ANJUMAN INSTITUTE OF MANAGEMENT AND COMPUTER APPLICATION

BHATKAL 581320

(Affiliated to Karnatak University, Dharwad)

2023– 2024

DEPARTMENT OF BACHELOR OF COMPUTER APPLICATION (BCA)

PROJECT REPORT ON

" STREAMLINING BLOOD BANK MANAGEMENT "

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT OF THE BCA PROJECT WORK AS SPECIFIED BY THE KARNATAK UNIVERSITY, DHARWAD BCA VI SEMISTER DEGREE.

SUBMITTED BY:

SYED MIRANJI MIYA

(U02JV21S0033)

GUIDED BY:

Prof. Prakash Naik

(HOD & Lecturer of AIMCA Bhatkal)

ANJUMAN INSTITUTE OF MANAGEMENT AND COMPUTER APPLICATION
ANJUMANABAD BHATKAL (U.K) - 581 320
(Affiliated to Karnatak University, Dharwad)

Department of Bachelor of Computer Application (BCA)

CERTIFICATE

This is certified that the project work entitled "Streamlining Blood Bank Management" is a Bonafied work carried out by Syed Miranji Miya Reg No:U02JV21S0033 in partial fulfillment for the award of Bachelor of Computer Application from the AIMCA, Bhatkal in the year 2023-24. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said Degree.

Project Guide **Principal**

Examiners:

1. ……
2. ……

DECLARATION

I hereby declare that the project report title **"Streamlining Blood Bank Management"** has been prepared by me during the year 2023-2024 under the valuable guidance and supervision of Prof. Prakash Naik Lecturer of Anjuman Institute of Management and Computer Application, Bhatkal for the fulfilment of the requirement of Bachelor's Degree in Computer Application.

I also declare that project result is of my own effort and has not been submitted to any other University for the award of any degree.

-SYED MIRANJI MIYA

ACKNOWLEDGEMENT

In this segment I would like to thank all the members those who have been a part of my project in one way or the other.

First and foremost, I would like to thank our project guide Prof. D. N. Noorappanavar (Hasan)Prof. Prakash Naik , and Prof. Deepa Bhatkal. Department of Computer Science, AIMCA Bhatkal for his/her moral support and frequently suggestions in developing this project.

My humble thanks to the review committee Prof. Prakash Naik , Prof. D. N. Noorappanavar (Hasan) for instructing us to modify the things and helping me to get things done in time.

I am very grateful to avail this opportunity to thank our **Principal Mohammad Mohsin K** Anjuman Institute of Management and Computer Application Bhatkal, for all the facilities provide to me and supporting me all academic endeavours.

Finally, I take this opportunity to express our gratitude and respect to all those who directly or indirectly helped and encouraged me during the course of the project.

- SYED MIRANJI MIYA

ABSTRACT

This project has been developed using Visual Studio Code (VScode). **HTML, CSS** and **JAVA Script** as the front end and **NodeJS** and **Mongodb** as the backend i.e. database to store the records. The project entitled "**Streamlining Blood Bank Management"**.

STREAMLINING BLOOD BANK MANAGEMENT - The efficient management of blood banks is crucial for ensuring timely access to safe blood transfusions, which are vital in medical emergencies and routine medical procedures alike. This project focuses on the development of a comprehensive Blood Bank Management System (BBMS) to streamline the operations of blood banks, enhance donor-recruitment strategies, and facilitate the efficient distribution of blood to healthcare facilities.

- Building or renovating medical facilities. This might involve architectural planning, equipment procurement, and ensuring compliance with safety standards.

- Implementing or upgrading electronic health record (EHR) systems, other digital tools. HIS projects aim to enhance data accuracy, accessibility, and interoperability.

CONTEXT

I

INTRODUCTION

INTRODUCTION

STREAMLINING BLOOD BANK MANAGEMENT

I developed **"BLOOD BANK MANAGEMENT"** which provides the user and admin to ease in adding and editing the value accordingly, with regards date wise, month wise, year wise. In this project we use PHP and MySQL database.

The modules in BLOOD BANK MANAGEMENT SYSTEM

User Module-

In This **Online Blood Bank Management System project there** has two sides of the user interface which are the **Management Panel/Admin Panel** and the **Public/Customer Site**.

II

SYSTEM STUDY

SYSTEM STUDY

EXISTING SYSTEM WITH LIMITATION.

Existing System with Limitations

The existing blood bank management systems are typically manual or semi-automated, leading to various limitations:

*Data Inaccuracy: Manual entry errors.

*Limited Accessibility: Difficulty in accessing real-time data.

*Inefficiency: Slow processing and response times.

* Poor Inventory Management: Challenges in tracking and managing blood stock levels.

PROPOSED SYSTEM WITH OBJECTIVES.

The proposed system aims to address these limitations by implementing a fully automated, centralized blood bank management system. Objectives include:

*Improving Data Accuracy: Automated data entry and validation.

*Enhancing Accessibility: Real-time data access through web and mobile applications.

*Increasing Efficiency: Streamlined processes for donor registration, blood collection, and inventory management.

*Optimizing Inventory Management: Management of blood stock levels.

FEASIBILITY STUDY.

Feasibility is carried out to assess the risk involved in the development of the project. The impact of the new system on organization needs to be assessed. It is also to be seen whether the system is capable of meeting the user needs and whether it will lead to effective use of the available resources.

The objective of feasibility study is not to solve the problem but to understand the scope of the software. It is also used to measure the requirements that needs to be taken up for development in the current version of the system. The analysis is carried out at the beginning as well as throughout the life cycle of the project.

a. Technical Feasibility.
b. Economic Feasibility.
c. Operational Feasibility.

a)TECHNICAL FEASIBILITY

Technical Feasibility is concerned with the availability and capability of hardware, Software and the people.

1) Hardware:

A computer should be available that is powerful enough to handle the proposed system

2) Software:

According to the convince of the web, the web developer should decide which computer language or software going to use for the proposed project. The developer should keep in mind that the software language is going to use should be easily available, efficient, portable.

3) People:

The developers of the system should be ready to learn new set of skills if necessary. It will be beneficial if he/ she is through with the operating system and system and SDLC.

b)ECONOMICAL FEASIBILITY

The cost is low when compare to the manual method. The design and the implementation of the project **blood bank management** is to save the time and is user friendly.

c)OPERATIONAL FEASIBILITY

A system can be technically feasible and still not be implementable. The organization are so having the will to implement it. Management must actively support
development of the system.

III

SYSTEM ANALYSIS

SYSTEM ANALYSIS

ENTITY-RELATIONSHIP MODEL

An entity relationship diagram, also called entity relationship model, is a graphical representation of entities and their relationships to each other, typically used in computing in regard to the organization of data within databases or information systems.

An entity relationship diagram is a specialized graphic that illustrates the relationships between entities in a database. Also known as E-R Diagram.

E-R Diagram represent the schemas rather than the instances. This is more useful because a database schema changes rarely, whereas the extension changes frequently. In addition, the schema is usually easier to display than the extension of a database.

An ER diagram is a pictorial representation of the information that can be captured by a database. Such a "picture" serves two purposes.

- It allows database professionals to describe an overall design concisely yet accurately.
- (Most of) it can be easily transformed into the relational schema.

Elements of E-R diagram

ENTITY:

The basic object that the ER diagram represents is an entity, which is a thing in the real world with an independent existence. An entity may be an object with a physical existence or it may be an object with a conceptual existence. Each, entity have attributes.

Ex: Smith, Bob, CS, Engineering,354,459…

Entity Set: a set of entities of the same type that share the same properties.

Ex: the set of students, the set of departments, the set of course.

WEAK ENTITY:

A weak entity is an entity that depends on the existences of another entity. In more technical terms it can defined as an entity that cannot be identified by its own attributes. It uses a foreign combined with its attributed to from the primary key.

ATTRIBUTE:

An entity is represented by a set of attributes. Attributes are descriptive properties possessed by all members of an entity set.

Attribute Type

- **Simple Attribute**: cannot be divided into subparts.
- **Composite Attribute**: which can be divided into subparts.
- **Single-valued Attribute**: each entity has only one value.
- **Multi-valued Attribute**: an entity may have zero, one, or more values.
- **Derived Attribute**: can be computed from other attributes.

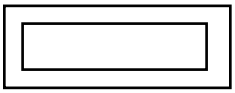
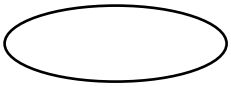
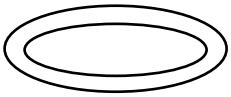
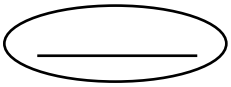
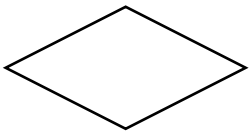
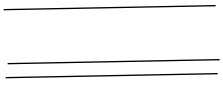
RELATIONSHIPS

A relationship describes how entities interact. Simply, an association among several entities. A set of relationships may have common features.

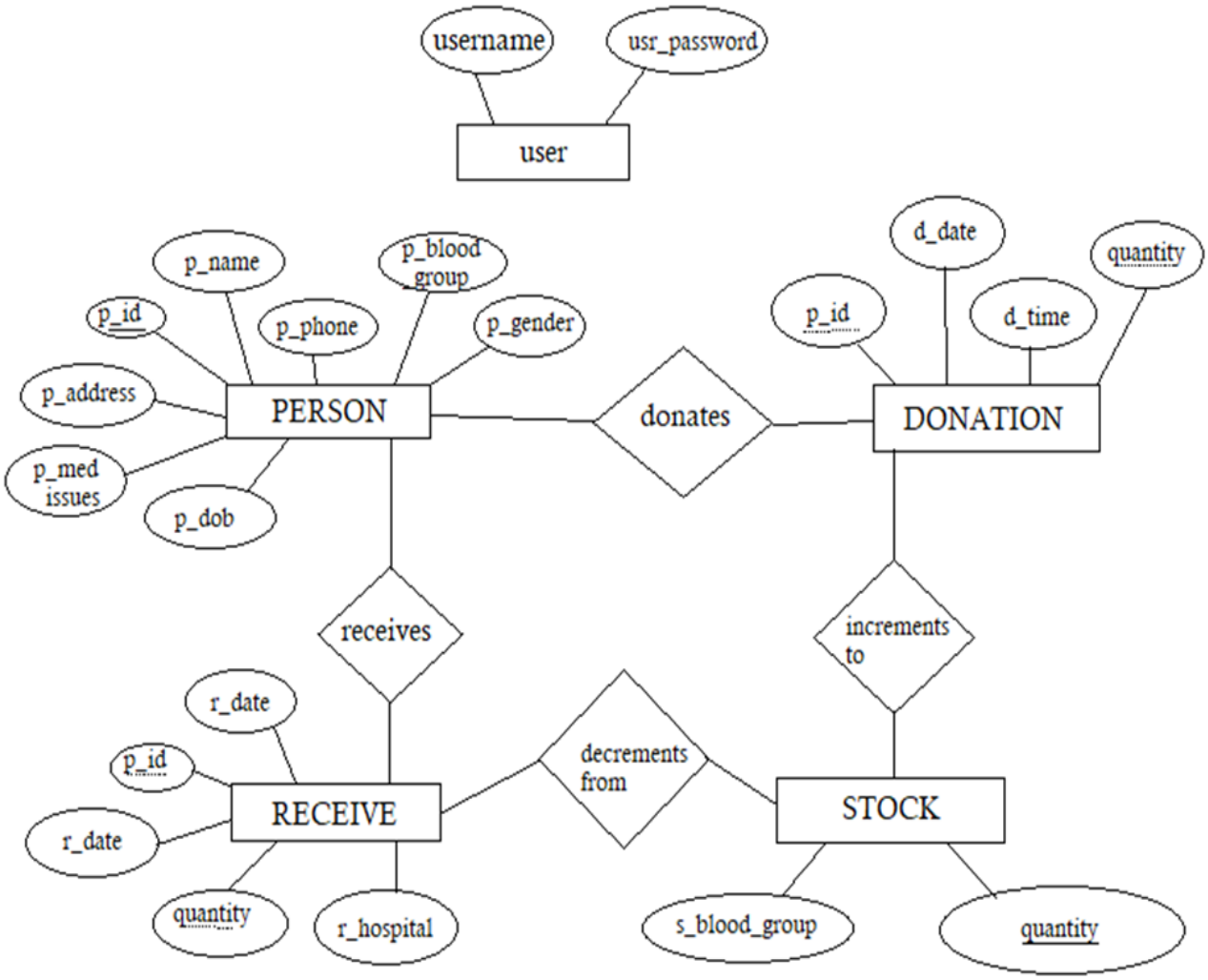
*Donors to Blood Donations: One-to-Many relationship. A donor can make multiple blood donations.

.

E-R DIAGRAM NOTATIONS WITH MEANING

SI. No	Entity set/ Attributes	ER-Notations	Meaning in Life
1	Strong Entity Set		Rectangular Box (Single Line)
2	Weak Entity Set		Rectangular Box (Double Line)
3	Attribute		Oval Box (Single Line)
4	Multi-valued Attribute		Oval Box (Double Line)
5	Derived Attribute		Oval Box (dotted Line)
6	Primary Key		Underlined
7	Relationship Set		Diamond Box (Single Line)
8	Participation Constraints		Single Line (Total) Double Line (Partial)

ER-diagram :


username
usr_password
user
p_name
p_blood group
p_id
p_phone
p_gender
p_address
PERSON
p_med issues
p_dob
donates
d_date
quantity
p_id
d_time
DONATION
receives
increments to
r_date
p_id
RECEIVE
r_date
quantity
r_hospital
decrements from
STOCK
s_blood_group
quantity

DATA FLOW DIAGRAM

Data Flow Diagram are a graphical tool used to describe and analyses the movement of data through a system. DFD's are used to capture the essential feature of both existing real system and feature physical implementation of the system.

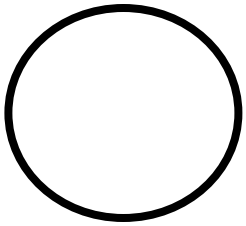
The DFD is a graphical technique that depicts information flow and the transforms that are applies as data move from input to output. The DFD is also known as Bubble chart or Data flow Graphs or Context Diagram.

The DFD is a graphical technique that depicts information flow and the transforms that are applies as data move from input to output. The DFD is also known as Bubble chart or Data flow Graphs or Context Diagram.

Data flow diagrams are constructed from four basic building blocks.

- Processes
- Data Flow
- Store
- Terminator

THE PROCESSES



- Also called bubble, Function and transformation.
- Shows part of the system that transforms input to outputs.
- Represented graphically as a circle.
- Named with single word, phase, or sentences.

DATA FLOW

It is represented by and arrow line and name of the data is specified by the said of the line as label. This is used for data movement.

TWO DATA PACKETS

Represented graphically by an arrow into out of service process.

Describes movements of information in the systems “Data in motion”.

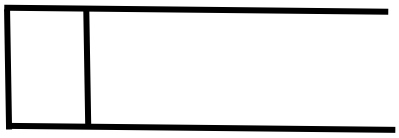
If flow shows direction.

Double – headed arrow stands for dialogue – Convenient packaging of two data packets.

Data flow can diverge or coverage in a DFD.

DATA STORE

It is represented by one open end rectangle. The data base used in the system are specified by this notation.



SOURCES OR LINKS

It is represented by one end rectangle. It is used for specifying form where data comes are where it reaches.

Steps Required in **DFD**

- Identify System, Processing transformations. Transactions concerned reading, validating and formatting inputs.
- Identify input transformations. Transformations concerned with validating and formatting inputs.
- Identify output transformations. Transformations concerned with And writing output. Group under the output.
- Functions.

SYSTEM REQUIREMENTS AND SPECIFICATION:

Hardware requirement and specification:

- RAM: min 512MB.
- Storage: min 2GB.
- Processor: intel core i3 or above.

Software requirement and specification:

- Operating system: windows 2000 or above, Android, IOS and LINUX.
- Tools: XAMPP, web browser.
- Technology: HTML, CSS and PHP (Front end).
- Database: NodeJS and Mongdb(Back end).

LANGUAGES USED:

1. HTML5:

HTML5 is the latest evolution of the standard that defines HTML The term represents two different concepts. It is a new version of the language HTML, with new elements, attributes, and behaviors, and a larger set of technologies that allows the building of more diverse and powerful Web sites and applications. This set is sometimes called HTML5 & friends and often shortened to just HTML5.

Sl no.	Tags	Information
a.	<!-- -->	comment
b.	<!Doctype>	Document type
c.	<a>	Hyperlink
d.	<body>	Body Element
e.	 	Inserts a single line break
f.	<button>	Push button
g.		Selection in a document
h.	<form>	Form
i.	<html>	Html document
j.	<h1> to <6>	Header1 to header6
k.	<img>	Image
l.	<input>	Input field
m.	<label>	Label for a form control

2. CSS3:

CSS stands for **C**ascading **S**tyle **S**heets. CSS describes how HTML elements are to be displayed on screen**,** paper**,** or in other media. CSS saves a lot of work. It can control the layout of multiple web pages all at once. External stylesheets are stored in CSSfiles.

Sl no.	Tags	Information
a.	animation	Specifies the keyframe-based animations.
b.	animation-name	Specifies the name of a@keyframes defined be applied to
c.	Background	Defines a variety of background properties within one declaration.
d.	background-color	Defines an element's background color.
e.	Border	Sets the width, style, and color for all four sides of an element's border.
f.	Color	Specify the color of the text of an element.
g.	Opacity	Specifies the transparency of an element.
h.	Height	Specify the height of an element.
i.	Width	Specify the width of an element.
j.	Content	Inserts generated content.
k.	align-items	Specifies the default alignment for items within the flex container.
l.	Font	Defines a variety of font properties within one declaration.

DBMS

A DBMS makes it possible for end users to create, read, update, and delete data in the database. The DBMS essentially serves as an interface between the database and end users or application programs, ensuring that data is consistently organized and remains easily accessible.

The DBMS manages three important things: the data, the database engine that allows data to be accessed, locked and modified- and the database schema, which defines the databases logical structure. These three foundational elements help provide concurrency, security, data integrity and uniform administration procedures. Typical database administration tasks support by the DBMS include change management, performance monitoring/ tuning and backup and recovery. Many database management systems are also responsible for automated rollbacks, restarts and recovery as well as the logging and auditing of activity.

The DBMS is perhaps most useful for providing a centralized view of data that can be accessed by multiple users, from multiple locations, in a controlled manner, A DBMS can limit what data the end users sees, as well as how that end user can view the data, providing many views of single database schema. End users and software programs are free from having to understand where the data is physically located or on what type of storage media it resides because the DBMS handles all requests.

The DBMS can offer both logical and physical data independence. That means it can protect users and application from needing to know where data is stored or having to be concerned about changes to the physical structure of data (storage and hardware). As long as programs use the application programming interface (API) for the database that is provided by the DBMS, developers won't have to modify programs just because changes have been made to the database.

TOOL USED:

1.VISUAL STUDIO CODE:

Visual Studio Code is a source-code editor developed by MICROSOFT for Windows, Linux and MacOS It includes support for debugging embedded Git control, syntax highlighting, intelligent code completion, snippets, and code refactoring. It is also customizable, so users can change the editor's themes, keyboards shortcuts and preferences. The source code is free and open source and released under the permissive MIT License The compiled binaries are freeware and free for private or commercial use.

Features: Visual Studio Code is a source code editor that can be used with a variety of programming languages. Instead of a project system it allows users to open one or more directories, which can then be saved in workspaces for future reuse. This allows it to operate as a language-agnostic code editor for any language, contrary to Microsoft visual studio which uses the proprietary .sln solution file and project-specific project files. It supports a number of programming languages and a set of features that differs per language. Unwanted files and folders can be excluded from the project tree via the settings. Many of Visual Studio Code features are not exposed through menus or the user interface, but can be accessed via the command palette.

Language supports: Visual Studio Code has out-of-the-box support for almost #, every major programming language. Several are included by default. Example C++, C#, CSS, Dockerfile, Go, HTML, Java, JavaScript, JSON, Less, Markdown, PHP, PowerShell, Python, SCSS, T-SQL, TypeScript.

IV

SYSTEM DESIGN

SYSTEM DESIGN

System design provides the understanding and procedural details necessary for implementing the system recommended in the system study. Emphasis is on the translating the performance requirements into design specifications.

The Design phase is a transition from a user-oriented document (System proposal) to a document oriented to the programmers or database personal.

System Design goes through two phases of development:

- Logical Design
- Physical Design

A data flow diagram shows the logical flow of the system. For a system it describes the input (source), output (destination), database (data stores) and procedures (data flows) all in a format that meets the user's requirement. When analysis prepares the logical system design, they specify the users' needs at a level of detail that virtually determines the information flow into an out of the system and the required data resources. The logical design also specifies input forms and screen layouts.

The activities following logical design are the procedure followed in the physical design e.g., producing programs, software, file and a working system. Design specifications instruct the user about what the system should do. System Design is the first design stage in which the basic approach to solving the problem is selected. During system design, the overall structure and style are decided.

DATABASE DESIGN

Database design is the process of developing database structures. To hold data to cater to user requirements. The final design must satisfy user needs in terms of completeness, integrity, performance and other factors. For a large enterprise, the database design will turn out to be extremely complex task leaving a lot to the skill assisted techniques are available to facilitate database design. The primary input to the database design process is the organization's statement of requirements. Poor definition of these requirements is a major cause of poor database design, resulting in database of limited scope and utilities which are usable to adapt to changes.

The major step in database design is to identify the entities and relationship that reflects the organization's data naturally. The objective of this step is to specify conceptual structure of the data and is referred to as data modelling.

There are several methodologies to model the data logically. We adopted ER Modelling as our data modelling techniques for analysis and logical modelling of systems data requirement. It uses three basic concepts: Entities, Attributes, Relations.

Table design:

user:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	uid	int(8)			No	None		AUTO_INCREMENT	Chang
2	uname	varchar(40)	latin1_swedish_ci		No	None			Chang
3	uemail	varchar(80)	latin1_swedish_ci		No	None			Chang
4	upass	varchar(32)	latin1_swedish_ci		No	None			Chang
5	Creation_date	datetime			No	current_timestamp()			Chang

expense:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(10)			No	None		AUTO_INCREMENT	Change Drop More
2	pname	varchar(2000)	latin1_swedish_ci		No	None			Change Drop More
3	pprice	float			No	None			Change Drop More
4	uid	int(3)			No	None			Change Drop More
5	date	date			No	None			Change Drop More
6	isdel	int(1)			No	None			Change Drop More

income

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(10)			No	None		AUTO_INCREMENT	Change Drop More
2	income	varchar(2000)	latin1_swedish_ci		No	None			Change Drop More
3	tvalue	float			No	None			Change Drop More
4	uid	int(3)			No	None			Change Drop More
5	date	date			No	None			Change Drop More
6	isdel	int(1)			No	None			Change Drop More

V

SYSTEM TESTING

SYSTEM TESTING

TESTING

Once source has been generated, software must be tested to uncover (and correct) as many errors as possible before delivery to your customer. Your goal is to design a series of test cases that have a likelihood of finding errors but how?? That's where software testing techniques enter the picture. These techniques provide systematic guidance for designing test that

(1) Exercise the internal logic of software components, and
(2) Exercise the input and output domains of the program to uncover errors in program function, behaviour, and performance.

During early stages of testing a software engineer all tests. However, as the testing process progresses, testing specialists may become involved.

Importance of testing is that reviews and other SQA activities can and do uncover errors, but they are not sufficient. Every time the program executed, the customers tests it! Therefore, you have to execute the program before it gets to the customers with the specific intent of finding and recovering all errors. In order to find the highest possible number of errors, tests must be conducted systematically and test cases must be designed using disciplined techniques.

Testing Principles

Before applying methods to design effective test cases, a software engineer must understand the basic principles that guide software testing. Davis suggests a set1 of testing principles that :

- All tests should be traceable to design customer requirements. The objective of software testing is to uncover errors. It follows that the most severe (from the customer's point of view) are those that cause the program to fail to meet its requirements.
- Tests should be planned long before testing begins. Test planning can begin as soon as the design model has been solidified. There, all tests can be planned and designed before any code has been generated.
- The Pareto principle applies to software testing. Stated simply, the Pareto principle implies that 80% of all errors uncovered during testing will likely be traceable to 20% of all program components. The problem, of course, is to isolate these suspect components and to thoroughly test them.
- Testing should begin "in the small" and process toward testing "in the large". The first tests planned and executed generally focus on individual components. As testing progresses, focus shifts in an attempt to find errors in integrated clusters of components and ultimately in the entire system.
- Exhaustive testing is not possible. The number of path permutations for even a moderately sizes program is exceptionally large. For this reason, it is impossible to execute every combination of paths during testing. It is possible, however, to adequately cover program logic and to ensure that all conditions in the component level design have been exercised.
- To be most effective, testing should be conducted by an independent third party. By most effective, we mean testing that has the highest probability of finding errors (the primary objective of testing), the software engineer who created the system is not the best person to conduct all tests for the software.

TESTING STEPS

Software is tested from two different perspective

1. Internal program logic is exercised using “white box” test case design techniques.
2. Software requirements are exercised using “black box” test case design techniques.

In both cases, the intent is to find the maximum number of errors with the minimum amount of effort and time.

Black Box Testing

The technique of testing without having any knowledge of the interior workings of the application is Black Box Testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, when performing a black box test, a tester will interact with the system’s user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

White Box Testing

White box testing is the detailed investigation of internal logic and structure of the code. White box testing is also called glass testing or open box testing. In order to perform white box testing on an application, the tester needs to process knowledge of the internal working of the code.

The tester needs to have a look inside the source code and find out which unit/chunk of the

code is behaving inappropriately.

STAGES IN THE TESTING PROCESS

UNIT TESTING

Unit testing focuses verification effort on the smallest unit of software design the software component or module. Using the component-level design description as a guide, important control. paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered errors is limited by the constrained scope established for unit testing. The unit test is white-box oriented, and the step can be conducted in parallel for multiple components.

Limitations of Unit Testing

Testing cannot catch each and every bug in an application. It is impossible to evaluate every execution path in every software application. The same is the case with unit testing. There is a limit to the number of scenarios and test data the developer can use to verify the source code. So, after he has exhausted all options there is no choice but to step unit testing and merge the code segment with other units.

MODULE TESTING

Module is a collection of dependent components such as an object classes an abstract data type or some looser collection of procedures and functions. A module encapsulates related components so can be tested without other system modules.

INTEGRATION TESTING

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design.

VALIDATION TESTING

Software validation is achieved through a series of black-box tests that demonstrate conformity with requirements. A test plan outlines the classes of tests to be conducted and a test procedure defines specific test cases that will be used to demonstrate conformity with requirements. Both the plan and procedure are designed to ensure that all functional requirements are satisfied, all behavioural characteristics are achieved, all performance requirements are attained, documentation is correct, and human-engineered and other requirements are met.

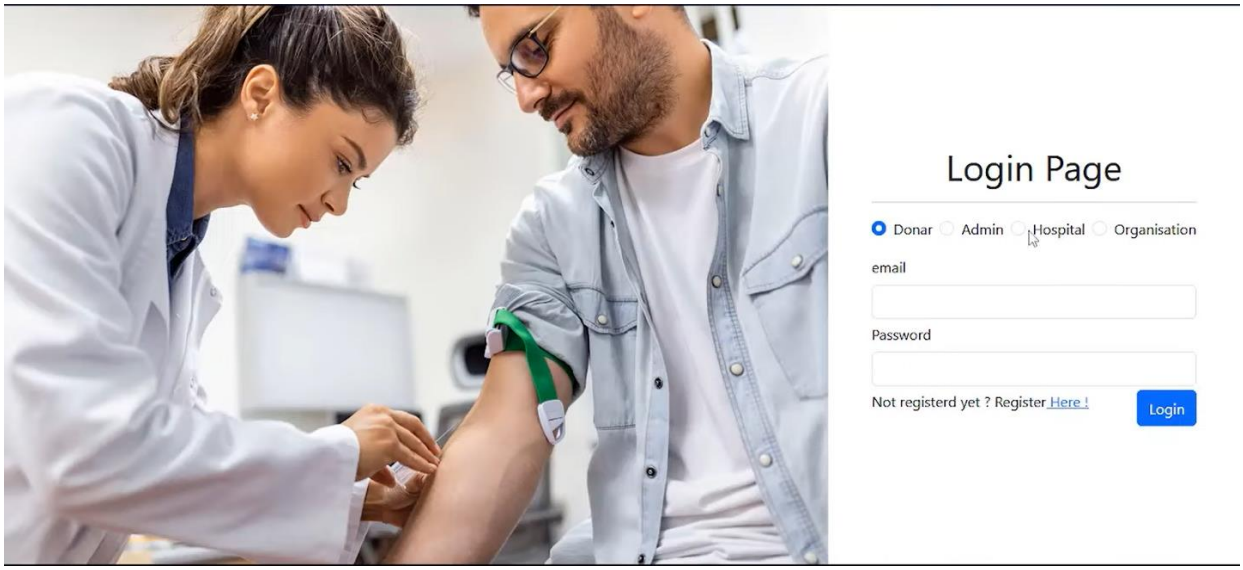
SYSTEM TESTING

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that system elements have been properly integrated and perform allocated functions.

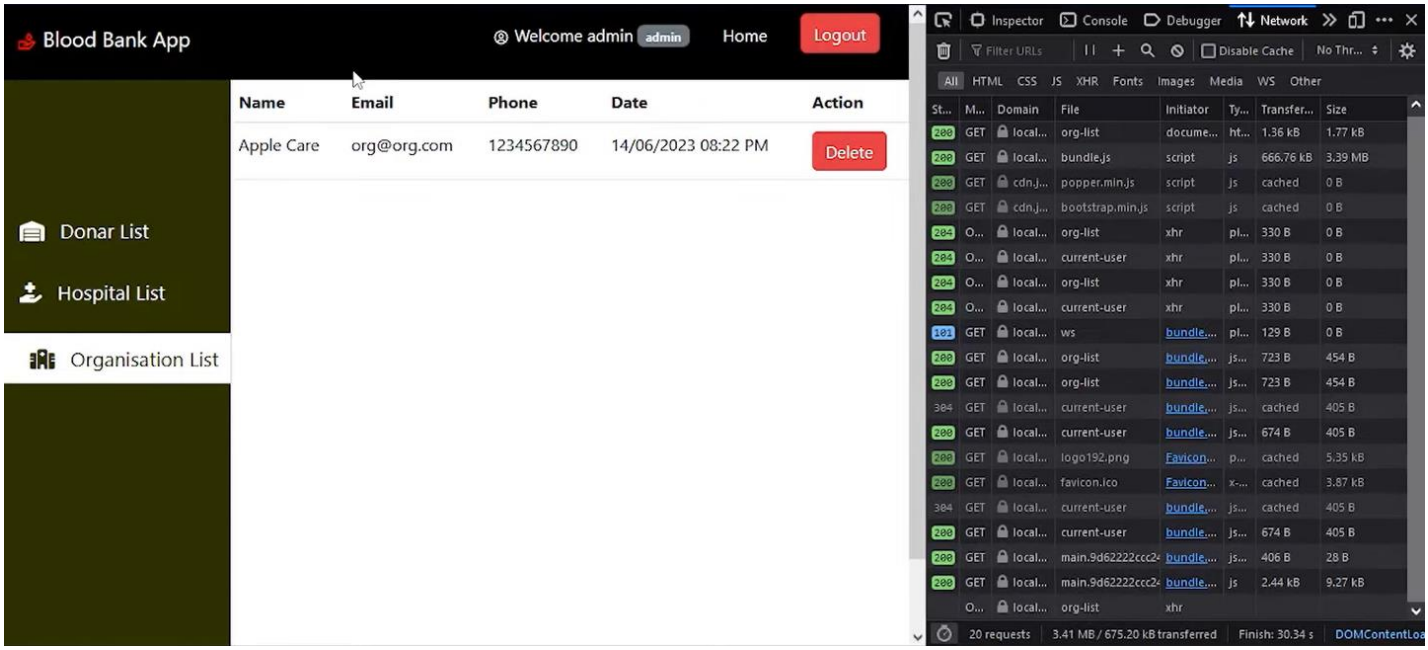
VI

DESIGN SNAPSHOTS

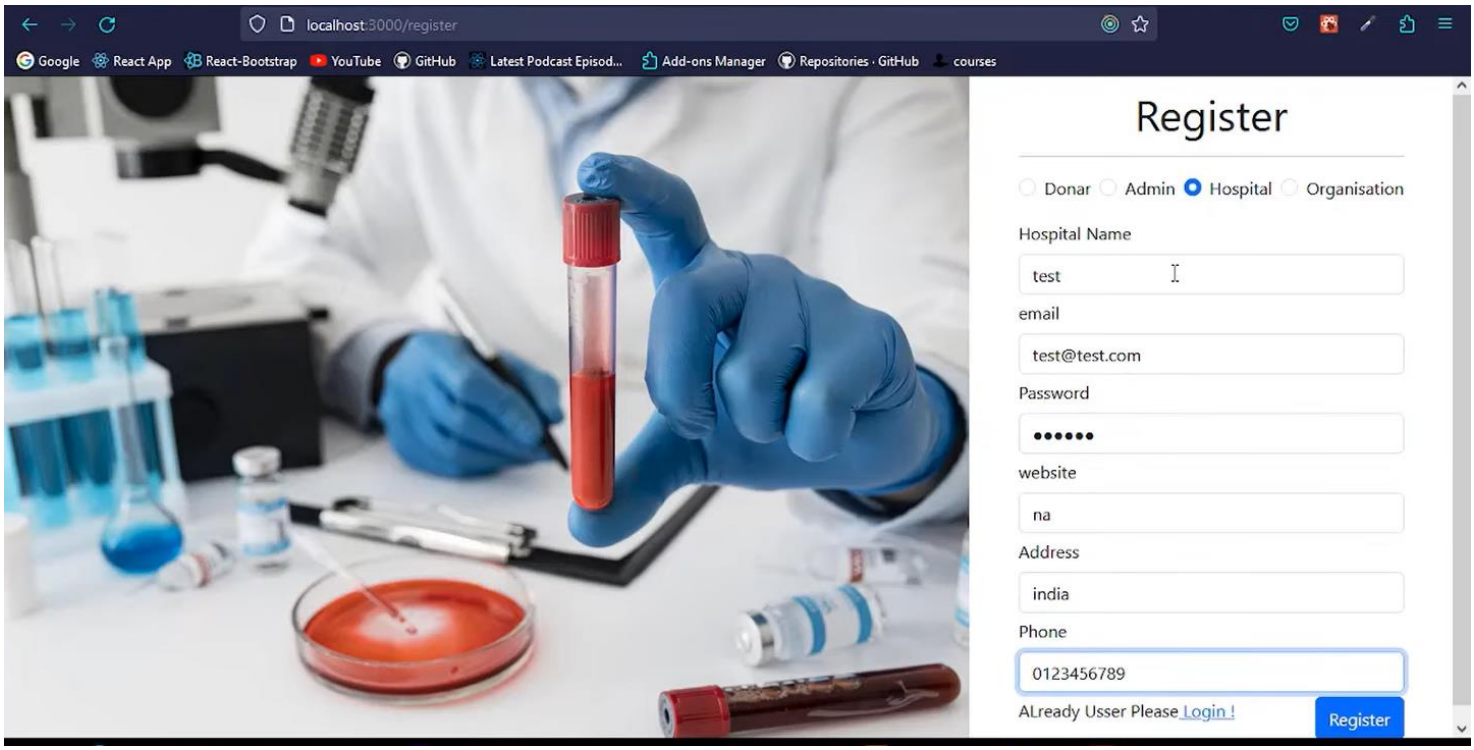
Login page:


Login Page
Donar
Admin
Hospital
Organisation
email
Password
Not registerd yet ? Register Here !
Login

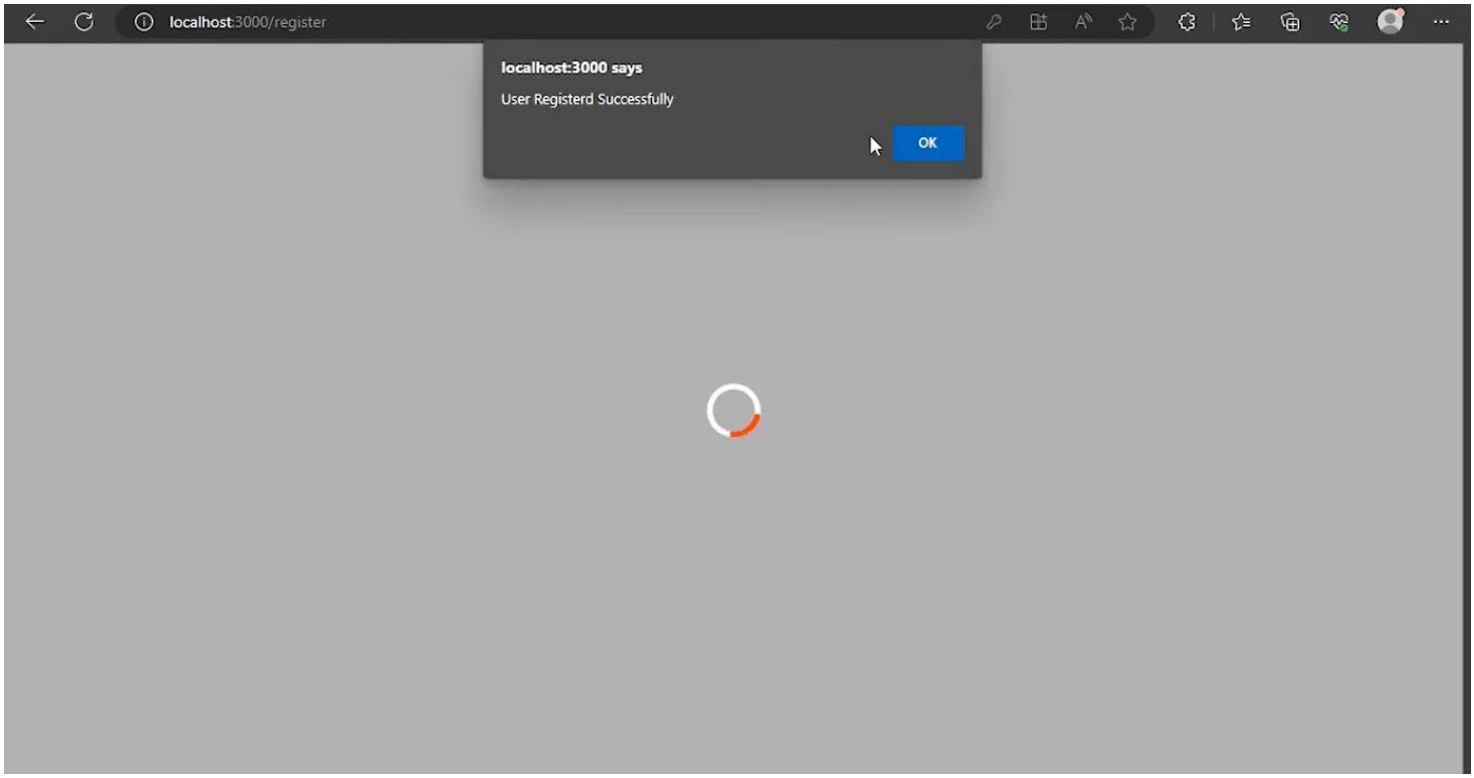
Management Dashboard:



Registration Page:



User Registration Successful :



VII

CONCLUSION

Conclusion

In conclusion, Streamlining blood bank management through advanced technology and efficient processes revolutionizes access to blood donations. Our system enables individuals to easily find the required blood through our website, enhancing accuracy and reducing administrative burdens. By improving inventory management and facilitating better coordination with real-time data access, we ensure that blood donations reach those in need efficiently, ultimately saving more lives and fostering a healthier community.

.

VII

BIBILOGRAPHY

BIBILOGRAPHY

1. www.github.com
2. www.youtube.com