

5. IMPLEMENTATION AND RESULTS

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve change over and evaluation of change over methods.

5.1.1 Modules

Modules work together to create a secure, transparent and decentralized system and also it uses the hash function. Hash functions are cryptographic algorithms. Hash functions help maintain the integrity of critical data such as donor information, donation records and supply chain transactions. This proposed system having various modules as follows:

- **Admin**
- **Blood Donor**
- **Blood Banker**
- **Hospital**



Fig 5.1.1 A schema of Block chain with unique hash code

Explanation: This Figure explain about how to data stored in each block and connected to each other like a chain and having unique hash code.

5.1.2 Admin Module

In this module, Admin will authorize Blood donors and Blood bankers and view all the files Blood details and will do the following operations:

- **View Blood Donors And Authorize**

In this Operation HTML,CSS, java, J2EE, mysql technologies are used.

- **View Blood Bankers And Authorize**

In this Operation HTML,CSS, java, J2EE, mysql technologies are used.

- **View Hospitals And Authorize**

In this Operation HTML,CSS, java, J2EE, mysql technologies are used.

- **Add Blood Groups**

In this Operation HTML,CSS, java, J2EE, mysql technologies are used and data store in MySQL with unique Hash code.

- **View All Blood Donors Transportation**

In this Operation HTML,CSS, java, J2EE, mysql technologies are used.

- **View Blood Donor Details By Blockchain**

In this Operation HTML,CSS, java, J2EE technologies are used and data store in MySQL with unique Hash code.

- **View Patient Details By Blockchain**

In this Operation HTML,CSS, java, J2EE technologies are used and data store in MySQL with unique Hash code.

- **View Donated Details By Blockchain**

In this Operation HTML,CSS, java, J2EE technologies are used and data store in MySQL with unique Hash code.

Admin Modules also consists of sub modules as follows:

- 1. Supply Chain Integrity Module:** It tracks each step of the donation process, verifies donor identities, monitors quality assurance measures and maintains a tamper-proof record of the chain of custody for each blood unit.
- 2. Decentralized Integrity Module:** This module utilizes distributed ledger technology to record and verify each step of the blood donation lifecycle, including donor registration, blood collection, processing and distribution.

3. User Identity Module: User Identity Module plays a pivotal role in securely managing and verifying the identities of donors, recipients and other stakeholders.

5.1.3 Blood Bankers Module

In this module, Blood banker has to register to both the hospitals and admin in order to get the blood from the donors and will do the following operations:

- **Register and Login**

In this Operation HTML,CSS,J2EE,mysql technologies are used.

- **View Blood Donor**

In this Operation HTML,CSS,J2EE,mysql technologies are used.

- **View Blood Detail Request**

In this Operation HTML,CSS,J2EE,mysql technologies are used.

- **View Request And Sale Bottle**

In this Operation HTML,CSS,J2EE,mysql technologies are used.

- **View Blood Available**

In this Operation HTML,CSS,J2EE,mysql technologies are used.

5.1.4 Blood Donor Module

In this module, Blood donor has to register to both hospital and Blood banker to get Blood and will do the following operations:

- **Register and Login**

In this Operation HTML,CSS,J2EE,mysql technologies are used.

- **Provide Blood to Hospitals**

In this Operation HTML,CSS,J2EE,mysql technologies are used.

- **Provide Blood to Blood Banker**

In this Operation HTML,CSS,J2EE,mysql technologies are used.

5.1.5 Hospital Module

In this module the Hospital will collect all blood from the donors and provides to the patients and will do the following operations:

- **Register and Login**

In this Operation HTML,CSS,J2EE,mysql technologies are used.

- **View Blood Donors**

In this Operation HTML,CSS,J2EE,mysql technologies are used.

- **Feed Patient Detail**

In this Operation HTML,CSS,J2EE,mysql technologies are used.

- **View Blood Banker & Required Blood**

In this Operation HTML,CSS,J2EE,mysql technologies are used.

- **View Blood Details**

In this Operation HTML,CSS,J2EE,mysql technologies are used.

- **View Patients**

In this Operation HTML,CSS,J2EE,mysql technologies are used.

5.2 Method of Implementation

Method of Implementation process can be done by using the following technologies and other concepts.

5.2.1 Java Technology

Java technology is both a programming language and a platform. The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

- Simple
- Architecture neutral
- Object oriented
- Portable
- Distributed
- High performance
- Interpreted
- Multithreaded
- Robust
- Dynamic

The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called Java byte

codes. The platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.

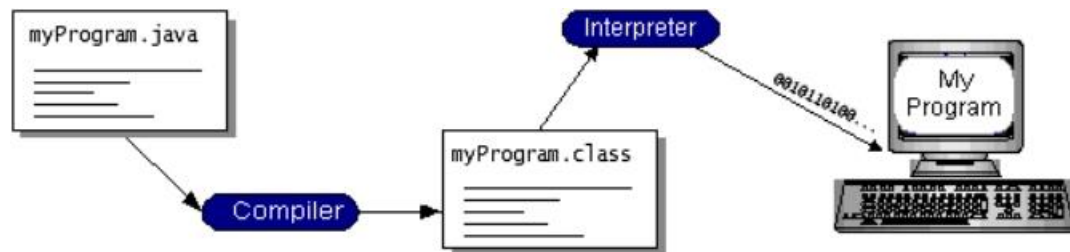


Fig 5.2.1.1 Flow diagram of Java Program

Explanation: This figure explain about the flow of the java program in various phases like compiler phase and interpreter phase.

Every Java interpreter, whether it's a development tool or a Web browser that can run applets, is an implementation of the JVM. Java byte codes help make “write once, run anywhere” possible. You can compile your program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the JVM. That means that as long as a computer has a JVM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation or on an iMac.

The Java Platform

A platform is the hardware or software environment in which a program runs. The Java platform differs from most other platforms in that it is a software-only platform that runs on top of other hardware-based platforms.

The Java platform has two components:

- The Java Virtual Machine (Java VM)
- The Java Application Programming Interface (Java API)

The Java API is a large collection of ready-made software components that provide many useful capabilities such as graphical user interface (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as packages. The next

section, What Can Java Technology Do? Highlights what functionality some of the packages in the Java API provide.

The following figure depicts a program that's running on the Java platform. As the figure shows, the Java API and the virtual machine insulate the program from the hardware.

Native code is code that after you compile it, the compiled code runs on a specific hardware platform. As a platform-independent environment, the Java platform can be a bit slower than native code. However, smart compilers, well-tuned interpreters, and just-in-time byte code compilers can bring performance close to that of native code without threatening portability. implementation of the Java platform gives the following features:

- **The essentials:** Objects, strings, threads, numbers, input and output, data structures, system properties, date and time, and so on.
- **Applets:** The set of conventions used by applets.
- **Networking:** URL (Uniform Resource Locator), TCP (Transmission Control Protocol), UDP (User Datagram Protocol) sockets and IP (Internet Protocol) addresses.
- **Internationalization:** It is a help for writing programs that can be localized for users worldwide. Programs can automatically adapt to specific locales and be displayed in the appropriate language.
- **Security:** Both low level and high level, including electronic signatures, public and private key management, access control, and certificates.
- **Software components:** It is also known as Java Beans TM. It can plug into existing component architectures.
- **Object serialization:** It allows lightweight persistence and communication via Remote Method Invocation (RMI).
- **Java Database Connectivity (JDBCTM):** It provides uniform access to a wide range of relational databases.

The Java platform also has APIs for 2D and 3D graphics, accessibility, servers, collaboration, telephony, speech, animation and more.

Client Server: Client server implementations are complex but the underlying concept is simple and powerful. A client is an application running with local resources but able to request the database and relate the services from separate remote server.

The software mediating this client server interaction is often referred to as MIDDLEWARE.

The typical client either a PC or a Work Station connected through a network to a more powerful PC, Workstation, Midrange or Main Frames server usually capable of handling request from more than one client. However, with some configuration server may also act as client. A server may need to access other server in order to process the original client request.

The key client server idea is that client as user is essentially insulated from the physical location and formats of the data needs for their application. With the proper middleware, a client input from or report can transparently access and manipulate both local database on the client machine and remote databases on one or more servers. An added bonus is the client server opens the door to multi-vendor database access indulging heterogeneous table joins.

What is a Client Server ?

Two prominent systems in existence are client server and file server systems. It is essential to distinguish between client servers and file server systems and both provides shared network access to data but the comparison dens there! The file server simply provides a remote disk drive that can be accessed by LAN applications on a file-by-file basis. The client server offers full relational database services such as SQL-Access, Record modifying, Insert, Delete with full relational integrity backup restore performance for high volume of transactions, etc. the client server middleware provides a flexible interface between client and server, who does what, when and to whom.

Why Client Server ?

Client server has evolved to solve a problem that has been around since the earliest days of computing: how best to distribute your computing, data generation and data storage resources in order to obtain efficient, cost effective departmental an enterprise-wide data processing. During mainframe era choices were quite limited. A central machine housed both the CPU and DATA (cards, tapes, drums and later disks). It access to these resources was initially confined to batched runs that produced departmental reports at the appropriate intervals. A strong central information service department ruled the corporation. The role of the rest of the corporation limited to requesting new or more frequent reports and to provide hand written forms from which the central data banks were created and updated. The earliest client server solutions therefore could best be characterized as “SLAVE-MASTER”.

❖ User Interface Design

The entire user interface is planned to be developed in browser specific environment with a touch of Intranet-based architecture for achieving the distributed concept.

The browser specific components are designed by using the HTML standards and the dynamism of the designed by concentrating on the constructs of the Java Server Pages.

❖ **Data Base Tier Connectivity**

The Communication architecture is designed by concentrating on the Standards of Servlets and Enterprise Java Beans. The database connectivity is established by using the Java Data Base Connectivity.

The standards of three-tier architecture are given major concentration to keep the standards of higher cohesion and limited coupling for effectiveness of the operations.

❖ **Importance of Java in internet**

Java has had a profound effect on the Internet. This is because of Java expands the Universe of objects that can move about freely in Cyberspace. In a network, two categories of objects are transmitted between the Server and the Personal computer. They are: Passive information and Dynamic active programs. The Dynamic, Self-executing programs cause serious problems in the areas of Security and probability but, Java addresses those concerns and by doing so, has opened the door to an exciting new form of program called the Applet.

JAVASCRIPT

JavaScript is a script-based programming language that was developed by Netscape Communication Corporation. JavaScript was originally called Live Script and renamed as JavaScript to indicate its relationship with Java. JavaScript supports the development of both client and server components of Web-based applications. On the client side, it can be used to write programs that are executed by a Web browser within the context of a Web page. On the server side, it can be used to write Web server programs that can process information submitted by a Web browser and then updates the browser's display accordingly.

Even though JavaScript supports both client and server Web programming, we prefer JavaScript at Client side programming since most of the browsers supports it. JavaScript is

almost as easy to learn as HTML and JavaScript statements can be included in HTML documents by enclosing the statements between a pair of scripting tags

```
<SCRIPTS>..  
</SCRIPT>.
```

```
<SCRIPT LANGUAGE = "JavaScript">
```

```
JavaScript statements
```

```
</SCRIPT>
```

Here are a few things we can do with JavaScript :

- Validate the contents of a form and make calculations.
- Add scrolling or changing messages to the Browser's status line.
- Animate images or rotate images that change when we move the mouse over them.
- Detect the browser in use and display different content for different browsers.
- Detect installed plug-ins and notify the user if a plug-in is required.

We can do much more with JavaScript including creating entire application.

Hyper Text Markup Language

Hypertext Markup Language (HTML) the languages of the World Wide Web (WWW), allows users to produces Web pages that include text, graphics and pointer to other Web pages (Hyperlinks).

HTML is not a programming language but it is an application of ISO Standard 8879, SGML (Standard Generalized Markup Language), but specialized to hypertext and adapted to the Web. The idea behind Hypertext is that instead of reading text in rigid linear structure. we can easily jump from one point to another point. We can navigate through the information based on our interest and preference. A markup language is simply a series of elements, each delimited with special characters that define how text or other items enclosed within the elements should be displayed. Hyperlinks are underlined or emphasized works that load to other documents or some portions of the same document.

HTML can be used to display any type of document on the host computer, which can be geographically at a different location. It is a versatile language and can be used on any platform or desktop.

HTML provides tags (special codes) to make the document look attractive. HTML tags are not case-sensitive. Using graphics, fonts, different sizes, colour, etc., can enhance the presentation of the document. Anything that is not a tag is part of the document itself.

Basic HTML Tags :

<!-- -->	Specifies comments
<A>.....	Creates hypertext links
.....	Formats text as bold
<BIG>.....</BIG>	Formats text in large font.
<BODY>...</BODY>	Contains all tags and text in the HTML document
<CENTER>...</CENTER>	Creates text
<DD>...</DD>	Definition of a term
<DL>...</DL>	Creates definition list
...	Formats text with a particular font
<FORM>...</FORM>	Encloses a fill-out form
<FRAME>...</FRAME>	Defines a particular frame in a set of frames
<H#>...</H#>	Creates headings of different levels
<HEAD>...</HEAD>	Contains tags that specify information about a document
<HR>...</HR>	Creates a horizontal rule
<HTML>...</HTML>	Contains all other HTML tags
<META>...</META>	Provides meta-information about a document
<SCRIPT>...</SCRIPT>	Contains client-side or server-side script
<TABLE>...</TABLE>	Creates a table
<TD>...</TD>	Indicates table data in a table

<code><TR>...</TR></code>	Designates a table row
<code><TH>...</TH></code>	Creates a heading in a table

ADVANTAGES

- A HTML document is small and hence easy to send over the net. It is small because it does not include formatted information.
- HTML is platform independent.
- HTML tags are not case-sensitive.

Java Database Connectivity

JDBC is a Java API for executing SQL statements. (As a point of interest, JDBC is a trademarked name and is not an acronym, nevertheless, JDBC is often thought of as standing for Java Database Connectivity. It consists of a set of classes and interfaces written in the Java programming language. JDBC provides a standard API for tool database developers and makes it possible to write database applications using a pure Java API.

Using JDBC, it is easy to send SQL statements to virtually any relational database. One can write a single program using the JDBC API and the program will be able to send SQL statements to the appropriate database. The combinations of Java and JDBC lets a programmer write it once and run it anywhere. Simply put JDBC makes it possible to do three things:

- Establish a connection with a database
- Send SQL statements
- Process the results

Two-tier and Three-tier Models

The JDBC API supports both two-tier and three-tier models for database access. In the two-tier model, a Java applet or application talks directly to the database. This requires a JDBC driver that can communicate with the particular database management system being accessed. A user's SQL statements are delivered to the database and the results of those statements are sent back to the user. The database may be located on another machine to which the user is connected via a network. This is referred to as a client-server configuration, with the users machine as the

client, and the machine housing the database as the server. The network can be an Intranet, which for example connects employees within a corporation or it can be the Internet.

In the three-tier model, commands are sent to a "middle tier" of services, which then send SQL statements to the database. The database processes the SQL statements and sends the results back to the middle tier, which then sends them to the user. MIS directors find the three-tier model very attractive because the middle tier makes it possible to maintain control over access and the kinds of updates that can be made to corporate data. Another advantage is that when there is a middle tier, the user can employ an easy-to-use higher-level API which is translated by the middle tier into the appropriate low-level calls. Finally, in many cases the three-tier architecture can provide performance advantages.

JDBC Driver Types

The JDBC drivers that we are aware of at this time fit into one of four categories:

- JDBC-ODBC bridge plus ODBC driver
- Native-API partly-Java driver
- JDBC-Net pure Java driver
- Native-protocol pure Java driver

▪ JDBC-ODBC Bridge

If possible, use a Pure Java JDBC driver instead of the Bridge and an ODBC driver. This completely eliminates the client configuration required by ODBC. It also eliminates the potential that the Java VM could be corrupted by an error in the native code brought in by the Bridge (that is, the Bridge native library, the ODBC driver manager library, the ODBC driver library, and the database client library).

▪ Java Server Pages (JSP)

Java server Pages is a simple, yet powerful technology for creating and maintaining dynamic-content web pages. It is based on the Java programming language, Java Server Pages offers proven portability, open standards and a mature re-usable component model. The Java Server Pages architecture enables the separation of content generation from content presentation. This separation not eases maintenance headaches. It also allows web team members to focus on their

areas of expertise. Now, web page designer can concentrate on layout and web application designers on programming, with minimal concern about impacting each other's work.

Features of JSP

➤ Portability

Java Server Pages files can be run on any web server or web-enabled application server that provides support for them. Dubbed the JSP engine, this support involves recognition, translation and management of the Java Server Page life cycle and its interaction components.

➤ Components

It was mentioned earlier that the Java Server Pages architecture can include reusable Java components. The architecture also allows for the embedding of a scripting language directly into the Java Server Pages file. The components currently supported include Java Beans and Servlets.

➤ Processing

A Java Server Pages file is essentially an HTML document with JSP scripting or tags. The Java Server Pages file has a JSP extension to the server as a Java Server Pages file. Before the page is served, the Java Server Pages syntax is parsed and processed into a Servlet on the server side. The Servlet that is generated outputs real content in straight HTML for responding to the client.

➤ Access Models

A Java Server Pages file may be accessed in at least two different ways. A client's request comes directly into a Java Server Page. In this scenario, suppose the page accesses reusable Java Bean components that perform particular well-defined computations like accessing a database. The result of the Beans computations called result sets is stored within the Bean as properties. The page uses such Beans to generate dynamic content and present it back to the client.

In both of the above cases, the page could also contain any valid Java code. Java Server Pages architecture encourages separation of content from presentation.

Steps in the execution of a JSP Application

1. The client sends a request to the web server for a JSP file by giving the name of the JSP file within the form tag of a HTML page.
2. This request is transferred to the Java Web Server. At the server side Java Web Server receives the request and if it is a request for a spy file server gives this request to the JSP engine.
3. JSP engine is program which can understands the tags of the jsp and then it converts those tags into a Servlet program and it is stored at the server side. This Servlet is loaded in the memory and then it is executed and the result is given back to the Java Web Server and then it is transferred back to the result is given back to the Java Web Server and then it is transferred back to the client.

JDBC Connectivity :The JDBC provides database-independent connectivity between the J2EE platform and a wide range of tabular data sources. JDBC technology allows an Application Component Providers to:

- Perform connection and authentication to a database server
- Manager transactions
- Move SQL statements to a database engine for preprocessing and execution
- Execute stored procedures
- Inspect and modify the results from Select statements

Tomcat 9.0 Web server

Tomcat is an open source web server developed by Apache Group. Apache Tomcat is the servlet container that is used in the official Reference implementation for the Java Servlet and Java Server Pages technologies.

The Java Servlet and Java Server Pages specifications are developed by Sun under the Java Community Process.

Web Servers like Apache Tomcat support only web components while an application server supports web components as well as business components (BEAs Web logic, is one of the popular application server).

To develop a web application with jsp, servlet install any web server like Tomcat etc to run your application.

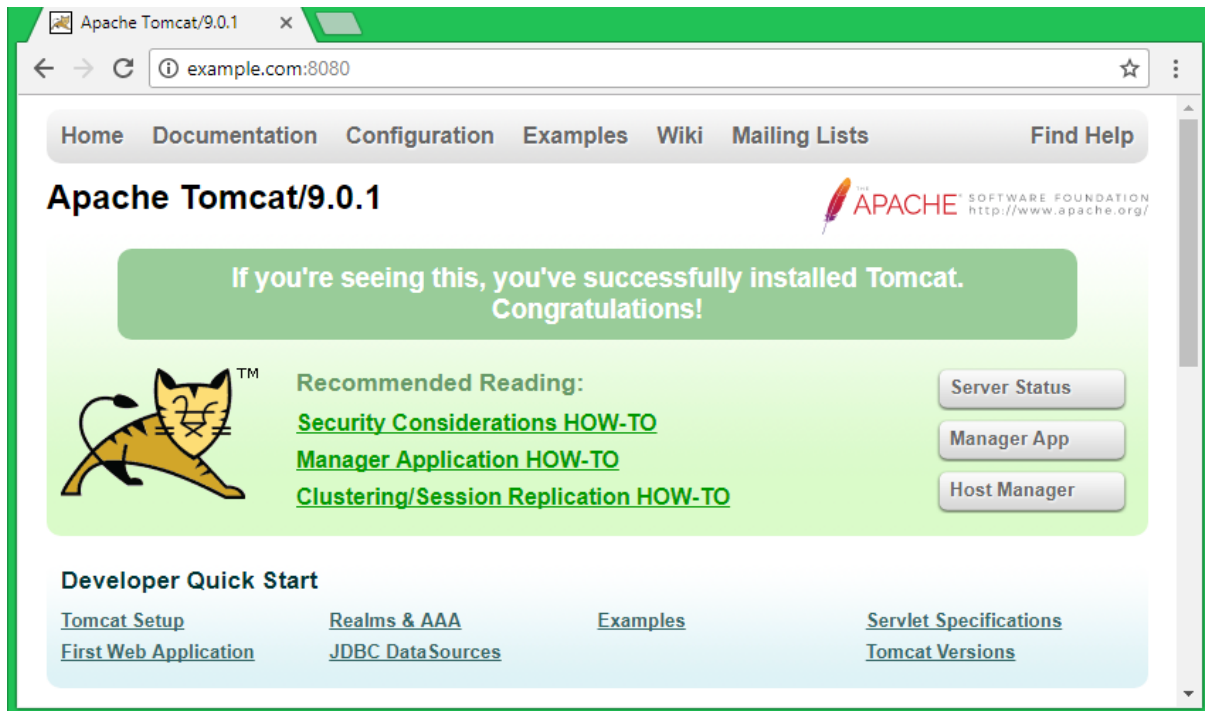


Fig 5.2.1.2 Apache Tomcat/9.0.1 Server

Explanation: The above figure explains about the Apache Tomcat Server 9.0.1 version, this server is useful for display dashboard of this project.

Networking

In Networking there are many protocols available and this protocols are used in this project in various stages.

- **IP datagram's**

The IP layer provides a connectionless and unreliable delivery system. It considers each datagram independently of the others. The IP layer handles routing through an Internet. It is also responsible for breaking up large datagram into smaller ones for transmission and reassembling them at the other end.

- **UDP (User datagram Protocol)**

UDP is also connectionless and unreliable. What it adds to IP is a checksum for the contents of the datagram and port numbers. These are used to give a client-server model.

- **TCP (Transmission control Protocol)**

TCP supplies logic to give a reliable connection-oriented protocol above IP. It provides a virtual circuit that two processes can use to communicate.

- **Internet addresses**

In order to use a service, you must be able to find it. The Internet uses an address scheme for machines so that they can be located.

The address is a 32 bit integer which gives the IP address. This encodes a network ID and more addressing. The network ID falls into various classes according to the size of the network address.

- **Network address**

Class A uses 8 bits for the network address with 24 bits left over for other addressing. Class B uses 16 bit network addressing.

Class C uses 24 bit network addressing and class D uses all 32. Subnet address.

- **Host address**

8 bits are finally used for host addresses within our subnet. This places a limit of 256 machines that can be on the subnet.

- **Port addresses**

A service exists on a host, and is identified by its port. This is a 16 bit number. To send a message to a server, you send it to the port for that service of the host that it is running on.

MIDP contains the following packages, the first three of which are core CLDC packages, plus three MIDP-specific packages.

- java.lang
- java.io
- java.util

- javax.microedition.io
- javax.microedition.lcdui
- javax.microedition.rm

5.3 Output Screens

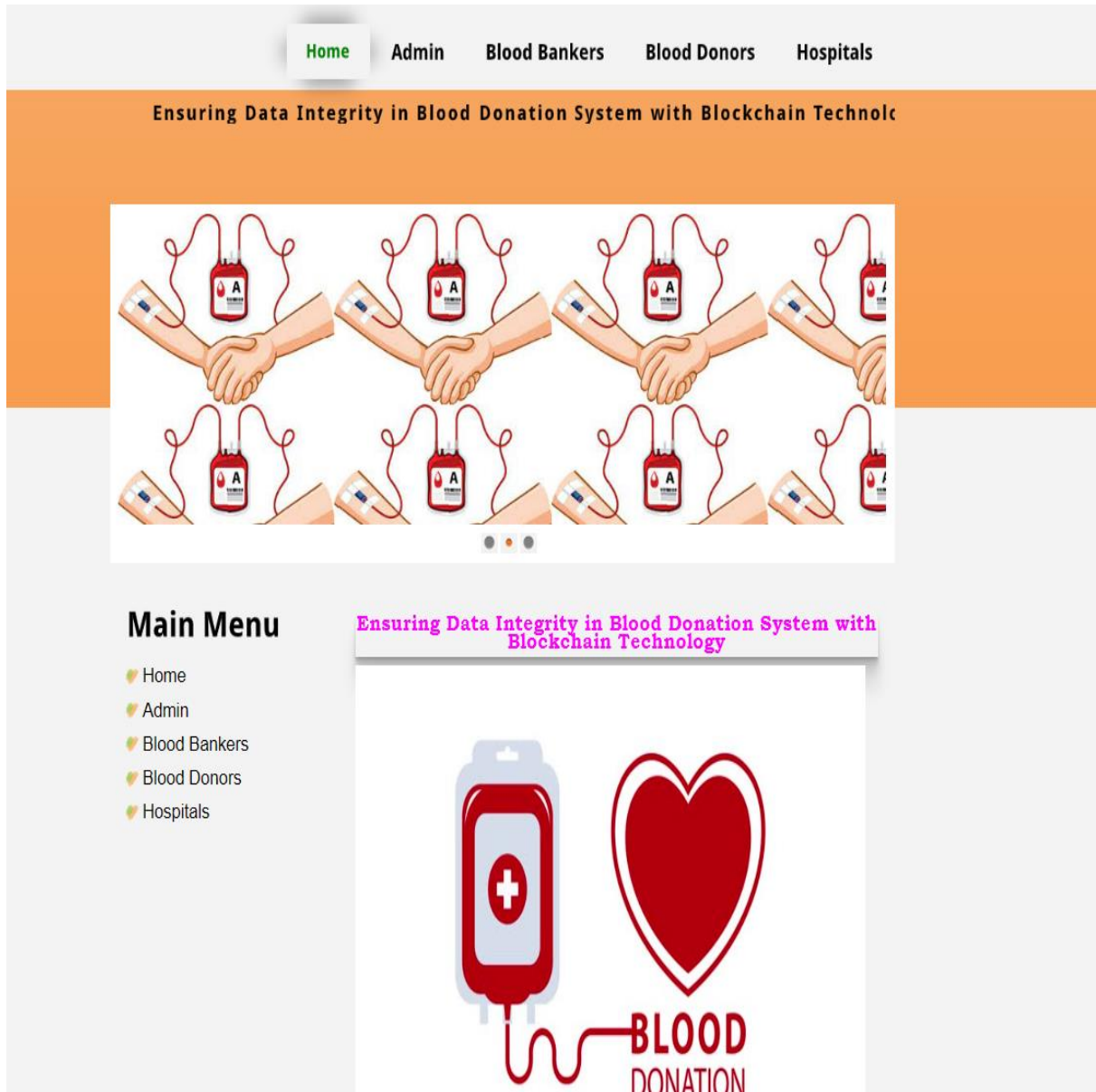


Fig 5.3.1.1 : Home Module and Main menu

Explanation: The above figure explain about home page of our project and display Main menu.

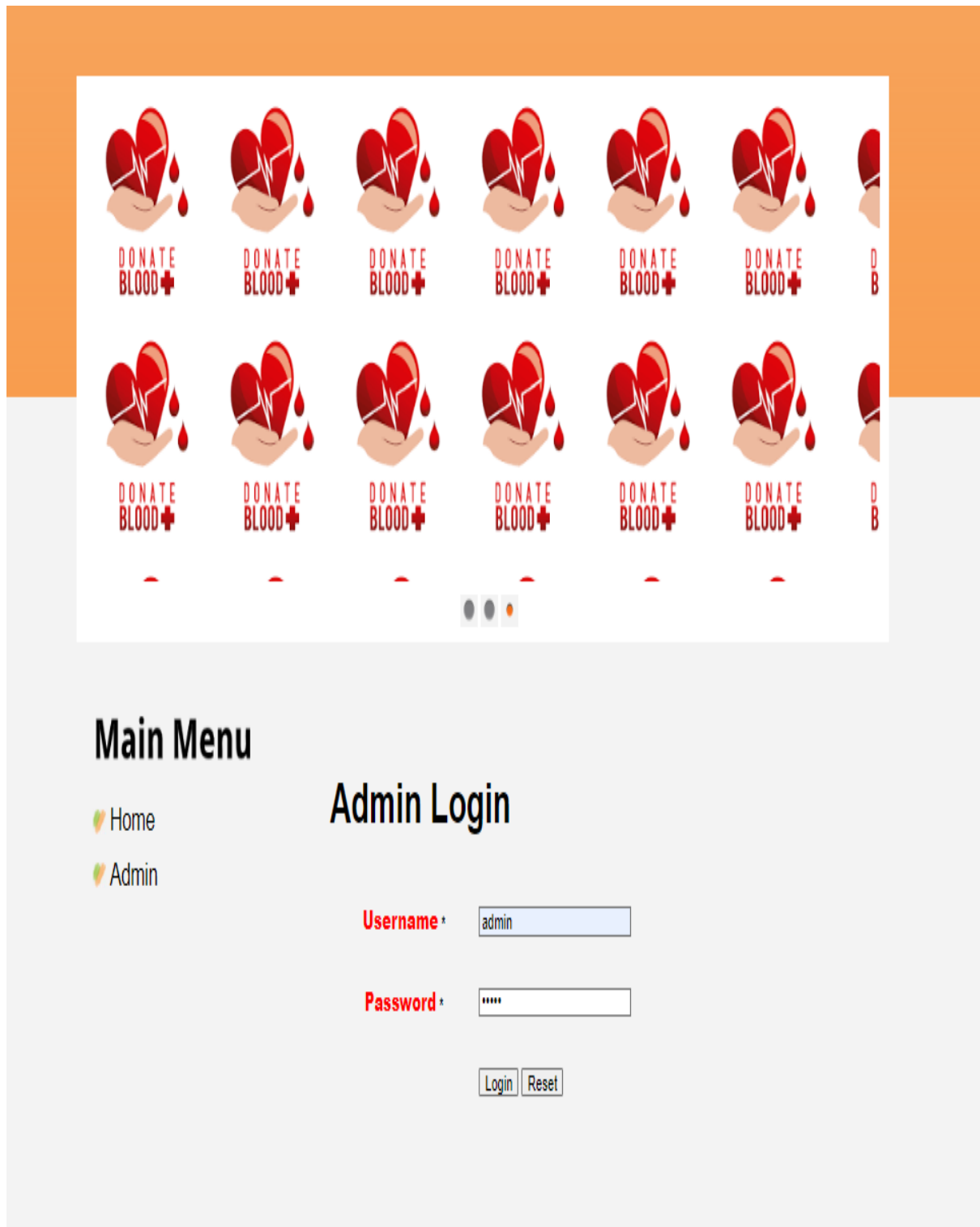


Fig 5.3.1.2 Admin Login

Explanation : The above figure explain about the login details of the admin in the Main Menu of the Dashboard.

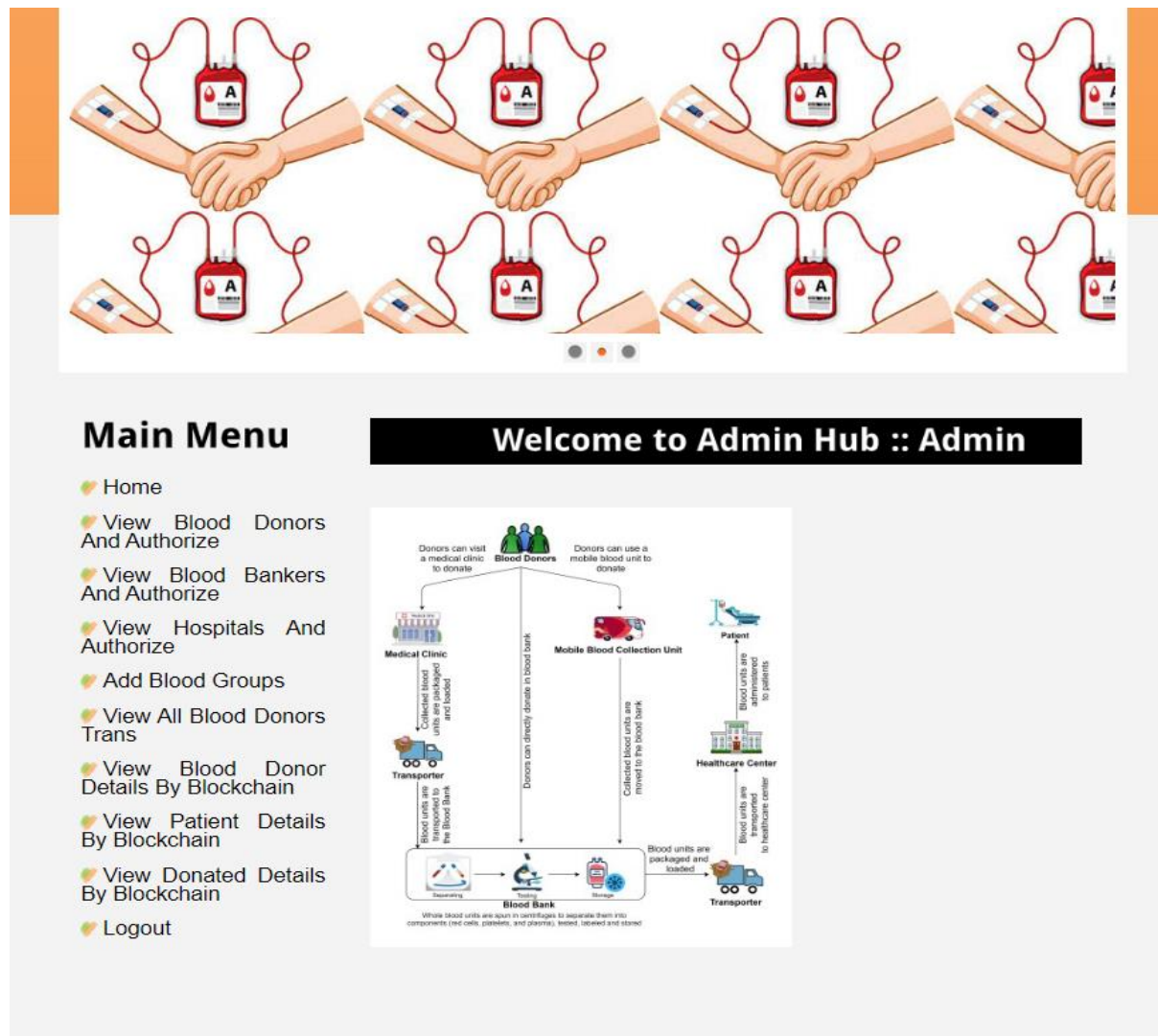


Fig 5.3.1.3: Admin Module main menu and its Operations

Explanation: The above figure explains about the Admin Module Operation and give the Authorization to the remaining Modules.

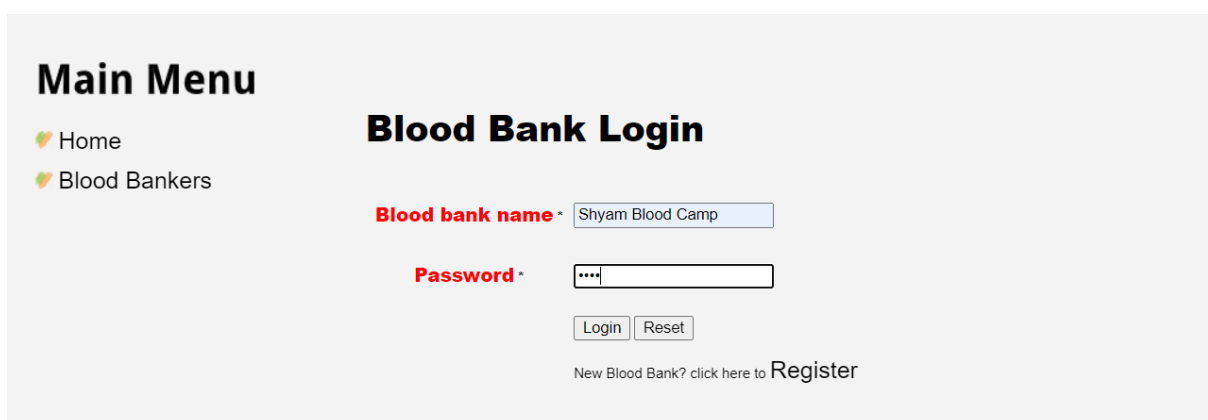


Fig 5.3.1.4 Blood Bank Login

Explanation: The above figure explains about the how the Blood Bank login with their details.

Main Menu

- 🍷 Home
- 🍷 Admin
- 🍷 Blood Bankers
- 🍷 Blood Donors
- 🍷 Hospitals

Register Here !!!

Name (required)

Password (required)

Address

Email Address (required)

Mobile Number (required)

Select Picture (required)

 slide2.jpg

Fig 5.3.1.5 Blood Bank Registration

Explanation: The above figure explains about the blood banks registration process with their valid details. The blood banker must register first then login to the blood bank module.

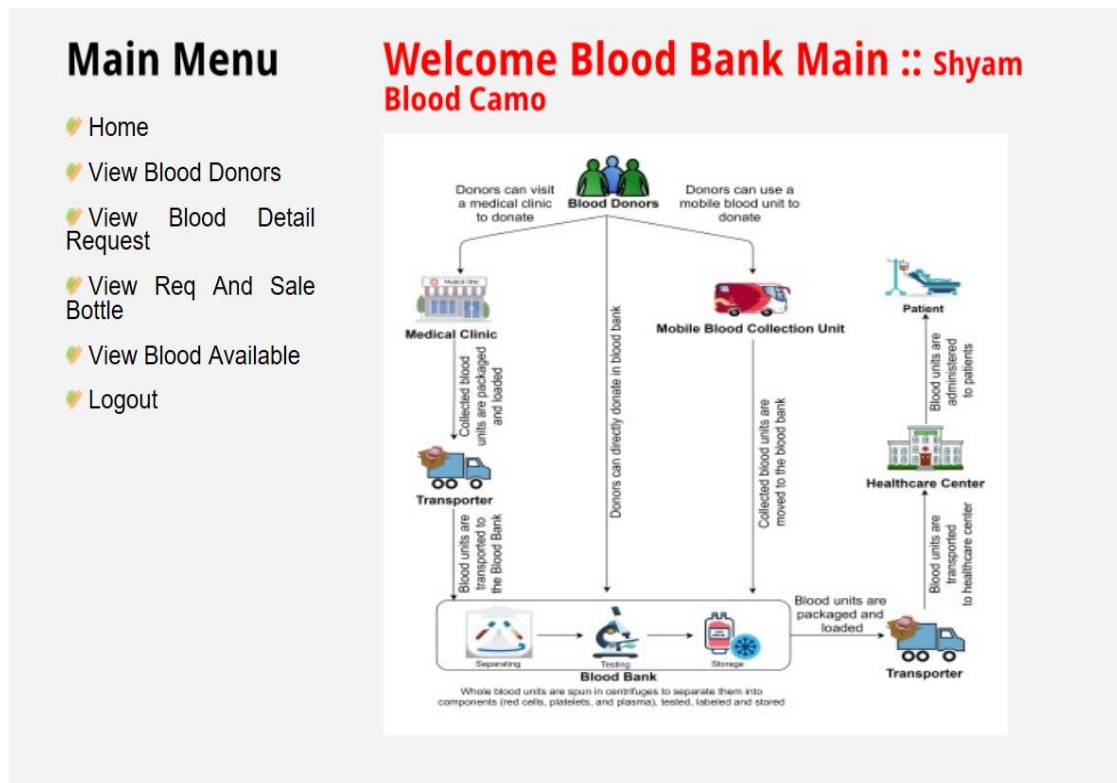


Fig 5.3.1.6 Blood Bank Module main menu and its Operations

Explanation : The above figure explains about the Blood bank main Module operations and methods.



Fig 5.3.1.7: Blood Bank details securely stored in the Database

Explanation: This figure explains about the Blood Bank details and storage of the blood bank details with the help of Blockchain technology with unique hash code.

Main Menu

- Home
- Blood Donors

Blood donor Login

User Name (required)

Password (required)

Blood group

B Positive ▾

Your Address

Email Address (required)

Mobile Number (required)

Date of Birth (required)

Select Gender (required)

Male ▾

Enter Pincode (required)

Enter Location (required)

Select Profile Picture (required)

images (3).jpg

Fig 5.3.1.8: Blood Donors Registration

Explanation: This figure explains about the Blood Donors registration Process with the valid details. This process is very important to the blood donors.

Main Menu

- Home
- Blood Donor

Blood Donor Login

Name (required)

Password (required)

New Blood Donor? click here to [Register](#)

Fig 5.3.1.9: Blood Donor Login

Explanation: This above figure explains about the blood donor login with their details

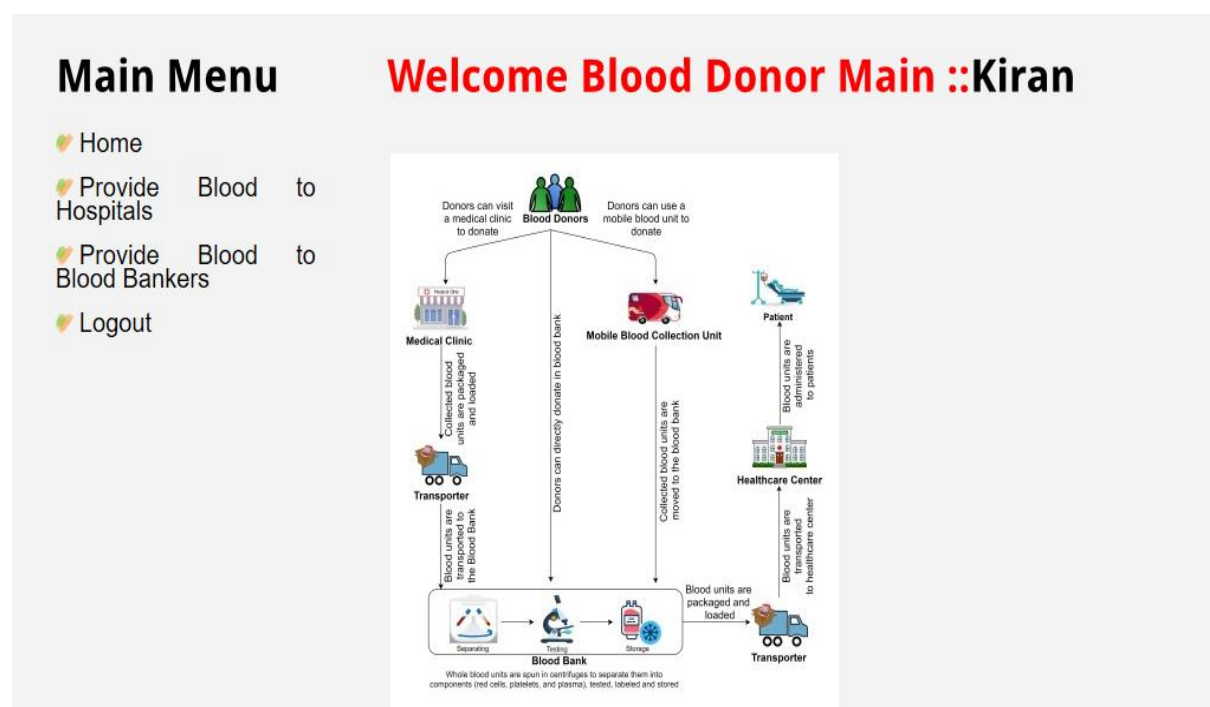
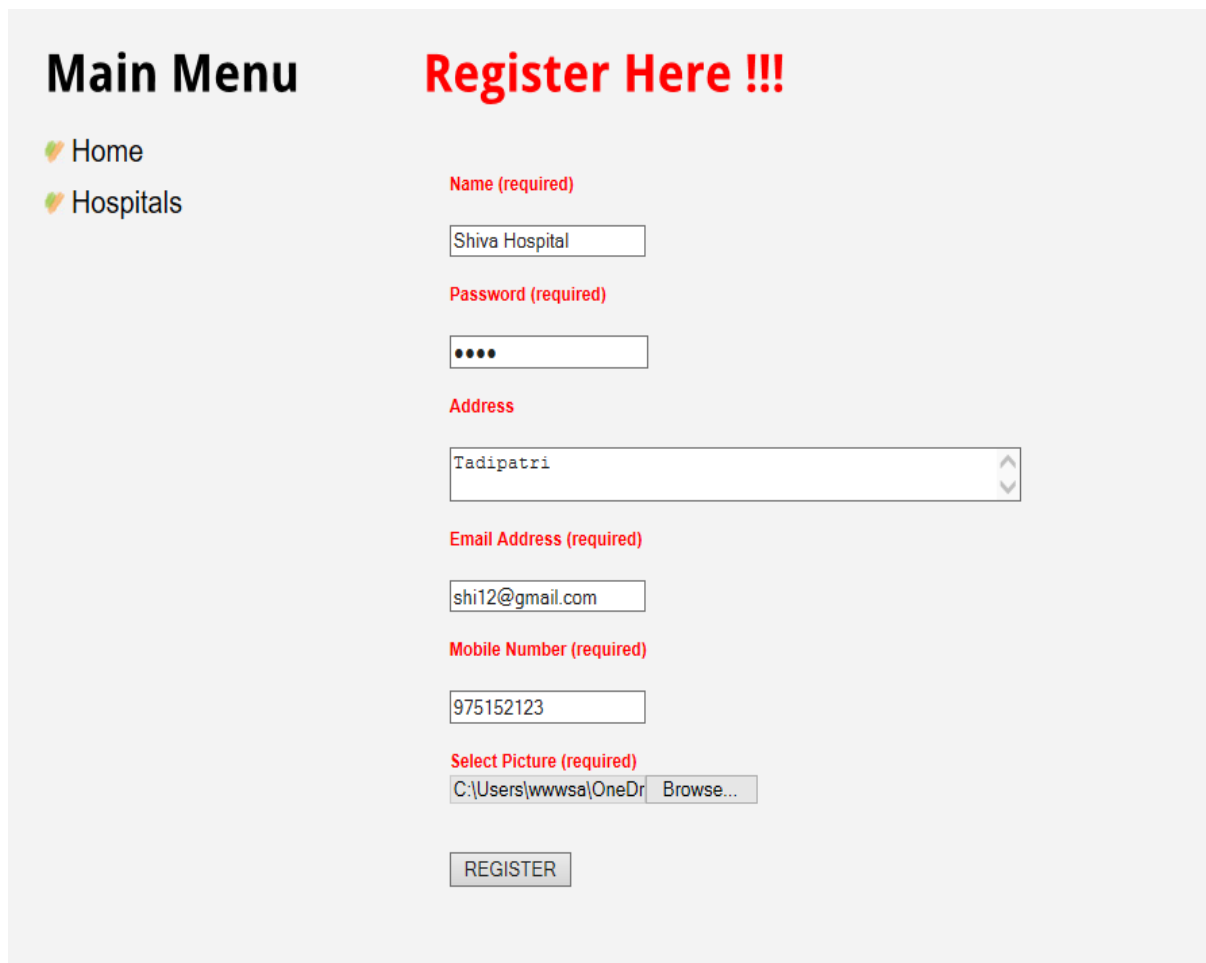


Fig: 5.3.1.10: Blood Donor Module main menu and its Operations

Explanation: The above screen explains about the blood donor entering into the blood donor main block and also able to do all operations in it.



The registration form is titled "Register Here !!!" in red. It includes a "Main Menu" on the left with links to Home and Hospitals. The form fields are: Name (required) with value "Shiva Hospital", Password (required) with masked characters, Address with value "Tadipatri", Email Address (required) with value "shi12@gmail.com", Mobile Number (required) with value "975152123", and a "Select Picture (required)" field with a file path and a "Browse..." button. A "REGISTER" button is at the bottom.

Main Menu

- Home
- Hospitals

Register Here !!!

Name (required)

Password (required)

Address

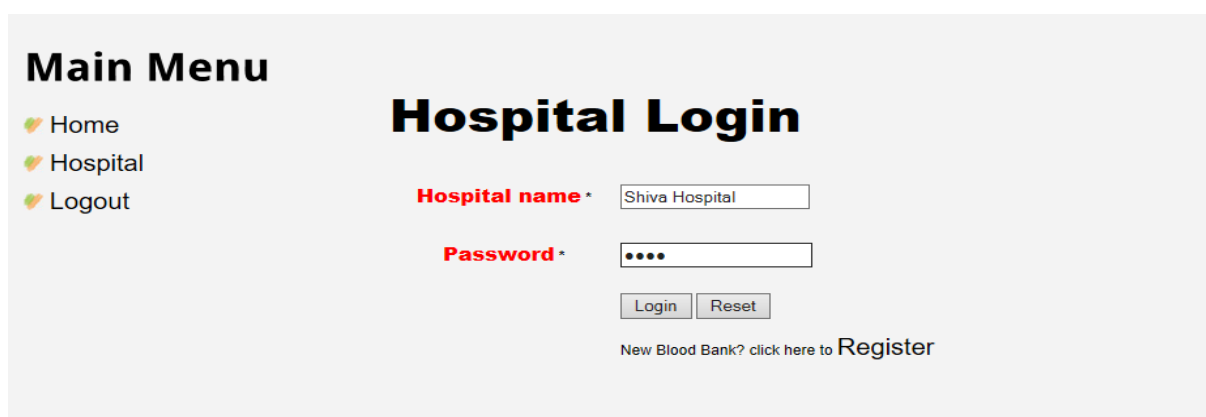
Email Address (required)

Mobile Number (required)

Select Picture (required)

Fig 5.3.1.11: Hospital Registration

Explanation: This above figure explains about the Hospital registration process and also explain for waiting for the admin authorization.



The login form is titled "Hospital Login". It includes a "Main Menu" on the left with links to Home, Hospital, and Logout. The form fields are: "Hospital name *" with value "Shiva Hospital" and "Password *" with masked characters. There are "Login" and "Reset" buttons. A link "New Blood Bank? click here to Register" is at the bottom.

Main Menu

- Home
- Hospital
- Logout

Hospital Login

Hospital name *

Password *

New Blood Bank? click here to [Register](#)

Fig 5.3.1.12: Hospital Login

Explanation: The figure explains how hospital is login with their hospital name with the

hospital name and password.

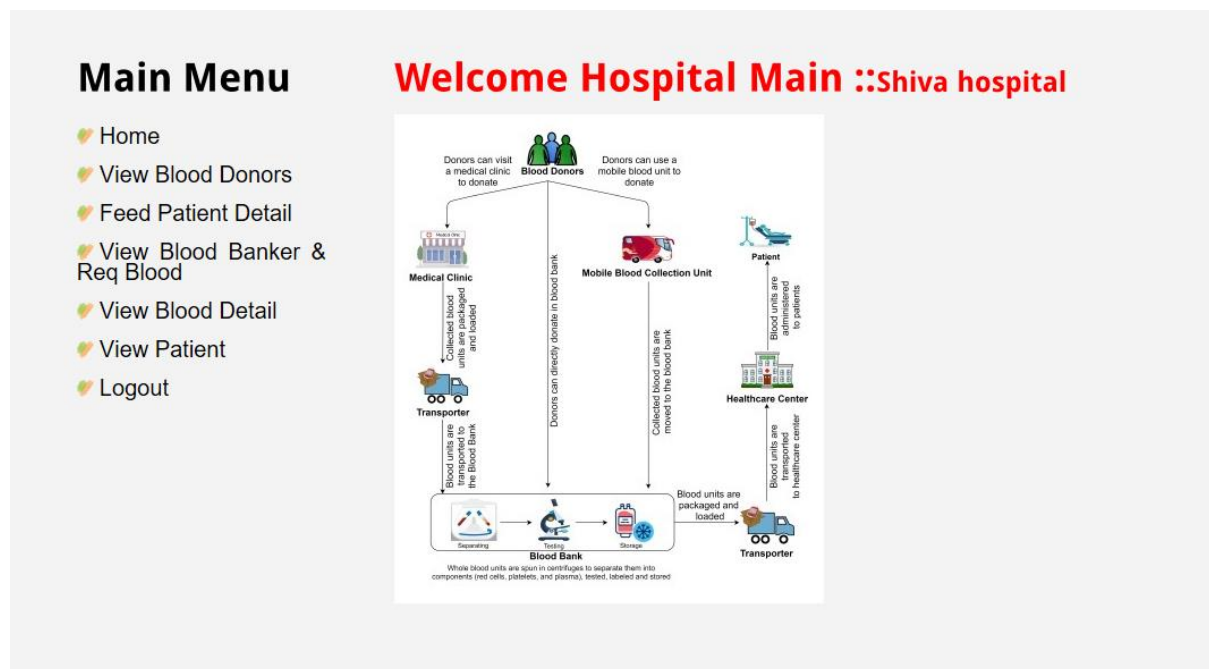


Fig 5.3.1.13: Hospital Module main menu and its Operations

Explanation: The above figure explains about the Shiva hospital main block operations and methods.

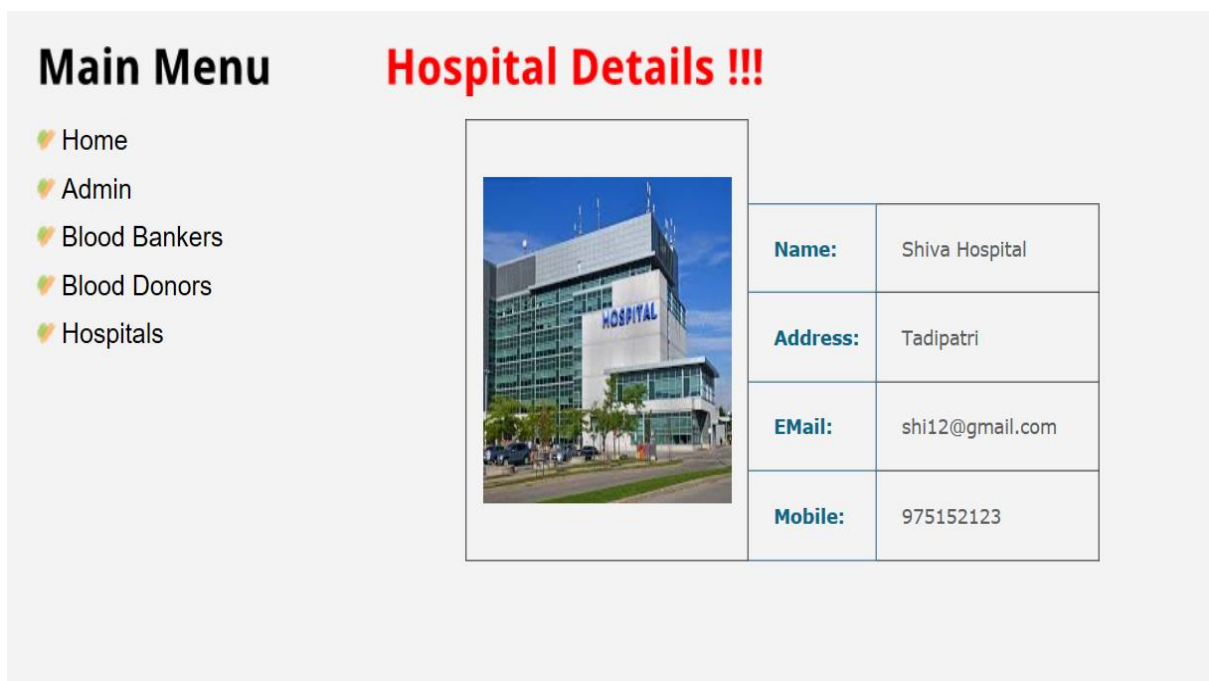


Fig 5.3.1.14: Hospital details securely stored in the Database

Explanation: This above figure explains about the hospital details stored in the database.

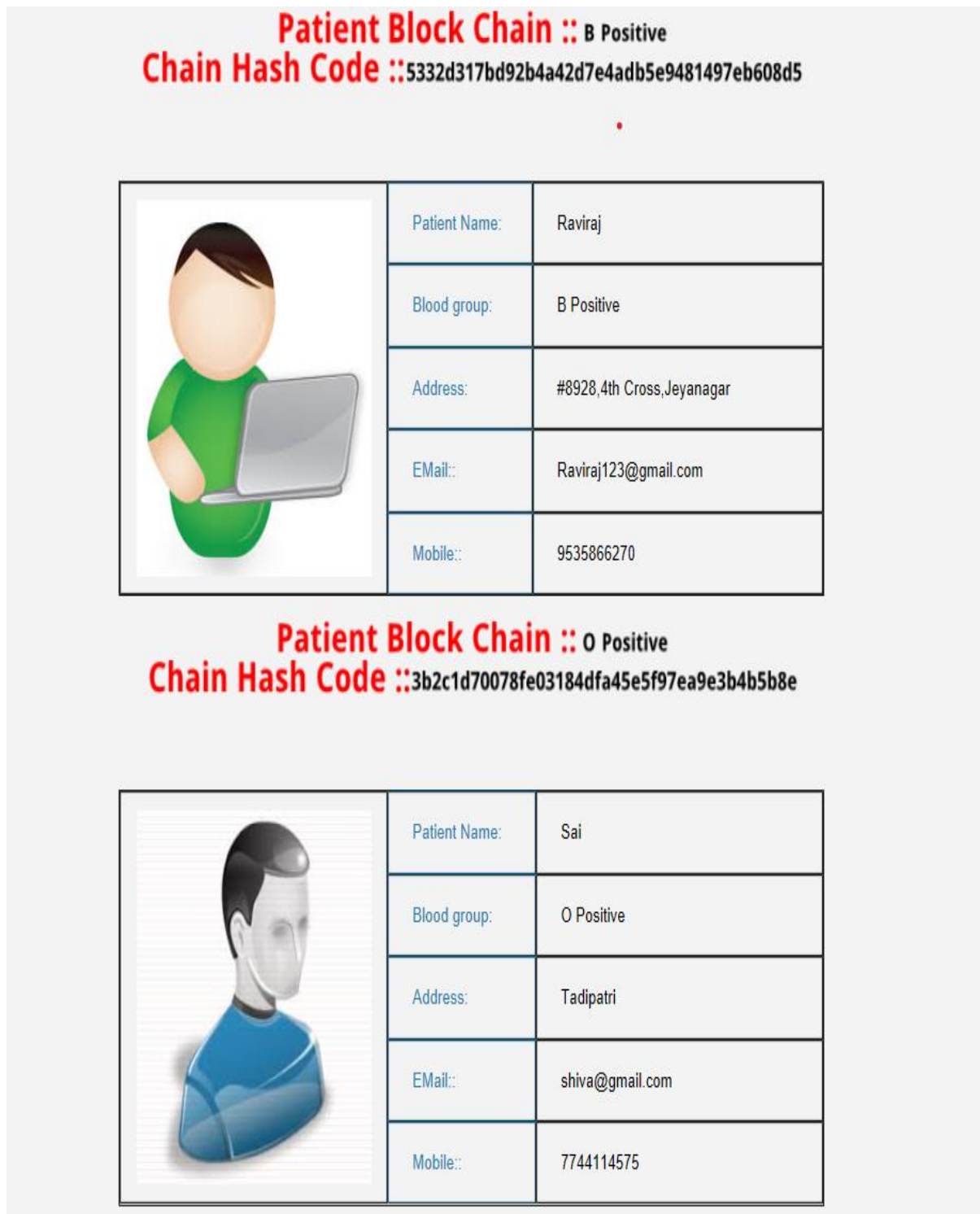


Fig 5.3.1.15: Patient details stored with Unique block chain hash code

Explanation: The above figure explains about the patient data stored in the database with unique hash code.

5.4 Summary

Implementation phase, the software deals with the translation of the design specifications into the source code. The ultimate of the implementation is to write the source code and the internal documentation so that it can be verified easily. The code and the documentation should be written in a manner that eases debugging, testing and modification. System flow charts, sample run packages, sample output. It is a part of the implementation and the validations of the project.