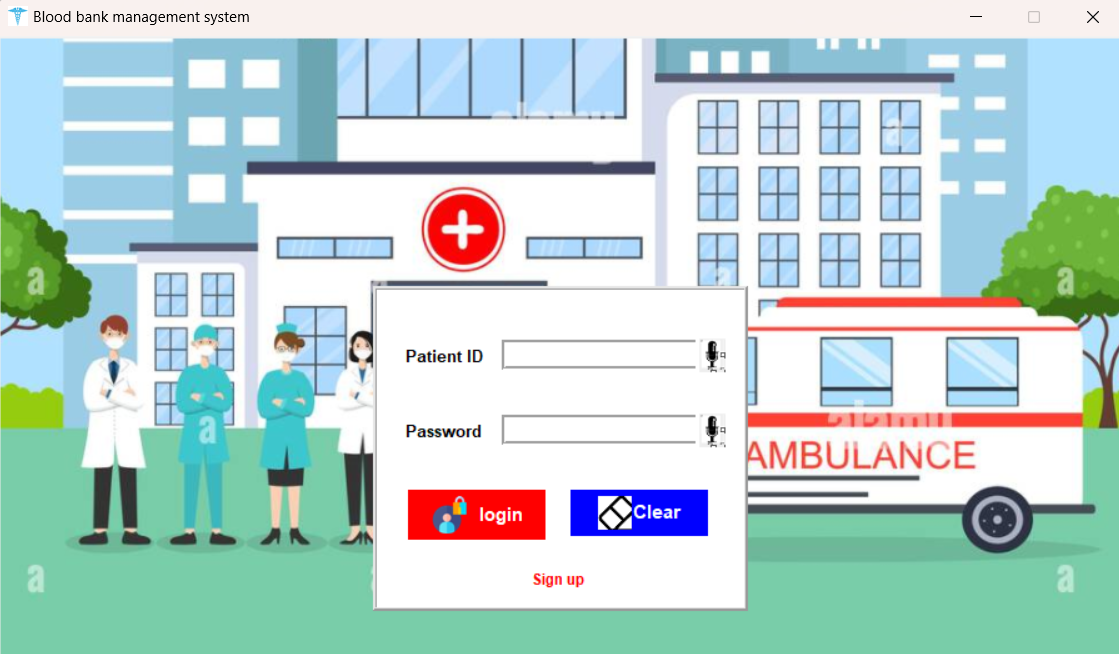
* Funds are needed because the initiative is entirely software-based and gives us the tools we need to use computers effectively. The money will only be required for the app's AWS deployment, which would run us about Rs. 7000.

### ANALYSIS AND FEATURE FINALIZATION SUBJECT TO CONSTRAINTS

It took a lot of thought to integrate the project's features and how. We have listed some of the features the app includes below, keeping in mind both the essential criteria every project must have and bringing forth some less well-executed ideas:

* + Feature 1: User Login

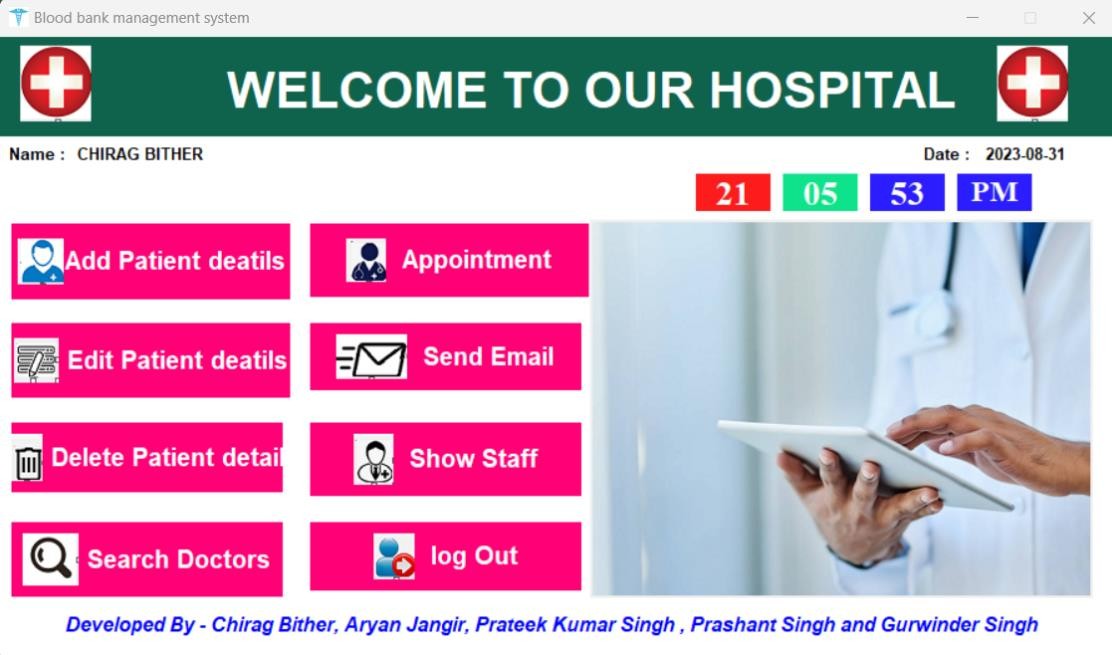
The login-based authentication is one of the key components of the E-Blood Bank Monitoring Hospitals application. The user creates an account for themselves and logs in with their user name and password. A dashboard will show up after logging in and remain visible until the user logs out. The user won't be able to read the content after you log out, and you'll have to go back to the login screen.



*Figure 1: USER Login authentication*

* + Feature 2: Interactive GUI

As you can see, the application's GUI is quite intriguing. The user will love using the app and working with the virtual helper SPARK.



*Figure 2: Interactive GUI*

* + Feature 3: User Sign-up

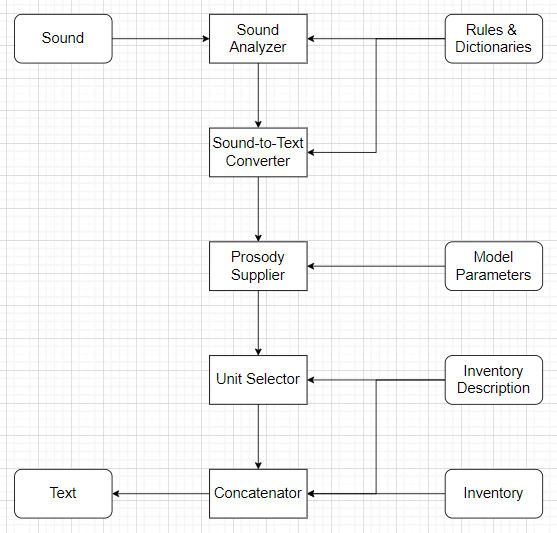
The sign-up feature is one of the most prevalent components seen in most applications. With the help of this programme, users can register for an account and carry out further tasks.



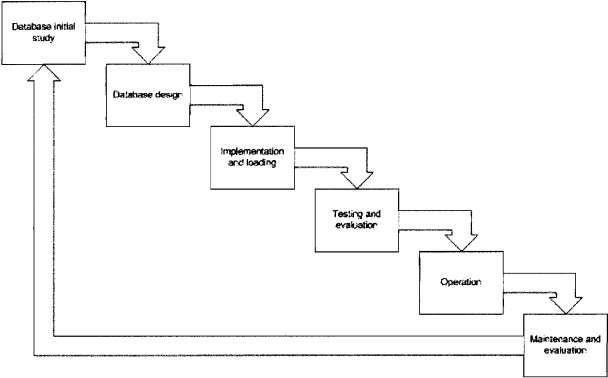
*Figure 3: Sign-up option*

### DESIGN FLOW

* **Block Diagram of text to speech converter:**



* **Scientific Diagram:**



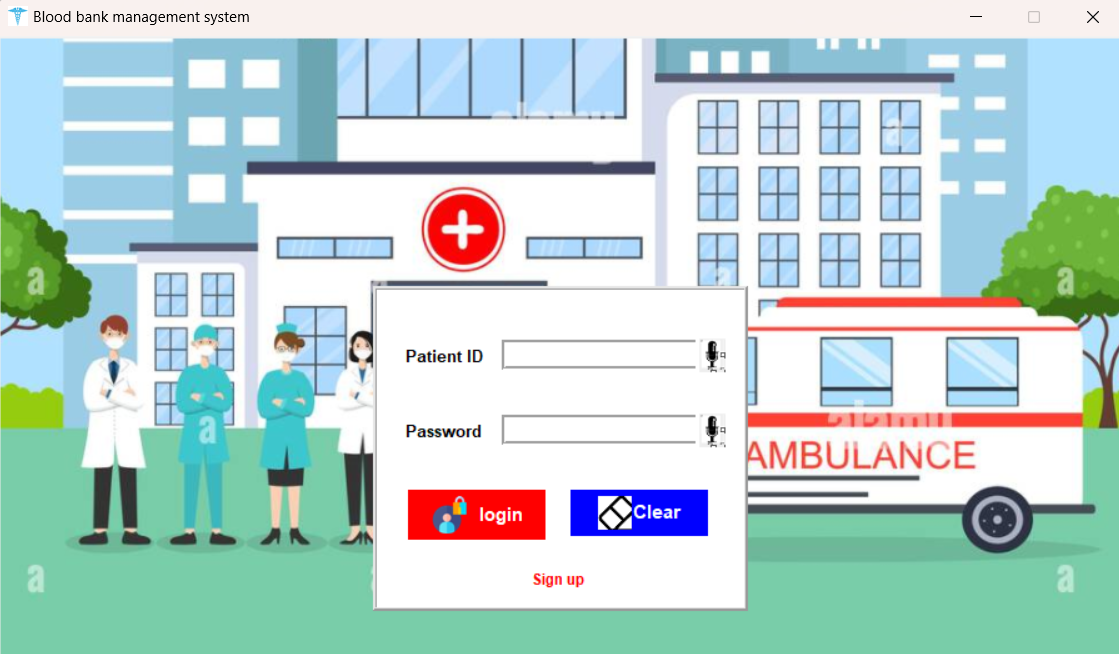
**CHAPTER IV: System Implementation**

### IMPLEMENTATION OF SOLUTION

It took a lot of thought to integrate the project's features and how. We have listed some of the features the app includes below, keeping in mind both the essential criteria every project must have and bringing forth some less well-executed ideas:

Feature 1: User Login

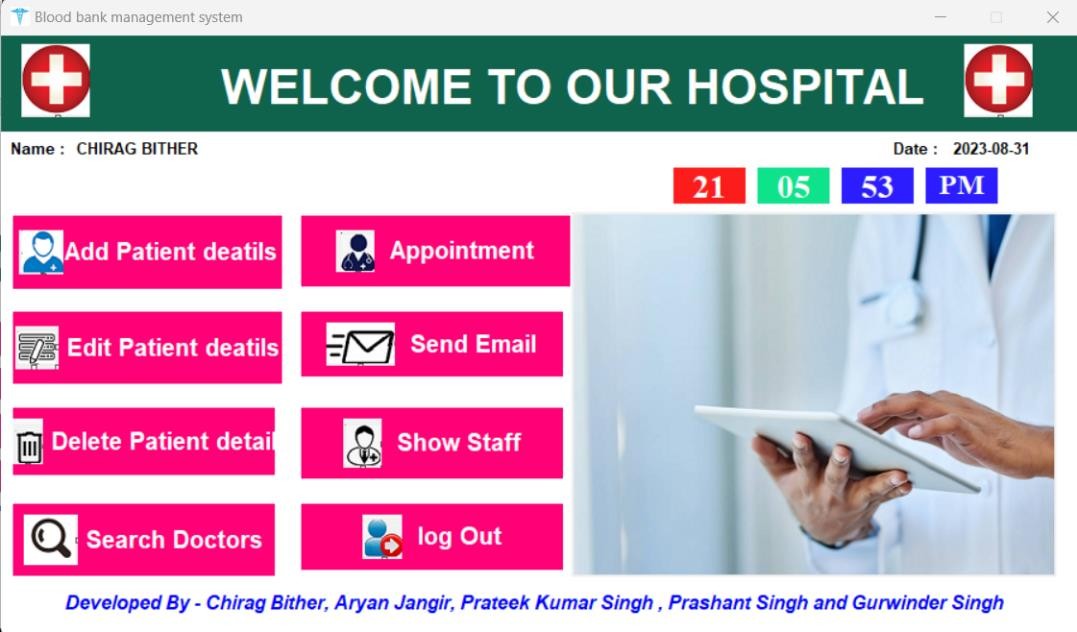
One of the main features of the blood bank system application isthe login- based authentication. The user registers himself/herself and logs in using his/her user-id and password. After logging in, a dashboard will be appeared which will persist until and unless a user either logs out. When you Log out, the user won’t be able to see the content and again you come in the login page.



*Figure 1: USER Login authentication*

Feature 2: Interactive GUI

As You can see the GUI of the application is quite engaging. The user will like to use the application and will appreciate the SPARK virtual assistant.



*Figure 2: Interactive GUI*

Feature 3: User Sign-up

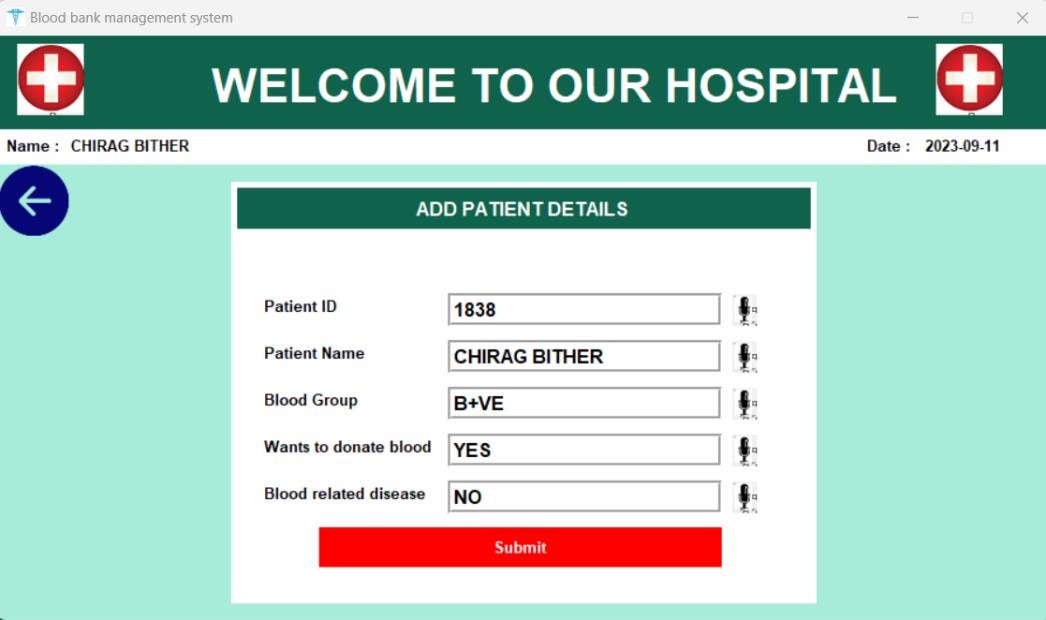
The sign-up feature is one of the most prevalent components seen in most applications. With the help of this programme, users can register for an account and carry out further tasks.



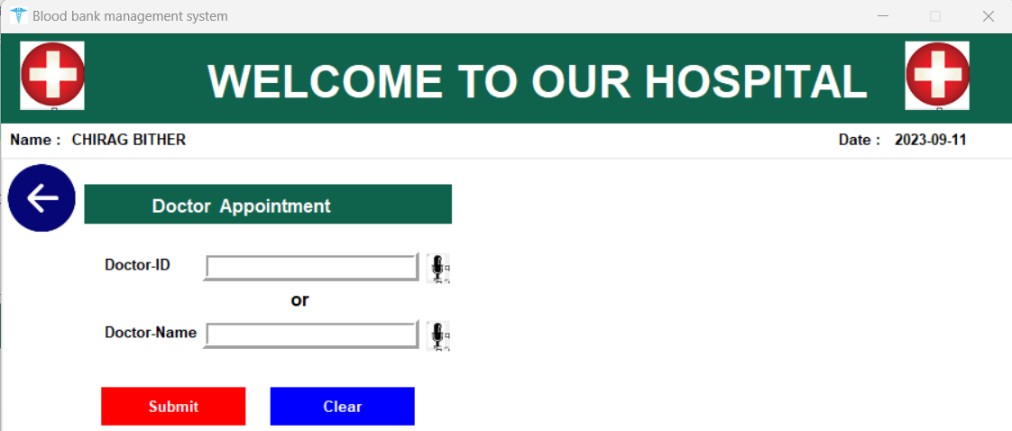
*Figure 3: Sign-up option*

Feature 4: several options for blood bank management Our E-Blood Bank Monitoring Hospitals contains numerous uses. There are options like Add patient details, Appointment, Edit patient detail, etc. Therefore, let's go over each choice individually.

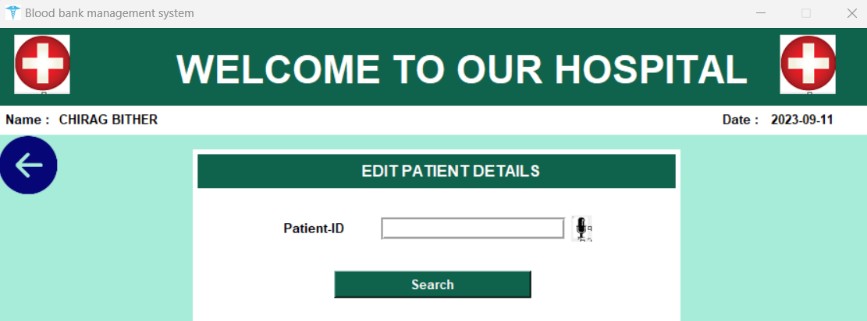
* + 1. **Add patient details**: In this we are able to register the patient details and save them in database.



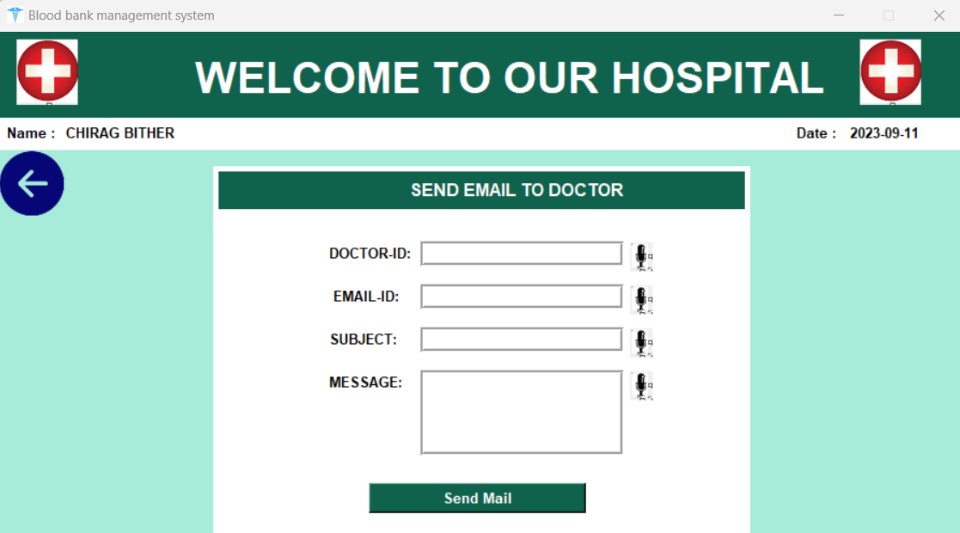
* + 1. **Appointment:** In this we are able to take the appointment from the doctor and doctor will see the patient according to that.



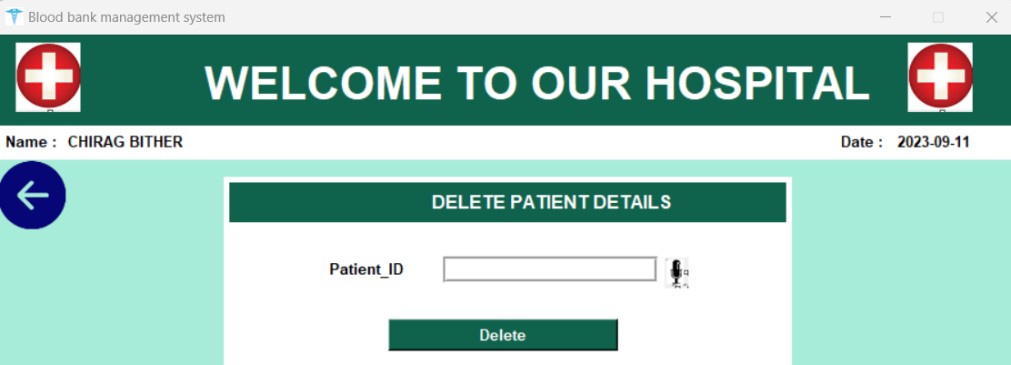
* + 1. **Edit patient details:** In this we are able to edit the patient details and save it in database.



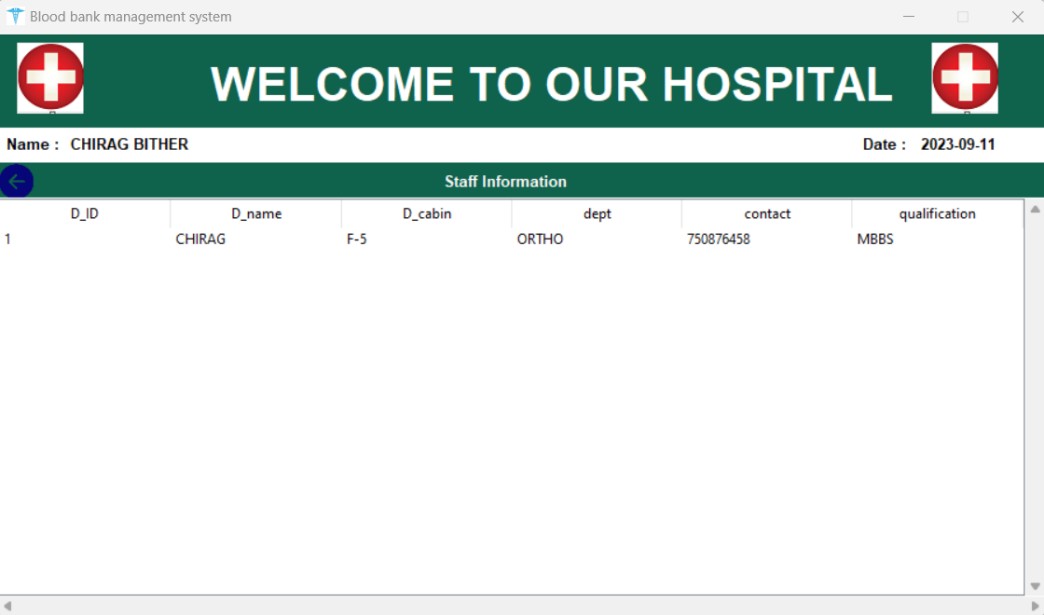
* + 1. **Send mail:** In this we are able to send the mail to doctor regarding to our problem.



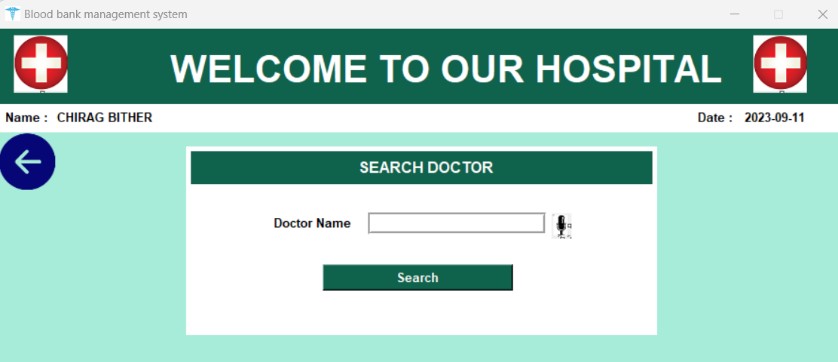
* + 1. **Delete patient details:** In this we are able to delete the unnecessary data of patient.



* + 1. **Show staff:** With the help of this we are able to see the staff of hospital or doctors in hospital.



* + 1. **Show doctor details:** With the help of this we are able to see the doctor details.

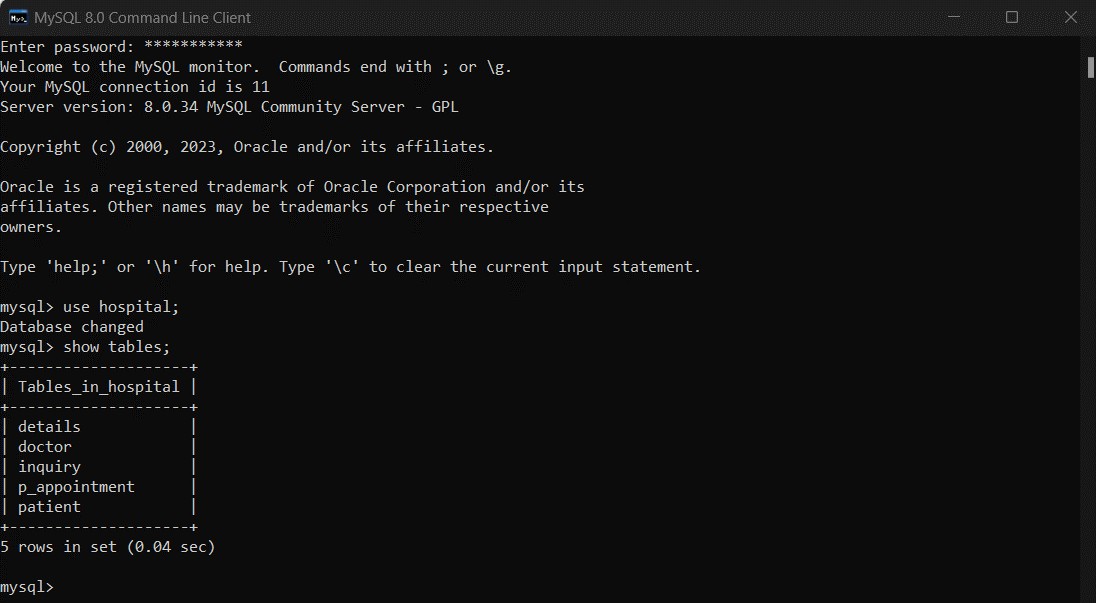


Feature 5: User Logout

The user can log out of the programme after using it completely by hitting the "Shutout" button located in the bottom left cornEr The user then arrives at the login page once more.

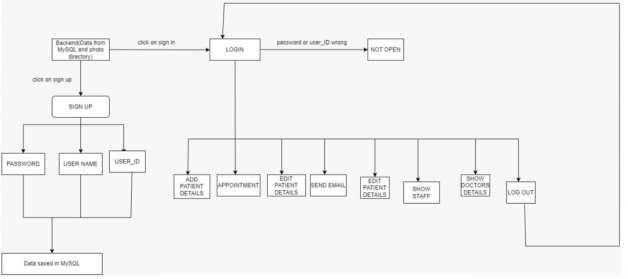


Feature 6: Data Storage

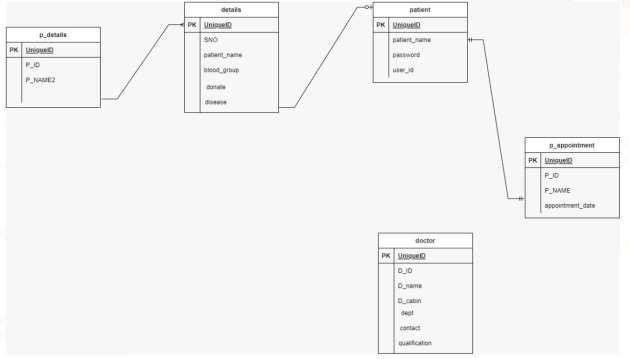
The app has login and user signup functions. All of the user's created data will remain in the database once the appropriate operations have been completed and the user has successfully logged out of the app. The content that was created by the user during their most recent login will be available to them when they log in again. In order to access the data, we use MySQL.

### Design Flow

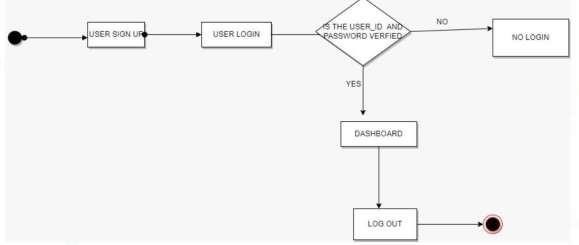
* **DFD Diagram:**



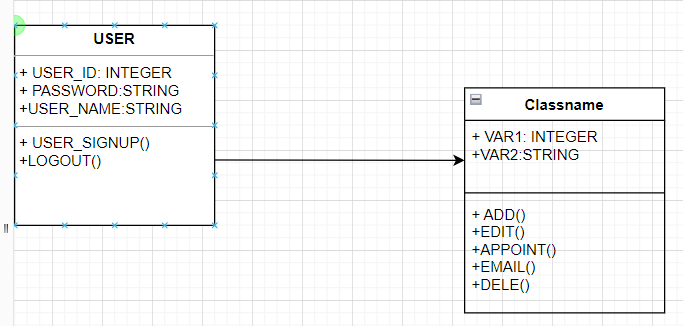
### ER Diagram:



* **Activity Diagram:**



### Class Diagram:



We start project planning after we determine whether the project is feasible. The table below demonstrates how my project was planned.

Table- Project Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | Task Name | Duration | Start | Finish |
| 2 | Planning | 25 days | 1/08/23 | 26/08/23 |
| 3 | Design | 20 days | 27/08/23 | 16/09/23 |
| 4 | Coding | 15 days | 18/09/23 | 09/10/23 |

Software a proactive strategy for reducing project-related uncertainty and potential loss is risk management. Product size, impact on business, customer-related factors, process risks, technology challenges, development environment uncertainties, staffing considerations (size and experience), project schedule, and cost are among the various categories of risks. The management of project risks involves the application of methods, techniques, and tools.

Identification of risks is a methodical endeavour to identify dangers to the project strategy. We may take the first step towards avoiding risks when it's possible and regulating them when it's required by recognising known and foreseeable dangers. We divided the risks into various categories in order to perform the risk identification, including:

* + 1. Project Risk
    2. Business Risk
    3. Technical Risk
    4. Pre-Known Risk
    5. Predictable Risk

Understanding the nature of the business and how it now functions is crucial before building any systems. In order to guarantee that all of the client's criteria are met throughout design, a complete investigation offers the precise data needed. The feasibility study serves as a major foundation for the research or study carried out throughout the analysis phase. It would be more accurate to argue that the feasibility and analysis phases overlap. The feasibility study is when high-level analysis starts. Contrary to popular belief, the system development life cycle (SDLC) does not include analysis as a phase. Initial system startup and system maintenance are the starting points for analysis. Analysis may be used for periodic maintenance even after the system has been successfully implemented.

## CHAPTER V: Conclusion and future work

* 1. **Conclusion:**

1. Objectives:

The main objective of this application is to provide an assistant for patient where patient can register and access the data for use anytime and anywhere. For many people it is very difficult to register themselves in hospital, so this Application willhelp the user to access the Data in Application. Best and accurate apps are also present in our application which will make the life of people easy. In several ways,working on this project has been enjoyable. The project's work as a whole has shed light on us in the following areas.

* + We now have a better understanding of how the HOSPITAL operates. This illustrates a common scenario from the real world.
  + Because suitable database design procedures must be followed in order to produce final results, our understanding of database design has improved.
  + Establishing a timeline for a project and sticking to it fosters effective time management.
  + A strong sense of teamwork and enhanced self-assurance in managing real-world projects have emerged.
  + There were issues with the validation at first, but after some discussion, we were able to implement validations.

1. Limitations of the system:

* This version does not support online payments.
* Not all sections have access to the data deletion and modify system.
* This system does not support accounts that have not been validated by SMS.
* Data may get lost as a result of poor management.

## Future Plan:

1. Enhanced Registration Process:

* In the future, we plan to improve the registration process by implementing features such as online registration, where users can fill out the necessary information and submit it electronically. This will eliminate the need for users to physically register at the hospital, making it more convenient for them.

1. Online Data Accessibility:

* We aim to develop a robust and secure online platform that allows patients to access their medical data anytime and anywhere. This will enable them to view their medical history, test results, prescriptions, and appointments through a secure login. Implementing strict privacy measures will ensure the confidentiality and integrity of the data.

1. Integration of Telemedicine:

* To further enhance patient care and convenience, we plan to integrate telemedicine capabilities into the application. This will allow patients to consult with healthcare professionals remotely, eliminating the need for unnecessary hospital visits for minor issues and improving accessibility for individuals in remote areas.

1. Mobile Application:

* In addition to the web-based application, we intend to develop a mobile application for iOS and Android devices. This will provide a more intuitive and

user-friendly experience for patients, enabling them to access their data and interact with the system seamlessly on their smartphones or tablets.

1. Expansion to Other Healthcare Facilities:

* Once the system is successfully implemented in one hospital, we aim to expand its usage to other healthcare facilities. This will involve customizing the system to meet the specific requirements and workflows of different hospitals, clinics, and medical centers, thereby benefiting a larger population of patients.

### REFERENCES:

1. Z. Liu, "Design and Implementation of Hospital Emergency Nursing Information Management System," 2016 International Conference on Smart City and Systems Engineering (ICSCSE), Hunan, 2016, pp. 218-221.
2. GB. Koyuncu and H. Koyuncu, "Intelligent Hospital Management System (IHMS),"2015 International Conference on Computational Intelligence and Communication Networks (CICN), Jabalpur, 2015, pp. 1602-1604.
3. Healthcare management system and domain search of nearest Medical services by Ruchi Dumbre, Purva Raut, Bhagyshree mahamuni, Priyanka Khose, Prof.Jagruti Wagh. IJISET -International Journal of Innovative Science, Engineering & Technology, Vol. 3 Issue 3, March 2016.
4. HOSPITAL MANAGEMENT SYSTEM by Digvijay H. Gadhari, Yadnyesh P. Kadam, Prof. Parineeta Suman Department of Computer Engineering, Saraswati College of Engineering, Kharghar, Mumbai, Maharashtra, India. International Journal for Research in Engineering Application & Management (IJREAM)-2016.
5. Asabe.S.A., Oye, N. D., Monday Goji,” HOSPITAL PATIENT DATABASE MANAGEMENT SYSTEM”, COMPUSOFT An International journal of Advance Computer Technology.
6. Fellegi, A and mooney, S. (1998). “Population and higher individual standard for the quality of life”.
7. Miller, R. J. (1994). “Modernizing Health care through Electronic Medical Record “Information System”.
8. Brown, P.J. (2000). “Evaluation of the quality of information retrieval of clinics finding from a COMPUSOFT”, An international journal of advanced computer technology, 2 (3), March -2013 (Volume-II, IssueIII) 70 computerized patient database using a semantic technological.
9. Laubbel, A. (1998) “Define medical health care, or medical documentation of A patient history and care.
10. Schneider, K., & Wagner, I. (1993). Constructing the Dossier Representative. Computer-Based Information Sharing in French Hospitals. In: Computer Supported

Cooperation Work, NO 1. Kluwer Academic Publishers.

1. Laing K. (2002). The benefits and challenges of the computerized electronic Medical record (Web page retrieved February 20, 2007 from the world wide web. [http://en](http://en/) wikipedia.org/wiki/medical prescription.

## APPENDIX

**hospital.py**

from tkinter import \*

from datetime import date

from tkinter import ttk

from tkinter import messagebox

from tkcalendar import \*

import time

import datetime

from datetime import datetime

from datetime import timedelta

import smtplib

import mysql.connector

import pyttsx3

import os

import webbrowser

import datetime

import speech\_recognition as sr

root = Tk()

root.title("Blood bank management system")

root.iconbitmap("aa.ico")

root.geometry("900x500+300+150")

root.resizable(0, 0)

engine = pyttsx3.init('sapi5')

voices = engine.getProperty('voices')

engine.setProperty('voice', voices[0].id)

def speak(audio):

engine.say(audio)

engine.runAndWait()

def wishMe():

hour = int(datetime.datetime.now().hour)

if hour>=0 and hour<12:

speak("Good Morning!")

elif hour>=12 and hour<18:

speak("Good Afternoon!")

else:

speak("Good Evening!")

speak("Welcome to our Blood bank management system. This project is made by Chirag Bither , Aryan Jangir , Prateek Kumar Singh , Prashant Singh and Gurwinder Singh")

class maincode:

def takeCommand1(self):

import speech\_recognition as sr

r = sr.Recognizer()

mic = sr.Microphone(device\_index=1)

with mic as source:

r.adjust\_for\_ambient\_noise(source, duration=1)

audio = r.listen(source, timeout=5)

try:

SQ2 = r.recognize\_google(audio, language='English')

except:

messagebox.showerror('Blood bank management system', 'Sorry! Not able to understand')

pass

return SQ2

def login(self):

self.var1 = self.e1.get()

self.var2 = self.e2.get()

import mysql.connector

mydb=mysql.connector.connect(host='localhost',user='root',password='Chirag@1302',database='hospital')

cursor=mydb.cursor()

cursor.execute("SELECT \* FROM patient WHERE Patient\_ID='"+self.var1+"' and Password='"+self.var2+"'")

self.ab = cursor.fetchone()

if self.ab!=None:

self.under\_fm=Frame(root,height=500,width=900,bg='#fff')

self.under\_fm.place(x=0,y=0)

self.fm2=Frame(root,bg='#0f624c',height=80,width=900)

self.fm2.place(x=0,y=0)

self.lbb=Label(self.fm2,bg='#0f624c')

self.lbb.place(x=15,y=5)

self.ig=PhotoImage(file='hospital.png')

self.lbb.config(image=self.ig)

self.lb3=Label(self.fm2,text='WELCOME TO OUR HOSPITAL',fg='White',bg='#0f624c',font=('Arial',30,'bold'))

self.lb3.place(x=180,y=17)

self.lbb5=Label(self.fm2,bg='#0f624c')

self.lbb5.place(x=800,y=5)

self.ig3=PhotoImage(file='hospital.png')

self.lbb5.config(image=self.ig3)

self.name=Label(root,text="Name : ",bg='#fff',fg="black",font=('Arial',10,'bold'))

self.name.place(x=5,y=83)

self.name1=Label(root,text=self.ab[1],fg='black',bg='#fff',font=('Arial',10,'bold'))

self.name1.place(x=60,y=83)

self.today=date.today()

self.dat=Label(root,text='Date : ',bg='#fff',fg='black',font=('Arial',10,'bold'))

self.dat.place(x=740,y=83)

self.dat2 = Label(root, text=self.today, bg='#fff', fg='black', font=('Arial', 10, 'bold'))

self.dat2.place(x=790, y=83)

self.cur()

else:

messagebox.showerror('Blood bank management system', 'Your ID or Password is not Valid')

def cur(self):

self.fm3=Frame(root,bg='#fff',width=900,height=390)

self.fm3.place(x=0,y=110)

def clock():

h = str(time.strftime("%H"))

m = str(time.strftime("%M"))

s = str(time.strftime("%S"))

if int(h) >=12 and int(m) >=0:

self.lb7\_hr.config(text="PM")

self.lb1\_hr.config(text=h)

self.lb3\_hr.config(text=m)

self.lb5\_hr.config(text=s)

self.lb1\_hr.after(200, clock)

self.lb1\_hr = Label(self.fm3, text='12', font=('times new roman', 20, 'bold'), bg='#fc1c1c', fg='white')

self.lb1\_hr.place(x=560, y=0, width=60, height=30)

self.lb3\_hr = Label(self.fm3, text='05', font=('times new roman', 20, 'bold'), bg='#0ee38b', fg='white')

self.lb3\_hr.place(x=630, y=0, width=60, height=30)

self.lb5\_hr = Label(self.fm3, text='37', font=('times new roman', 20, 'bold'), bg='#2b1dff', fg='white')

self.lb5\_hr.place(x=700, y=0, width=60, height=30)

self.lb7\_hr = Label(self.fm3, text='AM', font=('times new roman', 17, 'bold'), bg='#2b1dff', fg='white')

self.lb7\_hr.place(x=770, y=0, width=60, height=30)

clock()

self.canvas8 = Canvas(self.fm3, bg='black', width=400, height=300)

self.canvas8.place(x=475, y=37)

self.photo9=PhotoImage(file="bb.png")

self.canvas8.create\_image(0,0,image=self.photo9,anchor=NW)

self.develop=Label(self.fm3,text='Developed By - Chirag Bither, Aryan Jangir, Prateek Kumar Singh , Prashant Singh and Gurwinder Singh',bg='#fff',fg='blue',

font=('Cursive',12,'italic','bold'))

self.develop.place(x=50,y=350)

self.bt1=Button(self.fm3,text='Add Patient deatils',fg='#fff',bg='#ff0076',font=('Arial',15,'bold'),width=200,

height=-3,bd=10,relief='flat',command=self.addp,cursor='hand2')

self.bt1.place(x=10,y=40)

self.logo = PhotoImage(file='bt1.png')

self.bt1.config(image=self.logo, compound=LEFT)

self.small\_logo = self.logo.subsample(1,1)

self.bt1.config(image=self.small\_logo)

self.bt2 = Button(self.fm3, text=' Appointment', fg='#fff', bg='#ff0076', font=('Arial', 15, 'bold'),

width=200,height=-3, bd=10,relief='flat',command=self.appoint,cursor='hand2')

self.bt2.place(x=250, y=40)

self.log = PhotoImage(file='bt2.png')

self.bt2.config(image=self.log, compound=LEFT)

self.small\_log = self.log.subsample(1, 1)

self.bt2.config(image=self.small\_log)

self.bt3 = Button(self.fm3, text=' Edit Patient deatils', fg='#fff', bg='#ff0076', font=('Arial', 15, 'bold'),

width=200,height=-3,bd=10,relief='flat',cursor='hand2',command=self.edit)

self.bt3.place(x=10, y=120)

self.logb = PhotoImage(file='bt3.png')

self.bt3.config(image=self.logb, compound=LEFT)

self.small\_logb = self.logb.subsample(1, 1)

self.bt3.config(image=self.small\_logb)

self.bt4 = Button(self.fm3, text=' Send Email', fg='#fff', bg='#ff0076', font=('Arial', 15, 'bold'),

width=200,height=-3,bd=7,relief='flat',cursor='hand2',command=self.email)

self.bt4.place(x=250, y=120)

self.log4 = PhotoImage(file='bt4.png')

self.bt4.config(image=self.log4, compound=LEFT)

self.small\_log4 = self.log4.subsample(1, 1)

self.bt4.config(image=self.small\_log4)

self.bt5 = Button(self.fm3, text=' Delete Patient detail', fg='#fff', bg='#ff0076', font=('Arial', 15, 'bold'),

width=200,height=-3,bd=7,relief='flat',cursor='hand2',command=self.delete)

self.bt5.place(x=10, y=200)

self.log5 = PhotoImage(file='bt5.png')

self.bt5.config(image=self.log5, compound=LEFT)

self.small\_log5 = self.log5.subsample(1, 1)

self.bt5.config(image=self.small\_log5)

self.bt6 = Button(self.fm3, text=' Show Staff', fg='#fff', bg='#ff0076', font=('Arial', 15, 'bold'),

width=200,height=-3,bd=7, relief='flat',cursor='hand2',command=self.show)

self.bt6.place(x=250, y=200)

self.log6 = PhotoImage(file='bt6.png')

self.bt6.config(image=self.log6, compound=LEFT)

self.small\_log6 = self.log6.subsample(1, 1)

self.bt6.config(image=self.small\_log6)

self.bt7 = Button(self.fm3, text=' Search Doctors', fg='#fff', bg='#ff0076', font=('Arial', 15, 'bold'),

width=200,height=0,bd=7, relief='flat',cursor='hand2',command=self.search)

self.bt7.place(x=10, y=280)

self.log7 = PhotoImage(file='bt7.png')

self.bt7.config(image=self.log7, compound=LEFT)

self.small\_log7 = self.log7.subsample(1, 1)

self.bt7.config(image=self.small\_log7)

try:

self.bt8 = Button(self.fm3, text=' log Out', fg='#fff', bg='#ff0076', font=('Arial', 15, 'bold'),

width=200,

height=-3, bd=7, relief='flat',cursor='hand2',command=self.code)

self.bt8.place(x=250, y=280)

self.log8 = PhotoImage(file='bt8.png')

self.bt8.config(image=self.log8, compound=LEFT)

self.small\_log8 = self.log8.subsample(1, 1)

self.bt8.config(image=self.small\_log8)

except:

self.bt9 = ttk.Button(self.fm3, text="ram", bg='#11d09a', font=('Arial', 15, 'bold'), width=150,

height=0)

self.bt9.place(x=10, y=350)

self.log9 = PhotoImage(file='bt8.png')

self.bt9.config(image=self.log9, compound=LEFT)

self.small\_log9 = self.log9.subsample(3, 3)

self.bt9.config(image=self.small\_log9)

def mainclear(self):

self.e1.delete(0,END)

self.e2.delete(0,END)

def addp(self):

class temp(maincode):

def pa(self):

self.fm=Frame(root,bg='#a7ecd9',width=900,height=390)

self.fm.place(x=0,y=110)

self.fm1=Frame(self.fm,bg='#fff',width=500,height=360,bd=5,relief='flat')

self.fm1.place(x=200,y=15)

self.backbt = Button(self.fm, width=60, bg='#a7ecd9',activebackground='#a7ecd9', bd=0, relief='flat',\

command=self.cur)

self.backbt.place(x=0, y=0)

self.log = PhotoImage(file='back.png')

self.backbt.config(image=self.log, compound=LEFT)

self.small\_log0 = self.log.subsample(1, 1)

self.backbt.config(image=self.small\_log0)

self.f=Frame(self.fm1,bg='#0f624c',width=490,height=35)

self.f.place(x=0,y=0)

self.ll=Label(self.f,text='ADD PATIENT DETAILS',fg='#fff',bg='#0f624c',font=('Arial',12,'bold'))

self.ll.place(x=150,y=6)

self.lb=Label(self.fm1,text='Patient ID',fg='black',bg='#fff',font=('Arial',10,'bold'))

self.lb.place(x=20,y=90)

self.lb2 = Label(self.fm1, text='Patient Name', fg='black', bg='#fff', font=('Arial', 10, 'bold'))

self.lb2.place(x=20, y=130)

self.lb3 = Label(self.fm1, text='Blood Group', fg='black', bg='#fff', font=('Arial', 10, 'bold'))

self.lb3.place(x=20, y=170)

self.lb4= Label(self.fm1, text='Wants to donate blood', fg='black', bg='#fff', font=('Arial', 10, 'bold'))

self.lb4.place(x=20, y=210)

self.lb5 = Label(self.fm1, text='Blood related disease', fg='black', bg='#fff', font=('Arial', 10, 'bold'))

self.lb5.place(x=20, y=250)

self.ee1=Entry(self.fm1,width=25,bd=4,relief='groove',font=('arial',12,'bold'))

self.ee1.place(x=180,y=90)

self.ee2=Entry(self.fm1,width=25,bd=4,relief='groove',font=('arial',12,'bold'))

self.ee2.place(x=180,y=130)

self.ee3=Entry(self.fm1,width=25,bd=4,relief='groove',font=('arial',12,'bold'))

self.ee3.place(x=180,y=170)

self.ee4=Entry(self.fm1,width=25,bd=4,relief='groove',font=('arial',12,'bold'))

self.ee4.place(x=180,y=210)

self.ee5=Entry(self.fm1,width=25,bd=4,relief='groove',font=('arial',12,'bold'))

self.ee5.place(x=180,y=250)

self.bt=Button(self.fm1,text='Submit',width=41,bg='red',fg='#fff',font=('Arial',10,'bold'),bd=5,

relief='flat',command=self.submit1)

self.bt.place(x=70,y=290)

self.mike1 = Button(self.fm1, width=25, bg='white',activebackground='white', bd=0, relief='flat',

command=lambda:self.ee1.insert(END, self.takeCommand()))

self.mike1.place(x=420, y=90)

self.log = PhotoImage(file='mike.png')

self.mike1.config(image=self.log, compound=LEFT)

self.small\_log = self.log.subsample(1, 1)

self.mike1.config(image=self.small\_log)

self.mike1 = Button(self.fm1, width=25, bg='white',activebackground='white', bd=0, relief='flat',

command=lambda:self.ee2.insert(END, self.takeCommand()))

self.mike1.place(x=420, y=130)

self.log = PhotoImage(file='mike.png')

self.mike1.config(image=self.log, compound=LEFT)

self.small\_log1 = self.log.subsample(1, 1)

self.mike1.config(image=self.small\_log1)

self.mike2 = Button(self.fm1, width=25, bg='white',activebackground='white', bd=0, relief='flat',

command=lambda:self.ee3.insert(END, self.takeCommand()))

self.mike2.place(x=420, y=170)

self.log = PhotoImage(file='mike.png')

self.mike2.config(image=self.log, compound=LEFT)

self.small\_log2 = self.log.subsample(1, 1)

self.mike2.config(image=self.small\_log2)

self.mike3 = Button(self.fm1, width=25, bg='white',activebackground='white', bd=0, relief='flat',

command=lambda:self.ee4.insert(END, self.takeCommand()))

self.mike3.place(x=420, y=210)

self.log = PhotoImage(file='mike.png')

self.mike3.config(image=self.log, compound=LEFT)

self.small\_log3 = self.log.subsample(1, 1)

self.mike3.config(image=self.small\_log3)

self.mike4 = Button(self.fm1, width=25, bg='white',activebackground='white', bd=0, relief='flat',

command=lambda:self.ee5.insert(END, self.takeCommand()))

self.mike4.place(x=420, y=250)

self.log = PhotoImage(file='mike.png')

self.mike4.config(image=self.log, compound=LEFT)

self.small\_log4 = self.log.subsample(1, 1)

self.mike4.config(image=self.small\_log4)

def takeCommand(self):

import speech\_recognition as sr

r = sr.Recognizer()

mic = sr.Microphone(device\_index=1)

with mic as source:

r.adjust\_for\_ambient\_noise(source, duration=1)

audio = r.listen(source, timeout=5)

try:

SQ1 = r.recognize\_google(audio, language='English')

except:

messagebox.showerror('Blood bank management system', 'Sorry! Not able to uderstand')

pass

return SQ1

def submit1(self):

self.id=self.ee1.get()

self.ttl=self.ee2.get()

self.aut=self.ee3.get()

self.edi=self.ee4.get()

self.pri=self.ee5.get()

import mysql.connector

mydb=mysql.connector.connect(host='localhost',user='root',password='Chirag@1302',database='hospital')

cursor=mydb.cursor()

s="INSERT INTO details (SNO,patient\_name,blood\_group,donate,disease) VALUES(%s,%s,%s,%s,%s)"

b1=(self.id,self.ttl,self.aut,self.edi,self.pri)

cursor.execute(s,b1)

mydb.commit()

self.clear()

def clear(self):

self.ee1.delete(0,END)

self.ee2.delete(0,END)

self.ee3.delete(0,END)

self.ee4.delete(0,END)

self.ee5.delete(0,END)

obj=temp()

obj.pa()

def appoint(self):

class test(maincode):

def issue(self):

self.f = Frame(root, bg='#a7ecd9', width=900, height=390)

self.f.place(x=0, y=110)

self.fmi=Canvas(self.f,bg='#fff',width=900,height=390,bd=0,relief='flat')

self.fmi.place(x=0,y=0)

self.fc=Frame(self.fmi,bg='#fff',width=330,height=230,bd=4,relief='flat')

self.fc.place(x=70,y=20)

self.ffb=Frame(self.fc,bg='#0f624c',bd=2,relief='flat',width=330,height=35)

self.ffb.place(x=0,y=0)

self.lc=Label(self.ffb,text='Doctor Appointment',bg='#0f624c',fg='#fff',font=('Arial',12,'bold'))

self.lc.place(x=55,y=5)

self.lb=Label(self.fc,text='Doctor-ID',bg='#fff',fg='black',font=('Arial',10,'bold'))

self.lb.place(x=15,y=60)

self.ob=Label(self.fc,text='or',bg='#fff',fg='black',font=('cursive',12,'bold'))

self.ob.place(x=180,y=90)

self.em = Entry(self.fc, width=30, bd=5, relief='ridge', font=('Arial', 8, 'bold'))

self.em.place(x=105, y=60)

self.lb = Label(self.fc, text='Doctor-Name', bg='#fff', fg='black', font=('Arial', 10, 'bold'))

self.lb.place(x=15, y=120)

self.em2 = Entry(self.fc, width=30, bd=5, relief='ridge', font=('Arial', 8, 'bold'))

self.em2.place(x=105, y=120)

self.bt = Button(self.fc, text='Submit', width=14, bg='red', fg='#fff', font=('Arial', 10, 'bold'),

bd=5,relief='flat',command=self.check)

self.bt.place(x=15,y=180)

self.bt3=Button(self.fc,text='Clear',width=14,bg='blue',fg='#fff',font=('arial',10,'bold'),bd=5,

relief='flat',command=self.clr)

self.bt3.place(x=165,y=180)

self.backbt = Button(self.fmi,width=60, bg='#fff',activebackground='#fff',bd=0, relief='flat',

command=self.cur)

self.backbt.place(x=5, y=5)

self.log = PhotoImage(file='back.png')

self.backbt.config(image=self.log, compound=LEFT)

self.small\_log = self.log.subsample(1, 1)

self.backbt.config(image=self.small\_log)

self.mike1 = Button(self.fc, width=25, bg='white',activebackground='white', bd=0, relief='flat',

command=lambda:self.em.insert(END, self.takeCommand()))

self.mike1.place(x=300, y=60)

self.log = PhotoImage(file='mike.png')

self.mike1.config(image=self.log, compound=LEFT)

self.small\_log1 = self.log.subsample(1, 1)

self.mike1.config(image=self.small\_log1)

self.mike2 = Button(self.fc, width=25, bg='white',activebackground='white', bd=0, relief='flat',

command=lambda:self.em2.insert(END, self.takeCommand()))

self.mike2.place(x=300, y=120)

self.log = PhotoImage(file='mike.png')

self.mike2.config(image=self.log, compound=LEFT)

self.small\_log2 = self.log.subsample(1, 1)

self.mike2.config(image=self.small\_log2)

def takeCommand(self):

import speech\_recognition as sr

r = sr.Recognizer()

mic = sr.Microphone(device\_index=1)

with mic as source:

r.adjust\_for\_ambient\_noise(source, duration=1)

audio = r.listen(source, timeout=5)

try:

SQ1 = r.recognize\_google(audio, language='English')

except:

messagebox.showerror('Blood bank management system', 'Sorry! Not able to uderstand')

pass

return SQ1

def check(self):

self.ai=self.em.get()

self.b2=self.em2.get()

import mysql.connector

mydb=mysql.connector.connect(host='localhost',user='root',password='Chirag@1302',database='hospital')

cursor=mydb.cursor()

cursor.execute("SELECT \* FROM doctor WHERE D\_ID='"+self.ai+"' or D\_name='"+self.b2+"'")

self.var=cursor.fetchone()

if self.var!=None:

self.lb1=Label(self.fmi,text='Name :',fg='black',font=('Arial',10,'bold'))

self.lb1.place(x=60,y=255)

self.lb2 = Label(self.fmi, text=self.var[1], fg='black', font=('Arial', 10, 'bold'))

self.lb2.place(x=150, y=255)

self.lb3 = Label(self.fmi, text='Cabin :',fg='black', font=('Arial', 10, 'bold'))

self.lb3.place(x=60, y=275)

self.lb4 = Label(self.fmi, text=self.var[2],fg='black', font=('Arial', 10, 'bold'))

self.lb4.place(x=150, y=275)

self.lb5 = Label(self.fmi, text='Department:', fg='black', font=('Arial', 10, 'bold'))

self.lb5.place(x=60, y=295)

self.lb6 = Label(self.fmi, text=self.var[3], fg='black', font=('Arial', 10, 'bold'))

self.lb6.place(x=150, y=295)

self.lb7 = Label(self.fmi, text='Contact :', fg='black', font=('Arial', 10, 'bold'))

self.lb7.place(x=60, y=315)

self.lb8 = Label(self.fmi, text=self.var[4],fg='black', font=('Arial', 10, 'bold'))

self.lb8.place(x=150, y=315)

self.lb9 = Label(self.fmi, text='Qualification :', fg='black', font=('Arial', 10, 'bold'))

self.lb9.place(x=60, y=335)

self.lb10 = Label(self.fmi, text=self.var[5],fg='black', font=('Arial', 10, 'bold'))

self.lb10.place(x=152, y=335)

self.fr=Frame(self.fmi,bg='#fff',bd=5,relief='flat',width=450,height=320)

self.fr.place(x=420,y=20)

self.ff=Frame(self.fr,bg='#0f624c',bd=2,relief='flat',width=450,height=35)

self.ff.place(x=0,y=0)

self.lb=Label(self.ff,text='Patient Detail',bg='#0f624c',fg='#fff',font=('Arial',12,'bold'))

self.lb.place(x=165,y=5)

self.tt=Label(self.fr,text='Patient-ID',bg='#fff',fg='black',font=('arial',10,'bold'))

self.tt.place(x=50,y=60)

self.e1 = Entry(self.fr, width=30, bd=5, relief='ridge', font=('Arial', 8, 'bold'))

self.e1.place(x=160, y=60)

self.ttp = Label(self.fr, text='Patient-Name', bg='#fff', fg='black', font=('arial', 10, 'bold'))

self.ttp.place(x=50, y=110)

self.e2 = Entry(self.fr, width=30, bd=5, relief='ridge', font=('Arial', 8, 'bold'))

self.e2.place(x=160, y=110)

self.bt1 = Button(self.fr, text='Submit', width=35, bg='#0f624c', fg='#fff', font=('Arial', 10,

'bold'),bd=5,relief='flat',command=self.data)

self.bt1.place(x=60, y=160)

self.bt1 = Button(self.fr, text='Submit', width=35, bg='#0f624c', fg='#fff', font=('Arial', 10,

'bold'), bd=5,relief='flat',command=self.data, state=ACTIVE)

self.bt1.place(x=60, y=160)

self.mike10 = Button(self.fr, width=25, bg='white',activebackground='white', bd=0, relief='flat',

command=lambda:self.e1.insert(END, self.takeCommand()))

self.mike10.place(x=350, y=60)

self.log = PhotoImage(file='mike.png')

self.mike10.config(image=self.log, compound=LEFT)

self.small\_log10 = self.log.subsample(1, 1)

self.mike10.config(image=self.small\_log10)

self.mike11 = Button(self.fr, width=25, bg='white',activebackground='white', bd=0, relief='flat',

command=lambda:self.e2.insert(END, self.takeCommand()))

self.mike11.place(x=350, y=110)

self.log = PhotoImage(file='mike.png')

self.mike11.config(image=self.log, compound=LEFT)

self.small\_log11 = self.log.subsample(1, 1)

self.mike2.config(image=self.small\_log11)

def clr(self):

self.em.delete(0, END)

self.em2.delete(0, END)

def data(self):

self.vva=self.e1.get()

self.vvb=self.e2.get()

import mysql.connector

mydb=mysql.connector.connect(host='localhost',user='root',password='Chirag@1302',database='hospital')

cursor=mydb.cursor()

s="INSERT INTO p\_details (P\_ID,P\_NAME) VALUES(%s,%s)"

b1=(self.vva,self.vvb)

cursor.execute(s,b1)

mydb.commit()

self.boot=Tk()

self.boot.title("Select the date")

self.boot.iconbitmap("aa.ico")

self.boot.configure(bg='#fff')

self.boot.geometry("300x500+1202+50")

self.boot.resizable(0,0)

self.x = date.today()

self.cal = Calendar(self.boot, selectmode="day", bg='black',year=2020,month=9,day=6)

self.cal.place(x=20,y=200)

btn1 = Button(self.boot, text="Confirm Date",command=self.get\_data, bg='#ff0076',

font=('arial', 10, 'bold'),

fg='#fff', relief='flat')

btn1.place(x=90,y=400)

self.boot.mainloop()

def get\_data(self):

self.datecon=self.cal.selection\_get()

self.vva=self.e1.get()

self.vvb=self.e2.get()

import mysql.connector

mydb=mysql.connector.connect(host='localhost',user='root',password='Chirag@1302',database='hospital')

cursor=mydb.cursor()

s2="INSERT INTO p\_appointment (P\_ID,P\_NAME,appointment\_date) VALUES(%s,%s,%s)"

b2=(self.vva,self.vvb,self.datecon)

cursor.execute(s2,b2)

mydb.commit()

messagebox.showinfo("Blood management system","You successfully get an Appointment !")

obk=test()

obk.issue()

def edit(self):

class editing(maincode):

def edbooks(self):

self.ffm=Frame(root,bg='#a7ecd9',width=900,height=390)

self.ffm.place(x=0,y=110)

self.fm1 = Frame(self.ffm, bg='#fff', width=500, height=200, bd=5, relief='flat')

self.fm1.place(x=200, y=15)

self.ed = Frame(self.fm1, bg='#0f624c', bd=0, relief='flat', width=490, height=35)

self.ed.place(x=0,y=0)

self.lab = Label(self.ed, text='EDIT PATIENT DETAILS', bg='#0f624c', fg='#fff', font=('Arial', 12,

'bold'))

self.lab.place(x=165, y=5)

self.label3=Label(self.fm1,text='Patient-ID',bg='#fff',fg='black',font=('arial',10,'bold'))

self.label3.place(x=85,y=65)

self.entry=Entry(self.fm1,width=30,bd=4,relief='groove',font=('arial',8,'bold'))

self.entry.place(x=188,y=65)

self.button7 = Button(self.fm1, text='Search', bg='#0f624c', fg='#fff', width=24, height=0,

font=('Arial', 10, 'bold'),command=self.search)

self.button7.place(x=140,y=120)

self.backbt = Button(self.ffm, width=60, bg='#a7ecd9',activebackground='#a7ecd9',

bd=0, relief='flat', command=self.cur)

self.backbt.place(x=0, y=0)

self.log = PhotoImage(file='back.png')

self.backbt.config(image=self.log, compound=LEFT)

self.small\_log = self.log.subsample(1, 1)

self.backbt.config(image=self.small\_log)

self.mike1 = Button(self.fm1, width=25, bg='white',activebackground='white', bd=0, relief='flat',

command=lambda:self.entry.insert(END, self.takeCommand()))

self.mike1.place(x=380, y=62)

self.log = PhotoImage(file='mike.png')

self.mike1.config(image=self.log, compound=LEFT)

self.small\_log1 = self.log.subsample(1, 1)

self.mike1.config(image=self.small\_log1)

def takeCommand(self):

import speech\_recognition as sr

r = sr.Recognizer()

mic = sr.Microphone(device\_index=1)

with mic as source:

r.adjust\_for\_ambient\_noise(source, duration=1)

audio = r.listen(source, timeout=5)

try:

SQ1 = r.recognize\_google(audio, language='English')

except:

messagebox.showerror('Blood bank management system', 'Sorry! Not able to uderstand')

pass

return SQ1

def search(self):

self.datas=self.entry.get()

import mysql.connector

mydb=mysql.connector.connect(host='localhost',user='root',password='Chirag@1302',database='hospital')

cursor=mydb.cursor()

cursor.execute("SELECT \* FROM details WHERE SNO='"+self.datas+"'")

self.val=cursor.fetchone()

if self.val!=None:

self.edcat=Tk()

self.edcat.title("BLOOD BANK MANAGEMENT SYSTEM")

self.edcat.geometry("300x320+590+320")

self.edcat.configure(bg='#fff')

self.edcat.iconbitmap("aa.ico")

self.fc=Frame(self.edcat,bg='#0f624c',width=300,height=30)

self.fc.place(x=0,y=0)

self.lab=Label(self.fc,bg='#0f624c',fg='#fff',text='EDIT BOOKS',font=('arial',10,'bold'))

self.lab.place(x=112,y=5)

self.labid = Label(self.edcat, bg='#fff', fg='black', text='Patient-ID', font=('arial', 10,

'bold'))

self.labid.place(x=10, y=45)

self.labti = Label(self.edcat, bg='#fff', fg='black', text='Patient-Name', font=('arial', 10,

'bold'))

self.labti.place(x=10, y=90)

self.labaut = Label(self.edcat, bg='#fff', fg='black', text='Blood-group', font=('arial', 10,

'bold'))

self.labaut.place(x=10, y=135)

self.labed = Label(self.edcat, bg='#fff', fg='black', text='Donate blood', font=('arial', 10,

'bold'))

self.labed.place(x=10, y=180)

self.labpr = Label(self.edcat, bg='#fff', fg='black', text='Disease', font=('arial', 10,

'bold'))

self.labpr.place(x=10, y=225)

self.en1=Entry(self.edcat,width=25,bd=4,relief='groove',font=('arial',8,'bold'))

self.en1.place(x=100,y=45)

self.en2 = Entry(self.edcat, width=25, bd=4, relief='groove',font=('arial',8,'bold'))

self.en2.place(x=100, y=90)

self.en3 = Entry(self.edcat, width=25, bd=4, relief='groove',font=('arial',8,'bold'))

self.en3.place(x=100, y=135)

self.en4 = Entry(self.edcat, width=25, bd=4, relief='groove',font=('arial',8,'bold'))

self.en4.place(x=100, y=180)

self.en5 = Entry(self.edcat, width=25, bd=4, relief='groove',font=('arial',8,'bold'))

self.en5.place(x=100, y=225)

self.butt = Button(self.edcat, text='Submit', bg='#0f624c', fg='#fff', width=20, height=0,

font=('Arial', 10, 'bold'),command=self.savedit)

self.butt.place(x=67, y=270)

self.en1.insert(0, self.val[0])

self.en2.insert(0, self.val[1])

self.en3.insert(0, self.val[2])

self.en4.insert(0, self.val[3])

self.en5.insert(0, self.val[4])

self.edcat.mainloop()

else:

messagebox.showerror('BLOOD BANK MANAGEMENT SYSTEM','PLEASE! ENTER THE CORRECT Patient-ID')

def savedit(self):

self.id = self.en1.get()

self.ti = self.en2.get()

self.au = self.en3.get()

self.ed = self.en4.get()

self.pi = self.en5.get()

import mysql.connector

mydb=mysql.connector.connect(host='localhost',user='root',password='Chirag@1302',database='hospital')

cursor=mydb.cursor()

cursor.execute("UPDATE details SET SNO='"+self.id+"', patient\_name='"+self.ti+"',blood\_group='"+self.au+"',donate='"+self.ed+"',disease='"+self.pi+"' WHERE SNO='"+self.datas+"'")

mydb.commit()

messagebox.showinfo('BLOOD BANK MANAGEMENT SYSTEM','YOUR DATA IS UPDATED!')

obj=editing()

obj.edbooks()

def email(self):

class retu(maincode):

def \_\_init\_\_(self):

self.frame=Frame(root,bd=0,relief='flat',bg='#a7ecd9',width=900,height=390)

self.frame.place(x=0,y=110)

self.f1 = Frame(self.frame, bg='#fff', width=500, height=370, bd=5, relief='flat')

self.f1.place(x=200, y=15)

self.ed = Frame(self.f1, bg='#0f624c', bd=0, relief='flat', width=490, height=35)

self.ed.place(x=0, y=0)

self.lac = Label(self.ed, text='SEND EMAIL TO DOCTOR ', bg='#0f624c', fg='#fff', font=('Arial', 12, 'bold'))

self.lac.place(x=175, y=5)

self.label8 = Label(self.f1, text='DOCTOR-ID:', bg='#fff', fg='black', font=('arial', 10, 'bold'))

self.label8.place(x=100, y=65)

self.entry4 = Entry(self.f1, width=30,bd=4, relief='groove', font=('arial', 8, 'bold'))

self.entry4.place(x=188, y=65)

self.label9 = Label(self.f1, text=' EMAIL-ID:', bg='#fff', fg='black', font=('arial', 10, 'bold'))

self.label9.place(x=100, y=105)

self.entry5 = Entry(self.f1, width=30,bd=4, relief='groove', font=('arial', 8, 'bold'))

self.entry5.place(x=188, y=105)

self.label10 = Label(self.f1, text='SUBJECT:', bg='#fff', fg='black', font=('arial', 10, 'bold'))

self.label10.place(x=100, y=145)

self.entry6 = Entry(self.f1, width=30,bd=4, relief='groove', font=('arial', 8, 'bold'))

self.entry6.place(x=188, y=145)

self.label11 = Label(self.f1, text='MESSAGE:', bg='#fff', fg='black', font=('arial', 10, 'bold'))

self.label11.place(x=100, y=185)

self.entry7 = Text(self.f1, width=30,height=5,bd=4, relief='groove', font=('arial', 8, 'bold'))

self.entry7.place(x=188, y=185)

self.button9 = Button(self.f1, text='Send Mail', bg='#0f624c', fg='#fff', width=24, height=0,

font=('Arial', 10, 'bold'),command=self.mail)

self.button9.place(x=140, y=290)

self.backbt = Button(self.frame, width=60, bg='#a7ecd9', activebackground='#a7ecd9',

bd=0, relief='flat', command=self.cur)

self.backbt.place(x=0, y=0)

self.log = PhotoImage(file='back.png')

self.backbt.config(image=self.log, compound=LEFT)

self.small\_log = self.log.subsample(1, 1)

self.backbt.config(image=self.small\_log)

self.mike1 = Button(self.f1, width=25, bg='white',activebackground='white', bd=0, relief='flat',

command=lambda:self.entry4.insert(END, self.takeCommand()))

self.mike1.place(x=380, y=65)

self.log = PhotoImage(file='mike.png')

self.mike1.config(image=self.log, compound=LEFT)

self.small\_log1 = self.log.subsample(1, 1)

self.mike1.config(image=self.small\_log1)

self.mike2 = Button(self.f1, width=25, bg='white',activebackground='white', bd=0, relief='flat',

command=lambda:self.entry5.insert(END, self.takeCommand()))

self.mike2.place(x=380, y=105)

self.log = PhotoImage(file='mike.png')

self.mike2.config(image=self.log, compound=LEFT)

self.small\_log2 = self.log.subsample(1, 1)

self.mike2.config(image=self.small\_log1)

self.mike3 = Button(self.f1, width=25, bg='white',activebackground='white', bd=0, relief='flat',

command=lambda:self.entry6.insert(END, self.takeCommand()))

self.mike3.place(x=380, y=145)

self.log = PhotoImage(file='mike.png')

self.mike3.config(image=self.log, compound=LEFT)

self.small\_log3 = self.log.subsample(1, 1)

self.mike3.config(image=self.small\_log1)

self.mike4 = Button(self.f1, width=25, bg='white',activebackground='white', bd=0, relief='flat',

command=lambda:self.entry7.insert(END, self.takeCommand()))

self.mike4.place(x=380, y=185)

self.log = PhotoImage(file='mike.png')

self.mike4.config(image=self.log, compound=LEFT)

self.small\_log4 = self.log.subsample(1, 1)

self.mike4.config(image=self.small\_log4)

def takeCommand(self):

import speech\_recognition as sr

r = sr.Recognizer()

mic = sr.Microphone(device\_index=1)

with mic as source:

r.adjust\_for\_ambient\_noise(source, duration=1)

audio = r.listen(source, timeout=5)

try:

SQ1 = r.recognize\_google(audio, language='English')

except:

messagebox.showerror('Blood bank management system', 'Sorry! Not able to uderstand')

pass

return SQ1

def mail(self):

self.reciever=self.entry5.get()

self.message=self.entry6.get()

sender = "singhshrey215@gmail.com"

password="conspiracy"

try:

server = smtplib.SMTP\_SSL("smtp.gmail.com", 587)

server.login(sender, password)

server.sendmail(sender, self.reciever,self.message )

print("ok")

messagebox.showinfo("Blood bank management system","Send mail Successfully !")

except:

messagebox.showinfo("Blood bank management system","Send mail Successfully !")

object=retu()

def delete(self):

class dele(maincode):

def deleteee(self):

self.ff = Frame(root, bg='#a7ecd9', width=900, height=390)

self.ff.place(x=0, y=110)

self.f1 = Frame(self.ff, bg='#fff', width=500, height=200, bd=5, relief='flat')

self.f1.place(x=200, y=15)

self.ed = Frame(self.f1, bg='#0f624c', bd=0, relief='flat', width=490, height=35)

self.ed.place(x=0, y=0)

self.lac = Label(self.ed, text='DELETE PATIENT DETAILS ', bg='#0f624c', fg='#fff', font=('Arial', 12,'bold'))

self.lac.place(x=175, y=5)

self.label8 = Label(self.f1, text='Patient\_ID', bg='#fff', fg='black', font=('arial', 10, 'bold'))

self.label8.place(x=85, y=65)

self.entry14 = Entry(self.f1, width=30, bd=4, relief='groove', font=('arial', 8, 'bold'))

self.entry14.place(x=188, y=65)

self.button9 = Button(self.f1, text='Delete', bg='#0f624c', fg='#fff', width=24, height=0,

font=('Arial', 10, 'bold'),command=self.deldata)

self.button9.place(x=140, y=120)

self.backbt = Button(self.ff,width=60, bg='#a7ecd9',activebackground='#a7ecd9',

bd=0, relief='flat', command=self.cur)

self.backbt.place(x=0, y=0)

self.log = PhotoImage(file='back.png')

self.backbt.config(image=self.log, compound=LEFT)

self.small\_log = self.log.subsample(1, 1)

self.backbt.config(image=self.small\_log)

self.mike4 = Button(self.f1, width=25, bg='white',activebackground='white', bd=0, relief='flat',

command=lambda:self.entry14.insert(END, self.takeCommand()))

self.mike4.place(x=380, y=65)

self.log = PhotoImage(file='mike.png')

self.mike4.config(image=self.log, compound=LEFT)

self.small\_log4 = self.log.subsample(1, 1)

self.mike4.config(image=self.small\_log4)

def takeCommand(self):

import speech\_recognition as sr

r = sr.Recognizer()

mic = sr.Microphone(device\_index=1)

with mic as source:

r.adjust\_for\_ambient\_noise(source, duration=1)

audio = r.listen(source, timeout=5)

try:

SQ1 = r.recognize\_google(audio, language='English')

except:

messagebox.showerror('Blood bank management system', 'Sorry! Not able to uderstand')

pass

return SQ1

def deldata(self):

self.a=self.entry14.get()

import mysql.connector

mydb=mysql.connector.connect(host='localhost',user='root',password='Chirag@1302',database='hospital')

cursor=mydb.cursor()

cursor.execute("select \* from details WHERE SNO='"+self.a+"'")

self.da=cursor.fetchone()

if self.da!=None:

cursor.execute("DELETE FROM details WHERE SNO='"+self.a+"'")

mydb.commit()

messagebox.showinfo('Library System','YOUR DATA IS DELETED !')

else:

messagebox.showerror('Library System','YOUR DATA IS NOT FOUND !')

occ=dele()

occ.deleteee()

def search(self):

class demt(maincode):

def delmdata(self):

self.fc = Frame(root, bg='#a7ecd9', width=900, height=390)

self.fc.place(x=0, y=110)

self.fc1 = Frame(self.fc, bg='#fff', width=500, height=200, bd=5, relief='flat')

self.fc1.place(x=200, y=15)

self.edm = Frame(self.fc1, bg='#0f624c', bd=0, relief='flat', width=490, height=35)

self.edm.place(x=0, y=0)

self.lac = Label(self.edm, text='SEARCH DOCTOR ', bg='#0f624c', fg='#fff', font=('Arial', 12, 'bold'))

self.lac.place(x=175, y=5)

self.label8 = Label(self.fc1, text='Doctor Name', bg='#fff', fg='black', font=('arial', 10, 'bold'))

self.label8.place(x=85, y=65)

self.entryl= Entry(self.fc1, width=30, bd=4, relief='groove', font=('arial', 8, 'bold'))

self.entryl.place(x=188, y=65)

self.butto = Button(self.fc1, text='Search', bg='#0f624c', fg='#fff', width=24, height=0,

font=('Arial', 10, 'bold'),command=self.srch)

self.butto.place(x=140, y=120)

self.backbt = Button(self.fc,width=60, bg='#a7ecd9',activebackground='#a7ecd9',bd=0, relief='flat', command=self.cur)

self.backbt.place(x=0, y=0)

self.log = PhotoImage(file='back.png')

self.backbt.config(image=self.log, compound=LEFT)

self.small\_log = self.log.subsample(1, 1)

self.backbt.config(image=self.small\_log)

self.mike4 = Button(self.fc1, width=25, bg='white',activebackground='white', bd=0, relief='flat',

command=lambda:self.entryl.insert(END, self.takeCommand()))

self.mike4.place(x=380, y=65)

self.log = PhotoImage(file='mike.png')

self.mike4.config(image=self.log, compound=LEFT)

self.small\_log4 = self.log.subsample(1, 1)

self.mike4.config(image=self.small\_log4)

def takeCommand(self):

import speech\_recognition as sr

r = sr.Recognizer()

mic = sr.Microphone(device\_index=1)

with mic as source:

r.adjust\_for\_ambient\_noise(source, duration=1)

audio = r.listen(source, timeout=5)

try:

SQ1 = r.recognize\_google(audio, language='English')

except:

messagebox.showerror('Blood bank management system', 'Sorry! Not able to uderstand')

pass

return SQ1

def srch(self):

self.emp=self.entryl.get()

import mysql.connector

mydb=mysql.connector.connect(host='localhost',user='root',password='Chirag@1302',database='hospital')

cursor=mydb.cursor()

cursor.execute("SELECT \* from doctor WHERE D\_name='"+self.emp+"'")

self.srval=cursor.fetchone()

if self.srval!=None:

self.top=Tk()

self.top.title("BLOOD BANK MANGEMENT SYSTEM")

self.top.iconbitmap("aa.ico")

self.top.geometry("600x600+600+300")

self.top.resizable(0, 0)

self.top.configure(bg='black')

self.frm=Frame(self.top,bg='#0f624c',width=300,height=35)

self.frm.place(x=0,y=0)

self.mnlb=Label(self.frm,bg='#0f624c',fg='#fff',text="Avaliable",font=('arial',11,'bold'))

self.mnlb.place(x=120,y=5)

self.lb1 = Label(self.top, text='Doctor-ID', bg='black', fg='blue', font=('arial', 12, 'bold'))

self.lb1.place(x=40,y=80)

self.lb2=Label(self.top,text=self.srval[0],bg='black',fg='blue',font=('arial',12,'bold'))

self.lb2.place(x=150,y=80)

self.lb3 = Label(self.top, text='Doctor-Name', bg='black', fg='blue', font=('arial', 12, 'bold'))

self.lb3.place(x=40, y=160)

self.lb4 = Label(self.top, text=self.srval[1], bg='black', fg='blue', font=('arial', 12, 'bold'))

self.lb4.place(x=150, y=160)

self.lb5 = Label(self.top, text='Doctor-Cabin', bg='black', fg='blue', font=('arial', 12, 'bold'))

self.lb5.place(x=40, y=240)

self.lb6 = Label(self.top, text=self.srval[2], bg='black', fg='blue', font=('arial', 12, 'bold'))

self.lb6.place(x=150, y=240)

self.lb7 = Label(self.top, text='Department', bg='black', fg='blue', font=('arial', 12, 'bold'))

self.lb7.place(x=40, y=320)

self.lb8 = Label(self.top, text=self.srval[3], bg='black', fg='blue', font=('arial', 12, 'bold'))

self.lb8.place(x=150, y=320)

self.lb9 = Label(self.top, text='Contact', bg='black', fg='blue', font=('arial', 12, 'bold'))

self.lb9.place(x=40, y=400)

self.lb10 = Label(self.top, text=self.srval[4], bg='black', fg='blue', font=('arial', 12, 'bold'))

self.lb10.place(x=150, y=400)

self.lb11 = Label(self.top, text='qualification', bg='black', fg='blue', font=('arial', 12, 'bold'))

self.lb11.place(x=40, y=480)

self.lb12 = Label(self.top, text=self.srval[5], bg='black', fg='blue', font=('arial', 12, 'bold'))

self.lb12.place(x=150, y=480)

else:

messagebox.showwarning('Library System','YOUR DATA IS NOT AVAILABLE !')

object=demt()

object.delmdata()

def show(self):

class tst(maincode):

def \_\_init\_\_(self):

self.fc = Frame(root, bg='#a7ecd9', width=900, height=390)

self.fc.place(x=0, y=110)

self.popframe=Frame(self.fc,width=900,height=30,bg='#0f624c')

self.popframe.place(x=0,y=0)

self.lbn=Label(self.popframe,bg='#0f624c',text='Staff Information',fg='#fff',font=('arial',10,

'bold'))

self.lbn.place(x=380,y=5)

self.backbt = Button(self.popframe,width=30, bg='#0f624c',activebackground='#0f624c',

bd=0, relief='flat', command=self.cur)

self.backbt.place(x=0, y=0)

self.log = PhotoImage(file='back.png')

self.backbt.config(image=self.log, compound=LEFT)

self.small\_log = self.log.subsample(2, 2)

self.backbt.config(image=self.small\_log)

self.table\_frame=Frame(self.fc,bg='#fff',bd=1,relief='flat')

self.table\_frame.place(x=0,y=30,width=900,height=360)

self.scroll\_x=Scrollbar(self.table\_frame,orient=HORIZONTAL)

self.scroll\_y=Scrollbar(self.table\_frame,orient=VERTICAL)

self.book\_table=ttk.Treeview(self.table\_frame,columns=("D\_ID","D\_name","D\_cabin","dept",

"contact","qualification"),

xscrollcommand=self.scroll\_x.set,yscrollcommand=self.scroll\_y.set)

self.scroll\_x.pack(side=BOTTOM,fill=X)

self.scroll\_y.pack(side=RIGHT, fill=Y)

self.scroll\_x.config(command=self.book\_table.xview)

self.scroll\_y.config(command=self.book\_table.yview)

self.book\_table.heading("D\_ID",text="D\_ID")

self.book\_table.heading("D\_name", text="D\_name")

self.book\_table.heading("D\_cabin", text="D\_cabin")

self.book\_table.heading("dept", text="dept")

self.book\_table.heading("contact", text="contact")

self.book\_table.heading("qualification", text="qualification")

self.book\_table['show']='headings'

self.book\_table.column("D\_ID",width=10)

self.book\_table.column("D\_name", width=10)

self.book\_table.column("D\_cabin", width=10)

self.book\_table.column("dept", width=10)

self.book\_table.column("contact", width=10)

self.book\_table.column("qualification", width=10)

self.book\_table.pack(fill=BOTH,expand=1)

self.fetch\_data()

def fetch\_data(self):

import mysql.connector

mydb=mysql.connector.connect(host='localhost',user='root',password='Chirag@1302',database='hospital')

cursor=mydb.cursor()

cursor.execute("SELECT \* FROM doctor")

self.rows=cursor.fetchall()

if len(self.rows)!=0:

for self.row in self.rows:

self.book\_table.insert('',END,values=self.row)

mydb.commit()

oc=tst()

def code(self):

self.fm=Frame(root,height=500,width=900,bg='white')

self.fm.place(x=0,y=0)

self.canvas=Canvas(self.fm,height=500,width=900,bg='#22224b')

self.canvas.place(x=0,y=0)

self.photo=PhotoImage(file="images (17).png")

self.canvas.create\_image(-148,-337,image=self.photo,anchor=NW)

self.fm1=Frame(self.canvas,height=260,width=300,bg='white',bd=3,relief='ridge')

self.fm1.place(x=300,y=200)

self.b1=Label(self.fm1,text='Patient ID',bg='white',font=('Arial',10,'bold'))

self.b1.place(x=20,y=42)

self.e1=Entry(self.fm1,width=22,font=('arial',9,'bold'),bd=4,relief='groove')

self.e1.place(x=100,y=40)

self.lb2=Label(self.fm1,text='Password',bg='white',font=('Arial',10,'bold'))

self.lb2.place(x=20,y=102)

self.e2=Entry(self.fm1,width=22,show='\*',font=('arial',9,'bold'),bd=4,relief='groove')

self.e2.place(x=100,y=100)

self.btn1=Button(self.fm1,text=' login',fg='white',bg='red',width=100,font=('Arial',11,'bold'),

activebackground='white',activeforeground='black',command=self.login,bd=3,relief='flat',cursor='hand2')

self.btn1.place(x=25,y=160)

self.logo = PhotoImage(file='user.png')

self.btn1.config(image=self.logo, compound=LEFT)

self.small\_logo = self.logo.subsample(1, 1)

self.btn1.config(image=self.small\_logo)

self.btn2=Button(self.fm1,text='Clear',fg='white',bg='blue',width=100,font=('Arial',11,'bold'),

activebackground='white',activeforeground='black',bd=3,relief='flat',cursor='hand2',

command=self.mainclear)

self.btn2.place(x=155,y=160)

self.log = PhotoImage(file='cart.png')

self.btn2.config(image=self.log, compound=LEFT)

self.small\_log = self.log.subsample(1, 1)

self.btn2.config(image=self.small\_log)

self.forgot=Label(self.fm1,text='Sign up',fg='red',bg='#fff',bd=3,activeforeground='black',

font=('cursive',9,'bold'))

self.forgot.place(x=120,y=220)

self.forgot.bind("<Button>",self.mouseClick)

self.mike70 = Button(self.fm1, width=25, bg='white',activebackground='white', bd=0, relief='flat',

command=lambda:self.e1.insert(END, self.takeCommand1()))

self.mike70.place(x=255, y=38)

self.log = PhotoImage(file='mike.png')

self.mike70.config(image=self.log, compound=LEFT)

self.small\_log70 = self.log.subsample(1, 1)

self.mike70.config(image=self.small\_log70)

self.mike25 = Button(self.fm1, width=25, bg='white',activebackground='white', bd=0, relief='flat',

command=lambda:self.e2.insert(END,self.takeCommand1()))

self.mike25.place(x=255, y=98)

self.log = PhotoImage(file='mike.png')

self.mike25.config(image=self.log, compound=LEFT)

self.small\_log25 = self.log.subsample(1, 1)

self.mike25.config(image=self.small\_log25)

root.mainloop()

def mouseClick(self,event):

self.rog=Tk()

self.rog.title("Sign up")

self.rog.geometry("400x300+530+280")

self.rog.iconbitmap("aa.ico")

self.rog.resizable(0,0)

self.rog.configure(bg='#fff')

self.label=Label(self.rog,text="Add your deatils",bg='#fff',fg='red',font=("cursive",20,'bold'))

self.label.place(x=105,y=15)

self.user=Label(self.rog,text='Patient ID :',bg='#fff',fg='black',font=("cursive",10,'bold'))

self.user.place(x=40,y=95)

self.user=Label(self.rog,text='Patient Name :',bg='#fff',fg='black',font=("cursive",10,'bold'))

self.user.place(x=40,y=135)

self.user = Label(self.rog, text='password :', bg='#fff', fg='black', font=("cursive", 10, 'bold'))

self.user.place(x=40, y=180)

self.e1 = Entry(self.rog, width=24, font=('arial', 9, 'bold'), bd=4, relief='groove')

self.e1.place(x=170, y=95)

self.e2 = Entry(self.rog, width=24, font=('arial', 9, 'bold'), bd=4, relief='groove')

self.e2.place(x=170, y=135)

self.e3 = Entry(self.rog, width=24, font=('arial', 9, 'bold'), bd=4, relief='groove')

self.e3.place(x=170, y=175)

self.btn1 = Button(self.rog, text='Submit', fg='white', bg='#5500ff', width=20, font=('Arial', 13, 'bold'),

activebackground='white', activeforeground='black',bd=3, relief='flat',

cursor='hand2',command=self.sign\_up)

self.btn1.place(x=100, y=240)

def sign\_up(self):

self.a=self.e1.get()

self.b=self.e2.get()

self.c=self.e3.get()

mydb=mysql.connector.connect(host='localhost',user='root',password='Chirag@1302',database='hospital')

cursor=mydb.cursor()

cursor.execute("SELECT Patient\_ID FROM patient WHERE Patient\_ID='"+self.a+"'" )

self.data=cursor.fetchone()

if self.data==None:

s="INSERT INTO patient (Patient\_ID,Patient\_name,Password) VALUES(%s,%s,%s)"

b1=(self.a,self.b,self.c)

cursor.execute(s,b1)

mydb.commit()

messagebox.showinfo("Blood management system","You are successfully registered !")

else:

messagebox.showinfo("Blood management system","You are Already registered !")

self.rog.mainloop()

wishMe()

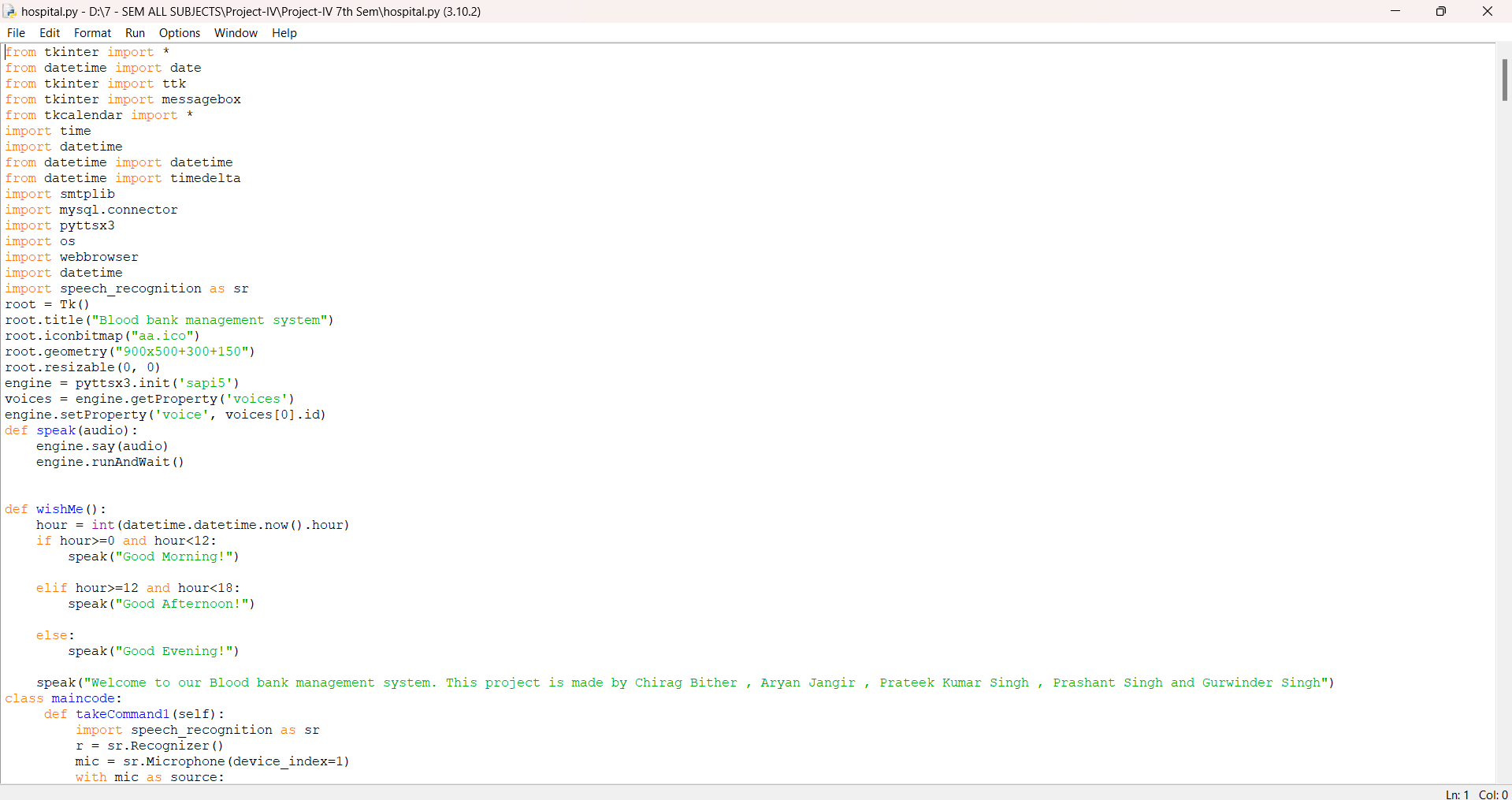
ob=maincode()

ob.code()

## USER MANUAL

**Step 1: Access the Application**

Open the E-Blood Bank Monitoring Hospitals application by running hospital.py file on your device using IDLE (Python) or similar application.



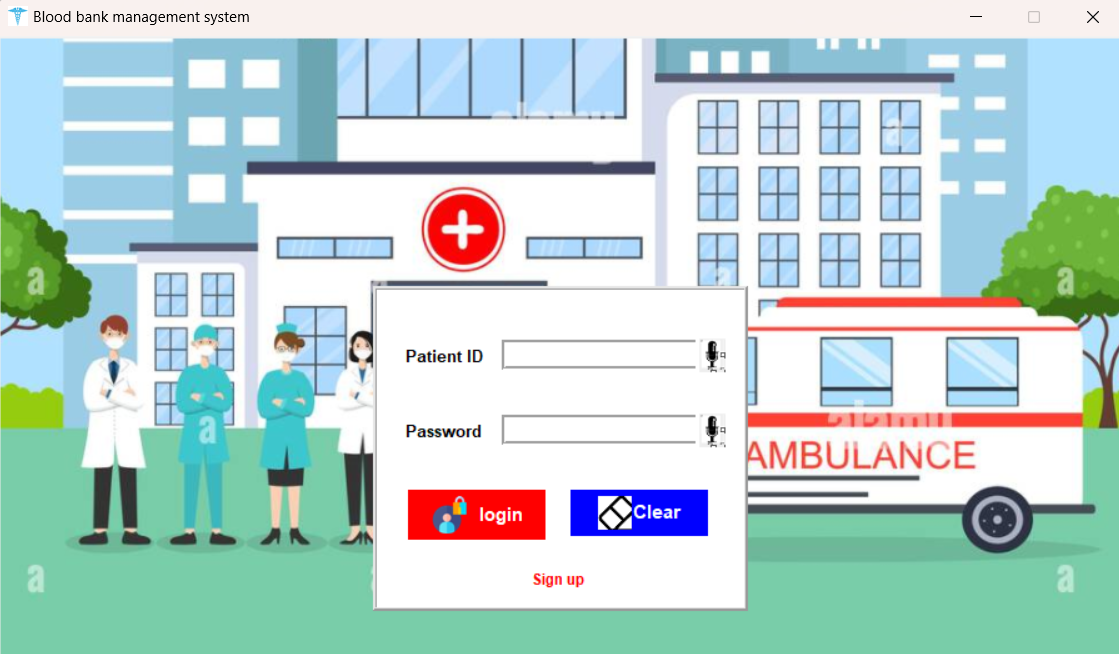
**Step 2: User Registration**

If you are a new user, navigate to the sign-up page. Provide the required details such as Patient ID, Patient Name, and a secure password. Click on the "Submit" button to create your account.



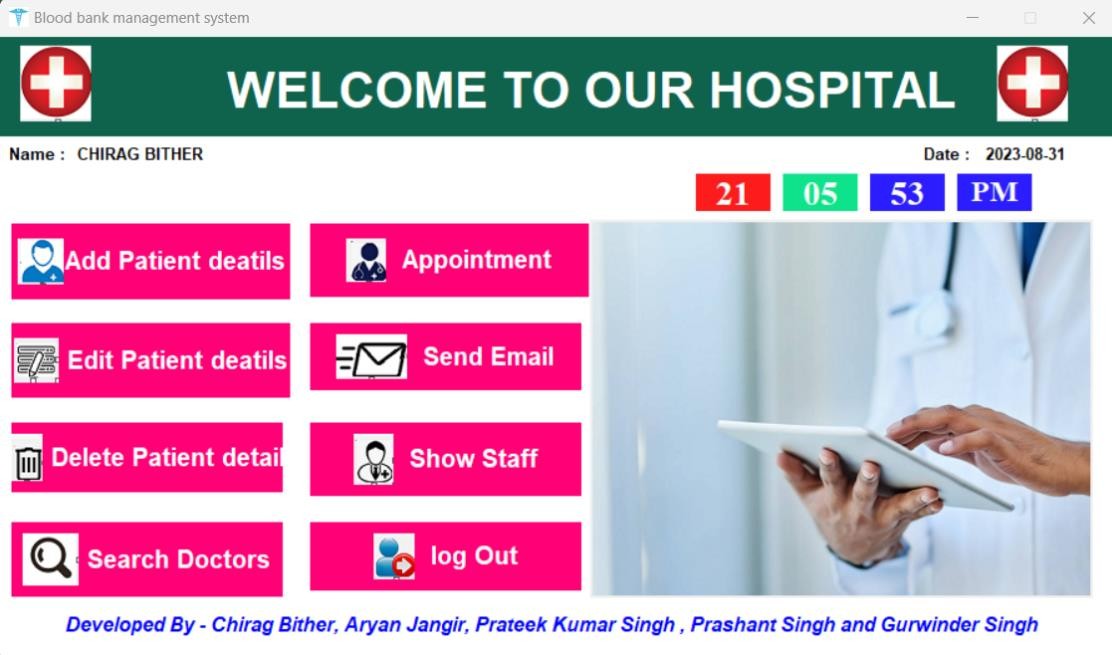
**Step 3: User Login**

Use your registered credentials to log in to the application. Enter your Patient ID and password on the login page and click "Login."



**Step 4: Dashboard Navigation**

Once logged in, you will be directed to the application dashboard. The dashboard serves as the central hub for various functionalities. Familiarize yourself with the available options.



**Step 5: Application Features**

* Add Patient Details:

Add patient information through the "Add Patient Details" feature.

* Book Appointment:

Access the "Book Appointment" section to schedule appointments with healthcare professionals.

* Edit Patient Details:

Modify and update patient information through the "Edit Patient Details" feature.

* Send Email:

Communicate with healthcare providers by using the "Send Email" functionality.

* Show Staff:

View details of hospital staff through the "Show Staff" option.

* Search Doctors:

Find specific doctors by utilizing the "Search Doctors" feature.

* Log Out:

Safely log out from your account using the "Log Out" button.

**Step 6: Additional Functionalities**

Discover additional features such as "Delete Patient Details" and any other functionalities designed to enhance user experience.

**Step 7: Data Security and Privacy**

* Your data is securely stored in the application's database.
* Avoid sharing login credentials with others.