## INTRODUCTION

The project blood bank management system is known to be a pilot project that is designed for the blood bank to gather blood from various sources and distribute it to the needy people who have high Requirements for it .The Software is designed to handle the daily transaction of the blood bank and Search the details when required .It also helps to register the details of donors, blood collection details as well as blood issued reports .The software Application is designed in such a manner that it can suit the needs of all the blood bank requirements in the course of future.It will help us to find the Blood group with its most efficient time to take care of the blood and it is more easy to hand over the blood to the hospital to help people to get blood on time.This all thing is been stored and been seen in this blood bank management system.

**1.1 OBJECTIVES**

The primary objectives of the “ANANDA BLOOD LINE” project are:

* Efficient management of blood bank operations, including inventory management, donor records, and distribution processes.
* Ensuring data integrity and security by implementing proper access controls and data validation.
* Streamlining the process of managing donor information, including their blood type and contact details.
* Enabling administrators to easily manage and analyze data related to blood inventory and donor records.
* Implementing a system for updating and managing the contact information of the blood bank.

## 

**2. SYSTEM ANALYSIS**

**2.1 Existing System**

The system has basic functionalities for two types of users: Donors/Receivers and Hospitals/Clinics (Doctors). Donors/Receivers can add blood samples to their blood bank, request blood, check the status of their requests, and update their details. Similarly, Hospitals/Clinics can perform the same operations. It stores information about blood donors, different blood groups, and the validity of each blood unit. The system may include modules such as donor registration, blood group tracking, and inventory management.

**2.1.1 DISADVANTAGES**

* **Security vulnerabilities:** If the system is not properly secured, it could be susceptible to attacks such as SQL injection and cross-site scripting, which could compromise the confidentiality and integrity of donor and patient data.
* **Maintenance challenges:** Over time, the system may become outdated or require updates and bug fixes. If the development team is not experienced with PHP and MySQL, they may struggle to maintain and update the system.
* **Integration issues:** If the blood bank management system needs to integrate with other systems, such as electronic health records or laboratory information systems, there may be compatibility issues or technical challenges that need to be addressed.

**2.2 PROPOSED SYSTEM**

Blood Bank Management Software is designed and suitable for several Blood Bank either operating as indiviuals organizations or part of organizations covers all blood banking process from donors recruitment, donor management, mobile session component preparation, screening covering all test, blood stock inventory maintenance, patient registration, cross matching, patient issues etc.

**2.1.2 ADVANTAGES**

* **Efficient Data Management:** A PHP and MySQL-based system can help manage large amounts of data related to blood donors, recipients, and inventory, making it easier to track and access information quickly.
* **Improved Blood Inventory Management:** The system can provide real-time updates on the availability and expiration dates of blood units, reducing wastage and ensuring timely supply to those in need.
* **Data Analysis and Reporting:** With MySQL's data analysis capabilities, the system can generate detailed reports on various aspects of blood bank operations, aiding in decision-making and strategic planning.

**2.3 SYSTEM STUDY**

**2.3.1 FEASIBILITY ANALYSIS**

Feasibility is the study of impact, which happens in the organization by the development of a system. The impact can be either positive or negative. When the positives nominate the negatives, then the system is considered feasible. Here the feasibility study can be performed in two ways such as technical feasibility and Economical Feasibility.

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY

**2.3.2 TECHNICAL FEASIBILITY**

Since there will not be much difficulty in getting required resources for the development and maintaining the system as well. All the resources needed for the development of the software as well as the maintenance of the same is available in the organization here we are utilizing the resources which are available already.

**2.3.3 ECONOMICAL FEASIBILITY**

Development of this application is highly economically feasible. The organization needed not spend much money for the development of the system already available. The only thing is to be done is making an environment for the development with an effective supervision. If we are doing so, we can attain the maximum usability of the corresponding resources. Even after the development, the organization will not be in condition to invest more in the organization. Therefore, the system is economically feasible.

## 

## 3. SYSTEM SPECIFICATION

**3.1 HARDWARE REQUIREMENTS**

Processor **:** AMD PRO A4-3350B R4

Processor Speed **:** 2.00GHz Onwards

RAM **:** 4 GB

Hard Disk **:** 500 GB

Monitor **:** LG 23”

Network card **:** Any card can provide a 100mbps speed

**3.2 SOFTWARE REQUIREMENTS**

Operating system **:** Windows 7**/**8**/**10

Front End **:** PHP Framework

Back End **:** MySQL

UI Design **:** CSS,Bootstrap

Development Tool **:** XAMPP Server

**4. SOFTWARE DESCRIPTION**

**4.1 INTRODUCTION TO FRONTEND**

**HTML Overview**

Hyper Text Markup Language (HTML) is the standard markup language for creating web pages and web applications**.** With Cascading Style Sheets (CSS), and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a webserver or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document**.**

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects, such as interactive forms may be embedded into the rendered page**.** It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img> and <input /> introduce content into the page directly. Others such as <p>...</p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page**.**

HTML can embed programs written in a scripting language such as JavaScript which affect the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997**.**

**Development**

In 1980, physicist Tim Berners-Lee, a contractor at CERN, proposed and prototyped ENQUIRE, a system for CERN researchers to use and share documents. In 1989, Berners- Lee wrote a memo proposing an Internet-based hypertext system Berners-Lee specified HTML and wrote the browser and server software in late 1990. That year, Berners-Lee and CERN data systems engineer Robert Cailliau collaborated on a joint request for funding, but the project was not formally adopted by CERN. In his personal notes from 1990 he listed "some of the many areas in which hypertext is used" and put an encyclopedia first**.**

The first publicly available description of HTML was a document called "HTML Tags", first mentioned on the Internet by Tim Berners-Lee in late 1991.It describes 18 elements comprising the initial, relatively simple design of HTML. Except for the hyperlink tag, these were strongly influenced by SGMLguid, an in-house Standard Generalized Markup Language (SGML)-based documentation format at CERN. Eleven of these elements still exist in HTML 4**.**

HTML is a markup language that web browsers use to interpret and compose text, images, and other material into visual or audible web pages. Default characteristics for every item of HTML markup are defined in the browser, and these characteristics can be altered or enhanced by the web page designer's additional use of CSS. Many of the text elements are found in the 1988 ISO technical report TR 9537 Techniques for using SGML, which in turn covers the features of early text formatting languages such as that used by the RUNOFF command developed in the early 1960s for the CTSS (Compatible Time-Sharing System) operating system: these formatting commands were derived from the commands used by typesetters to manually format documents. However, the SGML concept of generalized markup is based on elements rather than merely print effects, with also the separation of structure and markup; HTML has been progressively moved in this direction with CSS**.**

Berners-Lee considered HTML to be an application of SGML. It was formally defined as such by the Internet Engineering Task Force (IETF) with the mid-1993 publication of the first proposal for an HTML specification: "Hypertext Markup Language (HTML)" Internet-Draft by Berners-Lee and Dan Connolly, which included an SGML Document Type Definition to define the grammar. The draft expired after six months, but was notable for its acknowledgment of the NCSA Mosaic browser's custom tag for embedding in-line images, reflecting the IETF's philosophy of basing standards on successful prototypes. Similarly, Dave Raggett's competing Internet-Draft, "HTML+ (Hypertext Markup Format)", from late 1993, suggested standardizing already-implemented features like tables and fill-out forms**.**

**Markup**

HTML markup consists of several key components, including those called tags (and their attributes), character-based data types, character references and entity references**.** HTML tags most commonly come in pairs like <h1> and </h1>, although some represent empty elements and so are unpaired, for example <img>**.** The first tag in such a pair is the start tag, and the second is the end tag (they are also called opening tags and closing tags)**.** Another important component is the HTML document type declaration, which triggers standards mode rendering. The following is an example of the classic Hello world program, a common test employed for comparing programming languages, scripting languages and markup languages**.**

This example is made using 9 lines of code:

<!DOCTYPE html>

**<html>**

**<head>**

**<title>**This is a title**</title>**

**</head>**

**<body>**

**<p>**Hello world!**</p>**

**</body>**

**</html>**

**Elements**

HTML documents imply a structure of nested HTML elements. These are indicated in the document by HTML tags, enclosed in angle brackets thus: <p>**.** In the simple, general case, the extent of an element is indicated by a pair of tags: a "start tag" <p> and "end tag" </p>. The text content of the element, if any, is placed between these tags. Tags may also enclose further tag markup between the start and end, including a mixture of tags and text. This indicates further (nested) elements, as children of the parent element. The start tag may also include attributes within the tag. These indicate other information, such as identifiers for sections within the document, identifiers used to bind style information to the presentation of the document, and for some tags such as the <img> used to embed images, the reference to the image resource.

**CSS Overview**

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS is designed primarily to enable the separation of document content from document presentation, including aspects such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate css file, and reduce complexity and repetition in the structural content. Separation of formatting and content makes it possible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. It can also display the web page differently depending on the screen size or viewing device. Readers can also specify a different style sheet, such as a CSS file stored on their own computer, to override the one the author specified. Changes to the graphic design of a document (or hundreds of documents) can be applied quickly and easily, by editing a few lines in the CSS file they use, rather than by changing markup in the documents.

The CSS specification describes a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called cascade, priorities (or weights) are calculated and assigned to rules, so that the results are predictable. The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/css is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.

**sources**

CSS information can be provided from various sources. These sources can be the web browser, the user and the author. The information from the author can be further classified into inline, media type, importance, selector specificity, rule order, inheritance and property definition. CSS style information can be in a separate document or it can be embedded into an HTML document. Multiple style sheets can be imported. Different styles can be applied depending on the output device being used; for example, the screen version can be quite different from the printed version, so that authors can tailor the presentation appropriately for each medium.

The style sheet with the highest priority controls the content display. Declarations not set in the highest priority source are passed on to a source of lower priority, such as the user agent style. This process is called cascading. One of the goals of CSS is to allow users greater control over presentation. Someone who finds red italic headings difficult to read may apply a different style sheet. Depending on the browser and the web site, a user may choose from various style sheets provided by the designers, or may remove all added styles and view the site using the browser's default styling, or may override just the red italic heading style without altering other attributes.

**Example**

Consider this HTML fragment**:**

<!DOCTYPE html>

**<html>**

**<head>**

<**meta** charset="utf-8">

**<style>**

**#xyz** {color: red; }

**</style>**

**</head>**

**<body>**

<**p** id="xyz" style="color: blue;"> To demonstrate specificity

**</p>**

**</body>**

**</html>**

In the above example, the declaration in the style attribute overrides the one in the <style> element because it has a higher specificity.

**Browser support**

Each web browser uses a layout engine to render web pages, and support for CSS functionality is not consistent between them. Because browsers do not parse CSS perfectly, multiple coding techniques have been developed to target specific browsers with workarounds (commonly known as CSS hacks or CSS filters). Adoption of new functionality in CSS can be hindered by lack of support in major browsers. For example, Internet Explorer was slow to add support for many CSS 3 features, which slowed adoption of those features and damaged the browser's reputation among developers.In order to ensure a consistent experience for their users, web developers often test their sites across multiple operating systems, browsers, and browser versions, increasing development time and complexity. Tools such as Browser Stack have been built to reduce the complexity of maintaining these environments.

In addition to these testing tools, many sites maintain lists of browser support for specific CSS properties, including Can I Use and the Mozilla Developer Network. Additionally, the CSS 3 defines feature queries, which provide an @supports directive that will allow developers to target browsers with support for certain functionality directly within their CSS. CSS that is not supported by older browsers can also sometimes be patched in using Javascript polyfills, which are pieces of Javascript code designed to make browsers behave consistently. These workarounds-and the need to support fallback functionality- can add complexity to development projects, and consequently, companies frequently define a list of browser versions that they will and will not support.

**Vertical Control limitations**

Though horizontal placement of elements was always generally easy to control, vertical placement was frequently unintuitive, convoluted, or outright impossible. Simple tasks, such as centering an element vertically or placing a footer no higher than bottom of the viewport required either complicated and unintuitive style rules, or simple but widely unsupported rules. The Flexible Box Module improved the situation considerably and vertical control is much more straightforward and supported in all of the modern browsers. Older browsers still have those issues, but most of those (mainly Internet Explorer 9 and below) are no longer supported by their vendors.

**Absence of expressions**

There was no standard ability to specify property values as simple expressions (such as margin-left: 10% 3em + 4px;). This would be useful in a variety of cases, such as calculating the size of columns subject to a constraint on the sum of all columns. Internet Explorer versions 5 to 7 support a proprietary expression() statement, with similar functionality. This proprietary expression() statement is no longer supported from Internet Explorer 8 onwards, except in compatibility modes. This decision was taken for "standards compliance, browser performance, and security reasons".However, a candidate recommendation with a calc() value to address this limitation has been published by the CSS WG and has since been supported in all of the modern browsers.

**PHP**

PHP is a server-side scripting language designed primarily for web development but also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994, the PHP reference implementation is now produced by The PHP Development Team.PHP originally stood for Personal Home Page, but it now stands for the recursive acronym PHP: Hypertext Preprocessor.

PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management systems and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications. The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge. The PHP language evolved without a written formal specification or standard until 2014, leaving the canonical PHP interpreter as a de facto standard. Since 2014 work has gone on to create a formal PHP specification.

**History**

PHP development began in 1995 when Rasmus Lerdorf wrote several Common Gateway Interface (CGI) programs in C,which he used to maintain his personal homepage. He extended them to work with web forms and to communicate with databases, and called this implementation "Personal Home Page/Forms Interpreter" or PHP/FI.

PHP/FI could help to build simple, dynamic web applications. To accelerate bug reporting and to improve the code, Lerdorf initially announced the release of PHP/FI as "Personal Home Page Tools (PHP Tools) version 1.0" on the Usenet discussion group comp.infosystems. www.authoring.cgi on June 8, 1995. This release already had the basic functionality that PHP has as of 2013. This included Perl-like variables, form handling, and the ability to embed HTML. The syntax resembled that of Perl but was simpler, more limited and less consistent.

Lerdorf did not intend the early PHP to become a new programming language, but it grew organically, with Lerdorf noting in retrospect: "I don't know how to stop it, there was never any intent to write a programming language. I have absolutely no idea how to write a programming language, I just kept adding the next logical step on the way. "A development team began to form and, after months of work and beta testing, officially released PHP/FI 2 in November 1997.

The fact that PHP lacked an original overall design but instead developed organically has led to inconsistent naming of functions and inconsistent ordering of their parameters. In some cases, the function names were chosen to match the lower-level libraries which PHP was "wrapping", while in some very early versions of PHP the length of the function names was used internally as a hash function, so names were chosen to improve the distribution of hash values.

**Syntax**

The following "Hello, World!" program is written in PHP code embedded in an HTML document:

<!DOCTYPE html>

**<html>**

**<head>**

**<title>**PHP Test**</title>**

**</head>**

**<body>**

<?**php echo** '<p>Hello World</p>'; ?>

**</body>**

**</html>**

The PHP interpreter only executes PHP code within its delimiters. Anything outside its delimiters is not processed by PHP, although non-PHP text is still subject to control structures described in PHP code. The most common delimiters are <?php to open and ?> to close PHP sections. The shortened form <? also exists. This short delimiter makes script files less portable, since support for them can be disabled in the local PHP configuration and it is therefore discouraged. However, there is no recommendation against the use of the echo short tag <?=Prior to PHP 5.4.0, this short syntax for echo() only works with the short\_open\_tag configuration setting enabled, while for PHP 5.4.0 and later it is always available. The purpose of all these delimiters is to separate PHP code from non-PHP content, such as JavaScript code or HTML markup. The first form of delimiters, <?php and ?>, in XHTML and other XML documents, creates correctly formed XML processing instructions. This means that the resulting mixture of PHP code and other markup in the server-side file is itself well-formed XML.

Variables are prefixed with a dollar symbol, and a type does not need to be specified in advance. PHP 5 introduced type hinting that allows functions to force their parameters to be objects of a specific class, arrays, interfaces or call back functions. However, before PHP 7.0, type hints could not be used with scalar types such as integer or string.

Unlike function and class names, variable names are case sensitive. Both double- quoted ("") and here doc strings provide the ability to interpolate a variable's value into the string. PHP treats newlines as whitespace in the manner of a free-form language, and statements are terminated by a semicolon.PHP has three types of comment syntax: /\*\*/ marks block and inline comments; // as well as # are used for one-line comments. The echo statement is one of several facilities PHP provides to output text, e.g., to a web browser. In terms of keywords and language syntax, PHP is similar to the C style syntax. if conditions, for and while loops, and function returns are similar in syntax to languages such as C, C++, C#, Java and Perl.

**4.2 INTRODUCTION TO BACKEND**

**MySQL**

MySQL is an open-source relational database management system (RDBMS).Its name is a combination of "My", the name of co-founder Michael Widenius' daughter, and "SQL", the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. For proprietary use, several paid editions are available, and offer additional functionality.

MySQL is a central component of the LAMP open-source web application software stack (and other "AMP" stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python". Applications that use the MySQL database include: TYPO3, MODx, Joomla, WordPress, phpBB, MyBB, and Drupal. MySQL is also used in many high-profile, large-scale websites, including Google (though not for searches), Facebook, Twitter, Flickr, and YouTube.

MySQL is written in C and C++. Its SQL parser is written in yacc, but it uses a home-brewed lexical analyzer. MySQL works on many system platforms, including AIX, BSDi, FreeBSD, HP-UX, eComStation, i5/OS, IRIX, Linux, OS X, Microsoft Windows, NetBSD, Novell NetWare, OpenBSD, OpenSolaris, OS/2 Warp, QNX, Oracle Solaris, Symbian, SunOS, SCO OpenServer, SCO UnixWare, Sanos and Tru64, A port of MySQL to OpenVMS also exists.

**History**

MySQL was created by a Swedish company, MySQL AB, founded by David Axmark, Allan Larsson and Michael "Monty" Widenius. Original development of MySQL by Widenius and Axmark began in 1994. The first version of MySQL appeared on 23 May 1995. It was initially created for personal usage from mSQL based on the low-level language ISAM, which the creators considered too slow and inflexible. They created a new SQL interface, while keeping the same API as mSQL. By keeping the API consistent with the mSQL system, many developers were able to use MySQL instead of the (proprietarily licensed) mSQL antecedent.

The MySQL server software itself and the client libraries use dual-licensing distribution. They are offered under GPL version 2, beginning from 28 June 2000 (which in 2009 has been extended with a FLOSS License Exception) or to use a proprietary license.Support can be obtained from the official manual. Free support additionally is available in different IRC channels and forums. Oracle offers paid support via its MySQL Enterprise products. They differ in the scope of services and in price. Additionally, a number of third party organisations exist to provide support and services, including MariaDB and Percona.

MySQL has received positive reviews, and reviewers noticed it "performs extremely well in the average case" and that the "developer interfaces are there, and the documentation (not to mention feedback in the real world via Web sites and the like) is very, very good". It has also been tested to be a "fast, stable and true multi-user, multi-threaded sql database server".

**Features**

MySQL is offered under two different editions: the open source MySQL Community Server and the proprietary Enterprise Server. MySQL Enterprise Server is differentiated by a series of proprietary extensions which install as server plugins, but otherwise shares the version numbering system and is built from the same code base.

* A broad subset of ANSI SQL 99, as well as extension
* Cross-platform support
* Stored procedures, using a procedural language that closely adheres to SQL/PSM
* Triggers
* Cursors
* Updatable views
* Online DDL when using the InnoDB Storage Engine.
* Information schema
* Performance Schema that collects and aggregates statistics about server execution and query performance for monitoring purposes.
* A set of SQL Mode options to control runtime behavior, including a strict mode to better adhere to SQL standards.
* X/Open XA distributed transaction processing (DTP) support; two phase commit as part of this, using the default InnoDB storage engine
* Transactions with save points when using the default InnoDB Storage Engine. The NDB Cluster Storage Engine also supports transactions.
* ACID compliance when using InnoDB and NDB Cluster Storage Engines
* SSL support
* Query caching
* Sub-SELECTs (i.e. nested SELECTs)
* Built-in replication support (i.e., master-master replication and master-slave replication) with one master per slave, many slaves per master.Multi-master replication is provided in MySQL Cluster, and multi-master support can be added to unclustered configurations using Galera Cluster
* Full-text indexing and searching
* Embedded database library
* Unicode support
* Partitioned tables with pruning of partitions in optimizer
* Shared-nothing clustering through MySQL Cluster
* Multiple storage engines, allowing one to choose the one that is most effective for each table in the application
* Native storage engines InnoDB, MyISAM, Merge, Memory (heap), Federated, Archive, CSV, Blackhole, NDB Cluster.

**5. SYSTEM DESIGN**

**5.1 MODULE DESCRIPTION**

* Donor management
* Inventory management
* Blood collection and processing
* Blood transfusion managemen
* Analytics and reporting

**Donor management**

This involves maintaining a database of donors, their eligibility criteria, and scheduling blood donation appointments. It also includes tracking donor information like blood type, donation history, and contact details.

**Inventory management**

The system monitors the blood inventory levels, tracks the expiration dates of blood products, and ensures the availability of appropriate blood products for transfusions.

**Blood collection and processing**

The system records the blood collection process, tracks the processing of blood products, and maintains a record of quality control checks.

**Blood transfusion management**

The system manages the blood transfusion process, records transfusion details, tracks the patient's blood type and compatibility, and generates reports.

**Analytics and reporting**

The system provides data analytics and reporting functions to track key performance indicators (KPIs) like donor recruitment, inventory levels, and blood usage.

**5.2 ARCHITECTURE DESIGN**

ANANDA BLOOD LINE SYSTEM

Administrator

Users

System management

Login / Logout

View Doners

Manage Blood and Doners

Contect Both Admin and Doners

Add Blood and Category management

Request

All data’s management

**5.3 DATA FLOW DIAGRAM**

Admin

ANANDA BLOOD LINE

Recipient

Donor

**Fig 5.3 DFD Level 0**

Recieve Blood Health Information

Blood info

Recieve Request

Blood Request

Confirm Donation **/** Request Blood claim Confirmation

Recipient

Manage Recipient Information

Manage Blood Bank Transactions

Manage Donation Information

Donor

Admin

**Fig 5.3 DFD Level 1**

Donation Confirmation

Donor info

Donation Confirmation Donor info

New Donor View Recipient

Applicable as Recipient View Donor

Request Confirmation Blood Request

Blood Request Request Confirmation

## 

## ER DAIGRAM

|  |  |
| --- | --- |
| Blood Donors | |
| String | Id |
| String | Name |
| String | Mobile no |
| String | Email |
| String | Pass |
| String | Gender |
| String | Age |
| String | Blood Group |
| String | Address |
| String | Department |

|  |  |
| --- | --- |
| Blood Group | |
| String | id |
| String | Blood Group |

## 

Admin

|  |  |
| --- | --- |
| Contect us | |
| String | Id |
| String | Address |
| String | Email |
| String | Contect no |

|  |  |
| --- | --- |
| Blood Requir | |
| String | id |
| String | Blood Donor id |
| String | Name |
| String | Email |
| String | Contect no |
| String | Blood requir |
| String | date |

## 

**5.4 TABLE DESIGN**

**Table of Admin**

|  |  |  |
| --- | --- | --- |
| Field | Type | Other |
| ID | int(10) |  |
| AdminName | varchar(120) | NULLABLE |
| UserName | varchar(120) | NULLABLE |
| MobileNumber | bigint(10) | NULLABLE |
| Email | varchar(200) | NULLABLE |
| Password | varchar(200) | NULLABLE |
| AdminRegdate | timestamp | NULLABLE |

## Table of Blooddonors Table

|  |  |  |
| --- | --- | --- |
| Field | Type | Other |
| id | int(11) |  |
| FullName | varchar(100) | NULLABLE |
| MobileNumber | char(11) | NULLABLE |
| EmailId | varchar(100) | NULLABLE |
| Gender | varchar(20) | NULLABLE |
| Age | int(11) | NULLABLE |
| BloodGroup | varchar(20) | NULLABLE |
| Address | varchar(255) | NULLABLE |
| Message | mediumtext | NULLABLE |
| PostingDate | timestamp |  |
| status | int(1) | NULLABLE |
| Password | varchar(250) | NULLABLE |

## Table of Bloodgroup

|  |  |  |
| --- | --- | --- |
| Field | Type | Other |
| id | int(11) |  |
| BloodGroup | varchar(20) | NULLABLE |
| PostingDate | timestamp |  |

## Tablo of Bloodrequirer

|  |  |  |
| --- | --- | --- |
| Field | Type | Other |
| ID | int(10) |  |
| BloodDonarID | int(10) | NULLABLE |
| name | varchar(250) | NULLABLE |
| EmailId | varchar(250) | NULLABLE |
| ContactNumber | bigint(10) | NULLABLE |
| BloodRequirefor | varchar(250) | NULLABLE |
| Message | mediumtext | NULLABLE |
| ApplyDate | timestamp | NULLABLE |

## Table of Contactusinfo

|  |  |  |
| --- | --- | --- |
| Field | Type | Other |
| id | int(11) |  |
| Address | tinytext | NULLABLE |
| EmailId | varchar(255) | NULLABLE |
| ContactNo | char(11) | NULLABLE |

## Table of Contactusquery

|  |  |  |
| --- | --- | --- |
| Field | Type | Other |
| id | int(11) |  |
| name | varchar(100) | NULLABLE |
| EmailId | varchar(120) | NULLABLE |
| ContactNumber | char(11) | NULLABLE |
| Message | longtext | NULLABLE |
| PostingDate | timestamp |  |
| status | int(11) | NULLABLE |

## Table of Pages

|  |  |  |
| --- | --- | --- |
| Field | Type | Other |
| id | int(11) |  |
| PageName | varchar(255) | NULLABLE |
| type | varchar(255) |  |
| detail | longtext |  |

1. **SYSTEM IMPLEMENTAION**

**SYSTEM IMPLEMENTAION**

Implementation is the stage in the project where the theoretical design is turned into a working system and is giving confidence on the new system for the users that it will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the change over, an evaluation of change over methods. Apart from planning major task of preparing the implementation are education and training of users. The implementation process begins with preparing a plan for the implementation of the system. According to this plan, the activities are to be carried out, discussions made regarding the equipment and resources and the additional equipment has to be acquired to implement the new system.

In network backup system no additional resources are needed. Implementation is the final and the most important phase. The most critical stage in achieving a successful new system is giving the users confidence that the new system will work and be effective. The system can be implemented only after thorough testing is done and if it is found to be working according to the specification. This method also offers the greatest security since the old system can take over if the errors are found or inability to handle certain type of transactions while using the new system.

As the part of system testing we execute the program with the intent of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied. The ultimate aim is quality assurance. Tests are carried out and the results are compared with the expected document. In the case of erroneous results, debugging is done. Using detailed testing strategies a test plan is carried out on each module. The various tests performed are unit testing, integration testing and user acceptance testing.

**7. SYSTEM TESTING**

**7.1. UNIT TESTING**

The software units in the system are modules and routines that are assembled and integrated to perform a specific function. As a part of unit testing we executed the program for individual modules independently. This enables, to detect errors in coding and logic that are contained within each of the three modules. This testing includes entering data that is filling forms and ascertaining if the value matches to the type and entered into the database. The various controls are tested to ensure that each performs its action as required.

Unit testing verification efforts on the smallest unit of software design, module. This is known as "module testing". After testing each every field in the modules, the modules of the project is tested separately. Unit testing focuses verification efforts on the smallest unit of software design and field. For example, username and password are entered in correct manner and checked. While filling the details in the register form certain fields are left as empty and checked. The submit button successfully stores the data in the databases. This is done for each and every module individually.

"Unit testing" is a software testing method testing method by which individual unit of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedure, are tested to determine whether they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are short fragments created by programmers or occasionally by white box testers during the development process. It is also known as component testing.

Ideally, each test case is independent from the other. Substitutes such as method stubs, mock object, fakes and test harness can be used to assist testing a module in isolation, unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended. The primary goal of unit testing is to take the smallest piece of testable software in the application, isolation it from the reminder of the code, and determine whether it behaves exactly as you expect. Each unit tested separately before integrating them

into modules to test the interfaces between modules. Unit testing has proven its value in that a large percentage of defects are identified during its use.

The most common approach to unit testing requires drivers and stubs to be written, the driver simulates a calling unit and the stub simulates a called unit. The investment of developer in this activity sometimes results in demoting unit testing to a lower level of priority and that is almost always a mistake. Even though the drivers and stubs cost time and money, unit testing provides some undeniable advantages. It allows for automation of the testing process, reduce difficulties of discovering errors contained in more complex pieces of the application, and test coverage is often enhanced because attention is given to each unit. The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. A unit test provides a strict, written contract that the pieces of code must satisfy. As a result, it affords several benefits.

**Finds Problem Early:**

Unit testing finds problems early in the development cycle. In test-driven development (TDD), which is frequently used in both extreme programming and scrum, unit tests are created before the code itself is written. When the tests pass, that code is considered complete. The same unit tests are run against that function frequently as the larger code base is developed either as the code is changed or via an automated process with the build. If the unit tests fail, it is considered to be a bug either in the changed code or the test. Since the unit tests alert the development team of the problem before handling the code off to testers or clients, it is still early in the development process.

**Facilitates Change:**

Unit testing allows the programmer to re factor code at a later date, and make sure the module still works correctly. The procedure is to write test cases for all functions and methods so that whenever a change a fault, it can be quickly identified. Readily available unit tests make it easy for the programmer to check whether a piece of code is still working properly. In continuous unit testing environments, through the inherent practice of sustained maintenance, unit tests will continue to accurately reflect the intended use of the executable and code in the face of any change. Depending upon established development practices and unit test coverage, up-to-the-second accuracy can be maintained.

Unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data. Usage procedures, and operating procedures, are tested to determine whether they fit for use. Intuitively one can view a unit as the smallest testable part of an application. Unit testing is way of testing software components. The "unit" is the thing being tested. You can do both black and white box testing with unit tests; the concept is orthogonal to white/black-box testing.

**White Box Testing:**

Structured testing is known as "white box testing" or "glass box testing" program errors can be classified as missing path errors, computational errors and domain errors. "white-box testing" can be applied at the unit, integration and system level of the software testing process. Although traditional tester's tenders to think of white box testing as being done at the unit level, it is used integration more frequently today. It can test paths within a unit, paths between units during integration, and between subsystems during a system-level test. Though this of design can uncover many errors or problems, it has the potential to miss unimplemented parts of the specifications or missing requirements.

**Black Box Testing:**

Stress tests drive the system to its limits. They are design to internationally break the unit. Structure tests verify logical execution paths. Functional, performance and stress tests are collectively known as "black box testing"

"black-box" testing is a method of software testing that examines the functionally of an application without peering into its internal structure or workings.

This method of test can be applied to virtually every level of software testing:

* Unit, integration, system and acceptance it typically comprises most if not all higher level testing, but can also dominate unit testing as well
* In black box testing, you don't care how the internals of the thing being tested work. You invoke the exposed API and check the result; you don't care what to the thing being tested did to give you the result.
* In white box testing, you don't care how the internal of the thing being tested work. So instead of just checking the output of your thing, you might check that internal variables to the thing being tested are correct.
* Unit testing is a way of testing software components. The "unit" is the thing being tested. You can do both black and white box testing with unit tests; the concept is orthogonal to white/black-box testing.

**7.2. INTEGRATION TESTING**

Integration testing is done to take unit-tested modules and build a program structure that has dictated by design. All the modules were integrated after the completion of unit test. The modules are integrated by moving downward through, beginning with the main module.

After the successful integration of the modules, the system was found to be running with no error, here, on clicking the submit button the detail are updated on the data base, also an email with the user name and password is sent to the user's mail ID. Similarly transaction ID is generated, stored in the data base and it sent to the user's mail ID. Thus all the modules are integrated and are tested successfully. Integration testing is a phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and after validation testing. Integration takes as its input modules that have been unit tested, groups them in larger aggregates.

**Big Bang:**

In this approach, all or most of the developed modules are coupled together to form a complete software system or major part of the system and then used for integration testing. The big bang method is very effective for saving time in the integration testing process. However if the test cases and their results are not recorded properly, the entire integration process will be more complicated and may prevent the testing team from achieving the goal of integration testing.

A type of big bang integration testing is called usage model testing. Usage model testing can be used in both software and hardware integration testing. The basic behind this type of integration testing is to run user-like workloads in integrated user-like environment. In doing the testing in this manner, the environment is proofed, while the individual components are proofed indirectly through their use. The goal of the strategy is to avoid redoing the testing done by the developers, and instead flesh-out problems caused by the interaction of the environment. For integration testing.

Usage model testing can be more efficient and provides better test coverage than traditional focused functional testing. To more efficient and accurate, care must be used in defining the user-like workloads for creating realistic environment will work as expected for the target customers.

**Simplifies Integration:**

Integration testing may reduce uncertainly in the units themselves and can be used in a bottom-up testing style approach. By testing the parts of a program first and then testing the sum of its parts, integration testing becomes much easier. An elaborate In integration hierarchy of unit tests does not equal integration testing. Integration with peripheral units should be included in integration tests, but not in unit test integration testing typically still relies heavily on humans testing manually; high-level or global-scope testing can be difficult to automate, such that manual testing often appears faster and cheaper.

**Top-Down and Bottom-Up:**

**Bottom-Up Testing:**

"Bottom up testing" is approach to integrated testing where the lowest level components are tested first, then used to facilitate the testing of higher level components. The process is repeated until the component at the top of the hierarchy is tested.

All the bottom or low-level modules, procedures or functions are integrated and then tested. After the integration testing of lower integrated modules, the next level of modules will be formed and can be used for integration testing This approach is helpful only when all or most of the modules of the same development level are ready. This method also helps to determine the levels of software developed and makes it easier to report testing progress in the form of a percentage.

**Top-Down Testing:**

Top down testing is an approach to integrated testing where the top integrated modules are tested and the branch of the module is tested step by step until the end of the related module.

**Sandwich Testing:**

Sandwich testing is an approach to combine top down testing with bottom up testing The main advantage of the bottom-up approach is that bugs are more easily found. With top- down, it is easier to find a missing branch link.

**7.3. USER ACCEPTANCE TESTING**

Validation testing provides the assurance that software needs all the functional behavioural and performance requirements. Validation testing can be defined as the software functions in a manner that is expected by the user. This testing verifies that all elements combine properly and that overall system function and performance is achieved. After the integration of the modules, the validation test was carried out over by the system. It was found that all the modules work well together and meet the overall system function and performance.

Validation testing provides a sort of living documentation of the system. Develops looking to learn what functionally provided by a validation, and how to use it, can look at the unit tests to gain a basic understanding of the validation interface. Validation are independent that are used together for checking that a product, services, or system meets requirement and specification and that it fulfils its ISO 9000. The words "verification" and "validation" are sometimes preceded with "independent", indicating that the verification and validation is to be performed by a disinterested third party. "Independent verification and validation" can be abbreviated as "IV&V".

"Validation The assurance that a product, service, or system meets the needs of the customer and other identified stakeholder. It often involves acceptance and suitability with external customer. Contrast with verification". The evaluation of whether or not a product, service or system complies with a regulation, requirement, specification, or imposed condition. It is often as internal process. Contrast with validation

"Verification is intended to check that a product service, or system (or portion thereof, or set thereof) meets a set of design specification. In the development phase, verification procedures involve performing special tests to model or simulates a portion, or the entirely, service or system, then performing a review or analysis of the modelling result. In the post-development phase, verification procedures involve regularly repeating tests devised specifically to ensure that the product, service, or system continues to meet the initial design requirement, specification, and regulation as time progresses it is a process that is used to evaluate whether a product, service or system complies with regulations, specification, or condition imposed at the start of a development phase. Verification can be in development, scale-up, or production. This is often a internal process.

**8. CONCLUSION**

The Blood Bank Management System project was developed to streamline the process of blood donation and management. The main aim was to demonstrate the use of CRUD (Create, Read, Update, Delete) operations using MySQL. The system starts by registering the doctor's details, followed by adding blood donor details with their blood group, quantity, age, and other medical information. When a patient needs blood, doctors can easily check the available blood group and provide it to the patient. The system ensures that the donated blood is managed properly, and there are no negative effects on the receiver. The system also allows doctors and blood inventory management staff to organize and supply the blood needed efficiently. Overall, the project successfully addressed real-time issues related to blood donation and management.

**FUTURE ENHANCEMENTS:**

The web system some potential future enhancements for an Ananda Blood Line:

* **GPS :** We would like to gather more information regarding the contact persons in other cities as well as villages and will provide much more services for the people and help everyone with humanity. GPS is the best way to provide actual location. In future, we would like to add the location tracking system through GPS which will help us to upgrade the donor searching method.
* **Mobile Apps:** Now a days people are comfortable with various application those are easy to use and portable. We would like to make a portable and modified BBMS version based on android OS and IOS also in future.
* **SMS System:** SMS system is important because donor notification update, receiver information send for the remainder. It is very helpful to save time and avoid a critical situation.

1. **BIBLIOGRAPHY**

**Book Reference:**

1. Chris Lea, Wankyu Choi, Allan Kent, Ganesh Prasad, Chris Ullma, "Beginning PHP4", 1st edition, Wrox Press Inc, October, 2000
2. Roger S. Pressman, "Software Engineering: A Practioner's Approach", 6th Edition, Tata McGraw Hill, New Delhi, 2005.
3. Abraham Silberschatz, Hendry F.Korth, S. Sudharsan, "Database Systems Concepts", 4th Edition, Tata McGraw Hill International, Singapore, 2002.

**Website Reference:**

* www.w3schools.com/php/default.asp
* www.php.net/manual/en/tutorial.php
* www.phptpoint.com

1. **APPENDIX**
2. **SAMPLE SOURCE CODE**

**HOME PAGE**

<?php

error\_reporting(0);

include('includes/config.php');

?>

<!DOCTYPE html>

<html lang="zxx">

<head>

<title>Blood Bank Donar Management System | Home Page</title>

<script>

addEventListener("load", function () {

setTimeout(hideURLbar, 0);

}, false);

function hideURLbar() {

window.scrollTo(0, 1);

}

</script>

<!--// Meta tag Keywords -->

<!-- jQuery library -->

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"></script>

<!-- Popper JS library -->

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"></script>

<script type="text/javascript" src='js/bootstrap.js'></script>

<script src="Scripts/umd/popper.min.js"></script>

<!-- Custom-Files -->

<link rel="stylesheet" href="css/bootstrap.css">

<!-- Bootstrap-Core-CSS -->

<link rel="stylesheet" href="css/style.css" type="text/css" media="all" />

<!-- Style-CSS -->

<link rel="stylesheet" href="css/fontawesome-all.css">

<!-- Font-Awesome-Icons-CSS -->

<!-- //Custom-Files -->

<!-- Web-Fonts -->

<link href="//fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,700,700i,800,800i&amp;subset=cyrillic,cyrillic-ext,greek,greek-ext,latin-ext,vietnamese"

rel="stylesheet">

<link href="//fonts.googleapis.com/css?family=Roboto+Condensed:300,300i,400,400i,700,700i&amp;subset=cyrillic,cyrillic-ext,greek,greek-ext,latin-ext,vietnamese"

rel="stylesheet">

<!-- //Web-Fonts -->

</head>

<body>

<?php include('includes/header.php');?>

<!-- banner -->

<div class="slider">

<div class="callbacks\_container">

<ul class="rslides callbacks callbacks1" id="slider4">

<li>

<div class="banner-top1">

<div class="banner-info\_agile\_w3ls">

<div class="container">

<h3>

<span>Every drop counts</span>

donate blood and make a difference

</h3>

</div>

</div>

</div>

</li>

<li>

<!-- <div class="banner-top2">

<div class="banner-info\_agile\_w3ls">

<div class="container">

<h3>Your donation may be a<span>drop in the bucket</span> , but to someone in need, it could mean the world

</h3>

</div>

</div>

</div> -->

</li>

<li>

<!-- <div class="banner-top3">

<div class="banner-info\_agile\_w3ls">

<div class="container">

</div>

</div>

</div> -->

</li>

</ul>

</div>

</div>

<!-- //banner -->

<div class="clearfix"></div>

<!-- banner bottom -->

<div class="banner-bottom py-5">

<div class="d-flex container py-xl-3 py-lg-3">

<div class="banner-left-bottom-w3ls offset-lg-2 offset-md-1">

<h3 class="text-white my-3">High professional doctors</h3>

<p>all specialists have extensive practical experience and regularly training courses in educational centers of the

world</p>

</div>

<div class="button">

<a href="about.php" class="w3ls-button-agile" style="background-color:white">Read More

<i class="fas fa-hand-point-right"></i>

</a>

</div>

</div>

</div>

<!-- //banner bottom -->

<!-- blog -->

<div class="blog-w3ls py-5" id="blog">

<div class="container py-xl-5 py-lg-3">

<div class="w3ls-titles text-center mb-5">

<h3 class="title text-white">Some of the Donar</h3>

<span>

<i class="fas fa-user-md text-white"></i>

</span>

</div>

<div class="row package-grids mt-5">

<?php

$status=1;

$sql = "SELECT \* from tblblooddonars where status=:status order by rand() limit 6";

$query = $dbh -> prepare($sql);

$query->bindParam(':status',$status,PDO::PARAM\_STR);

$query->execute();

$results=$query->fetchAll(PDO::FETCH\_OBJ);

$cnt=1;

if($query->rowCount() > 0)

{

foreach($results as $result)

{ ?>

<div class="col-md-4 pricing" style="margin-top:2%;">

<div class="price-top">

<img src="images/blood-donor.jpg" alt="" class="img-fluid" />

<h3><?php echo htmlentities($result->FullName);?>

</h3>

</div>

<div class="price-bottom p-4">

<h4 class="text-dark mb-3">Gender: <?php echo htmlentities($result->Gender);?></h4>

<p class="card-text"><b>Blood Group :</b> <?php echo htmlentities($result->BloodGroup);?></p>

<a class="btn btn-primary" style="color:#fff" href="contact-blood.php?cid=<?php echo $result->id;?>">Request</a>

</div>

</div><?php }} ?>

</div>

</div>

</div>

<!-- //blog -->

<!-- treatments -->

<div class="screen-w3ls py-5">

<div class="container py-xl-5 py-lg-3">

<div class="w3ls-titles text-center mb-5">

<h3 class="title">BLOOD GROUPS</h3>

<span>

<i class="fas fa-user-md"></i>

</span>

<p class="mt-2">blood group of any human being will mainly fall in any one of the following groups..</p>

</div>

<div class="row">

<div class="col-lg-6">

<ul>

<li>A positive or A negative</li>

<li>B positive or B negative</li>

<li>O positive or O negative</li>

<li>AB positive or AB negative.</li>

</ul>

<p>A healthy diet helps ensure a successful blood donation, and also makes you feel better! Check out the following recommended foods to eat prior to your donation.</p>

</div>

<div class="col-lg-6">

<img class="img-fluid rounded" src="images/bdn.png" alt="">

</div>

</div>

<div class="row mb-4">

<div class="col-md-8">

<h4 style="padding-top: 30px;">UNIVERSAL DONORS AND RECIPIENTS</h4>

<p>

The most common blood type is O, followed by type A.

Type O individuals are often called "universal donors" since their blood can be transfused into persons with any blood type. Those with type AB blood are called "universal recipients" because they can receive blood of any type.</p>

</div>

<div class="col-md-4" style="padding-top: 30px;">

<a class="btn btn-lg btn-secondary btn-block login-button ml-lg-5 mt-lg-0 mt-4 mb-lg-0 mb-3" data-toggle="modal" data-target="#exampleModalCenter1" href="donor-list.php" data-toggle="modal" data-target="#exampleModalCenter1"> Become a Donar</a>

</div>

</div>

</div>

</div>

<!-- //treatments -->

<!-- footer -->

<?php include('includes/footer.php');?>

<!-- Js files -->

<!-- JavaScript -->

<script src="js/jquery-2.2.3.min.js"></script>

<!-- Default-JavaScript-File -->

<!-- banner slider -->

<script src="js/responsiveslides.min.js"></script>

<script>

$(function () {

$("#slider4").responsiveSlides({

auto: true,

pager: true,

nav: true,

speed: 1000,

namespace: "callbacks",

before: function () {

$('.events').append("<li>before event fired.</li>");

},

after: function () {

$('.events').append("<li>after event fired.</li>");

}

});

});

</script>

<!-- //banner slider -->

<!-- fixed navigation -->

<script src="js/fixed-nav.js"></script>

<!-- //fixed navigation -->

<!-- smooth scrolling -->

<script src="js/SmoothScroll.min.js"></script>

<!-- move-top -->

<script src="js/move-top.js"></script>

<!-- easing -->

<script src="js/easing.js"></script>

<!-- necessary snippets for few javascript files -->

<script src="js/medic.js"></script>

<script src="js/bootstrap.js"></script>

<!-- Necessary-JavaScript-File-For-Bootstrap -->

<!-- //Js files -->

</body>

</html>

**SIGN UP PAGE**

<?php

session\_start();

error\_reporting(0);

include('includes/config.php');

if(isset($\_POST['submit']))

{

$fullname=$\_POST['fullname'];

$mobile=$\_POST['mobileno'];

$email=$\_POST['emailid'];

$age=$\_POST['age'];

$gender=$\_POST['gender'];

$blodgroup=$\_POST['bloodgroup'];

$address=$\_POST['address'];

$message=$\_POST['message'];

$status=1;

$password=md5($\_POST['password']);

$ret="select EmailId from tblblooddonars where EmailId=:email";

$query= $dbh -> prepare($ret);

$query-> bindParam(':email', $email, PDO::PARAM\_STR);

$query-> execute();

$results = $query -> fetchAll(PDO::FETCH\_OBJ);

if($query -> rowCount() == 0)

{

$sql="INSERT INTO tblblooddonars(FullName,MobileNumber,EmailId,Age,Gender,BloodGroup,Address,Message,status,Password) VALUES(:fullname,:mobile,:email,:age,:gender,:blodgroup,:address,:message,:status,:password)";

$query = $dbh->prepare($sql);

$query->bindParam(':fullname',$fullname,PDO::PARAM\_STR);

$query->bindParam(':mobile',$mobile,PDO::PARAM\_STR);

$query->bindParam(':email',$email,PDO::PARAM\_STR);

$query->bindParam(':age',$age,PDO::PARAM\_STR);

$query->bindParam(':gender',$gender,PDO::PARAM\_STR);

$query->bindParam(':blodgroup',$blodgroup,PDO::PARAM\_STR);

$query->bindParam(':address',$address,PDO::PARAM\_STR);

$query->bindParam(':message',$message,PDO::PARAM\_STR);

$query->bindParam(':status',$status,PDO::PARAM\_STR);

$query->bindParam(':password',$password,PDO::PARAM\_STR);

$query->execute();

$lastInsertId = $dbh->lastInsertId();

if($lastInsertId)

{

echo "<script>alert('You have signup Scuccessfully');</script>";

}

else

{

echo "<script>alert('Something went wrong.Please try again');</script>";

}

}

else

{

echo "<script>alert('Email-id already exist. Please try again');</script>";

}

}

?>

<!DOCTYPE html>

<html lang="zxx">

<head>

<title>Blood Bank Donar Management System | About Us </title>

<!-- Meta tag Keywords -->

<script>

addEventListener("load", function () {

setTimeout(hideURLbar, 0);

}, false);

function hideURLbar() {

window.scrollTo(0, 1);

}

</script>

<!--// Meta tag Keywords -->

<!-- Custom-Files -->

<link rel="stylesheet" href="css/bootstrap.css">

<!-- Bootstrap-Core-CSS -->

<link rel="stylesheet" href="css/style.css" type="text/css" media="all" />

<!-- Style-CSS -->

<link rel="stylesheet" href="css/fontawesome-all.css">

<!-- Font-Awesome-Icons-CSS -->

<!-- //Custom-Files -->

<!-- Web-Fonts -->

<link href="//fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,700,700i,800,800i&amp;subset=cyrillic,cyrillic-ext,greek,greek-ext,latin-ext,vietnamese"

rel="stylesheet">

<link href="//fonts.googleapis.com/css?family=Roboto+Condensed:300,300i,400,400i,700,700i&amp;subset=cyrillic,cyrillic-ext,greek,greek-ext,latin-ext,vietnamese"

rel="stylesheet">

<!-- //Web-Fonts -->

</head>

<body>

<?php include('includes/header.php');?>

<!-- banner 2 -->

<div class="inner-banner-w3ls">

<div class="container">

</div>

<!-- //banner 2 -->

</div>

<!-- page details -->

<!-- <div class="breadcrumb-agile">

<div aria-label="breadcrumb">

<ol class="breadcrumb">

<li class="breadcrumb-item">

<a href="index.php">Home</a>

</li>

<li class="breadcrumb-item active" aria-current="page">Signup</li>

</ol>

</div>

</div> -->

<!-- //page details -->

<!-- about -->

<section class="about py-5">

<div class="container py-xl-5 py-lg-3">

<div class="login px-4 mx-auto mw-100">

<h5 class="text-center mb-4">Register Now</h5>

<form action="#" method="post" name="signup" onsubmit="return checkpass();">

<div class="form-group">

<label>Full Name</label>

<input type="text" class="form-control" name="fullname" id="fullname" placeholder="Full Name">

</div>

<div class="form-group">

<label>Mobile Number</label>

<input type="text" class="form-control" name="mobileno" id="mobileno" required="true" placeholder="Mobile Number" maxlength="10" pattern="[0-9]+">

</div>

<div class="form-group">

<label class="mb-2">Email Id</label>

<input type="email" name="emailid" class="form-control" placeholder="Email Id">

</div>

<div class="form-group">

<label class="mb-2">Age</label>

<input type="text" class="form-control" name="age" id="age" placeholder="Age" required="">

</div>

<div class="form-group">

<label class="mb-2">Gender</label>

<select name="gender" class="form-control" required>

<option value="">Select</option>

<option value="Male">Male</option>

<option value="Female">Female</option>

</select>

</div>

<div class="form-group">

<label class="mb-2">Blood Group</label>

<select name="bloodgroup" class="form-control" required>

<?php $sql = "SELECT \* from tblbloodgroup ";

$query = $dbh -> prepare($sql);

$query->execute();

$results=$query->fetchAll(PDO::FETCH\_OBJ);

$cnt=1;

if($query->rowCount() > 0)

{

foreach($results as $result)

{ ?>

<option value="<?php echo htmlentities($result->BloodGroup);?>"><?php echo htmlentities($result->BloodGroup);?></option>

<?php }} ?>

</select>

</div>

<div class="form-group">

<label>Address</label>

<input type="text" class="form-control" name="address" id="address" required="true" placeholder="Address">

</div>

<div class="form-group">

<label>Message</label>

<textarea class="form-control" name="message" required> </textarea>

</div>

<div class="form-group">

<label>Password</label>

<input type="password" class="form-control" name="password" id="password" required="">

</div>

<button type="submit" class="btn btn-primary submit mb-4" name="submit">Register</button>

<p class="account-w3ls text-center pb-4" style="color:#000">

Already Registered?

<a href="login.php" >Signin now</a>

</p>

</form>

</div>

</div>

</section>

<!-- //about -->

<?php include('includes/footer.php');?>

<!-- Js files -->

<!-- JavaScript -->

<script src="js/jquery-2.2.3.min.js"></script>

<!-- Default-JavaScript-File -->

<!-- banner slider -->

<script src="js/responsiveslides.min.js"></script>

<script>

$(function () {

$("#slider4").responsiveSlides({

auto: true,

pager: true,

nav: true,

speed: 1000,

namespace: "callbacks",

before: function () {

$('.events').append("<li>before event fired.</li>");

},

after: function () {

$('.events').append("<li>after event fired.</li>");

}

});

});

</script>

<!-- //banner slider -->

<!-- fixed navigation -->

<script src="js/fixed-nav.js"></script>

<!-- //fixed navigation -->

<!-- smooth scrolling -->

<script src="js/SmoothScroll.min.js"></script>

<!-- move-top -->

<script src="js/move-top.js"></script>

<!-- easing -->

<script src="js/easing.js"></script>

<!-- necessary snippets for few javascript files -->

<script src="js/medic.js"></script>

<script src="js/bootstrap.js"></script>

<!-- Necessary-JavaScript-File-For-Bootstrap -->

<!-- //Js files -->

</body>

</html>

**LOGIN PAGE**

<?php session\_start();

error\_reporting(0);

include('includes/config.php');

if(isset($\_POST['login']))

{

$email=$\_POST['email'];

$password=md5($\_POST['password']);

$sql ="SELECT id FROM tblblooddonars WHERE EmailId=:email and Password=:password";

$query=$dbh->prepare($sql);

$query->bindParam(':email',$email,PDO::PARAM\_STR);

$query-> bindParam(':password', $password, PDO::PARAM\_STR);

$query-> execute();

$results=$query->fetchAll(PDO::FETCH\_OBJ);

if($query->rowCount() > 0)

{

foreach ($results as $result) {

$\_SESSION['bbdmsdid']=$result->id;

echo($\_SESSION['bbdmsdid']);

}

$\_SESSION['login']=$\_POST['email'];

echo "<script type='text/javascript'> document.location ='index.php'; </script>";

} else{

echo "<script>alert('Invalid Details');</script>";

}

}

?>

<!DOCTYPE html>

<html lang="zxx">

<head>

<title>Blood Bank Donar Management System | About Us </title>

<!-- Meta tag Keywords -->

<script>

addEventListener("load", function () {

setTimeout(hideURLbar, 0);

}, false);

function hideURLbar() {

window.scrollTo(0, 1);

}

</script>

<!--// Meta tag Keywords -->

<!-- Custom-Files -->

<link rel="stylesheet" href="css/bootstrap.css">

<!-- Bootstrap-Core-CSS -->

<link rel="stylesheet" href="css/style.css" type="text/css" media="all" />

<!-- Style-CSS -->

<link rel="stylesheet" href="css/fontawesome-all.css">

<script src="jquery/jquery.js"></script>

<script type="text/javascript" src='js/bootstrap.min.js'></script>

<link rel="stylesheet" href="css/bootstrap.css" />

<!-- Font-Awesome-Icons-CSS -->

<!-- //Custom-Files -->

<!-- Web-Fonts -->

<link href="//fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,700,700i,800,800i&amp;subset=cyrillic,cyrillic-ext,greek,greek-ext,latin-ext,vietnamese"

rel="stylesheet">

<link href="//fonts.googleapis.com/css?family=Roboto+Condensed:300,300i,400,400i,700,700i&amp;subset=cyrillic,cyrillic-ext,greek,greek-ext,latin-ext,vietnamese"

rel="stylesheet">

<!-- //Web-Fonts -->

</head>

<body>

<?php include('includes/header.php');?>

<!-- banner 2 -->

<div class="inner-banner-w3ls">

<div class="container">

</div>

<!-- //banner 2 -->

</div>

<!-- page details -->

<div class="breadcrumb-agile">

<div aria-label="breadcrumb">

<ol class="breadcrumb">

<li class="breadcrumb-item">

<a href="index.php">Home</a>

</li>

<li class="breadcrumb-item active" aria-current="page">Login</li>

</ol>

</div>

</div>

<!-- //page details -->

<!-- about -->

<section class="about py-5">

<div class="container py-xl-5 py-lg-3">

<div class="login px-4 mx-auto mw-100">

<h5 class="text-center mb-4">Login Now</h5>

<form action="#" method="post" name="login">

<div class="form-group">

<label>Email ID</label>

<input type="email" class="form-control" name="email" placeholder="" required="">

</div>

<div class="form-group">

<label class="mb-2">Password</label>

<input type="password" class="form-control" name="password" id="password" placeholder="" required="">

</div>

<button type="submit" class="btn submit mb-4" name="login">Login</button>

<!-- <p class="forgot-w3ls text-center pb-4">

<a href="#" class="text-white">Forgot your password?</a>

</p> -->

<p class="account-w3ls text-center pb-4" style="color:#000">

Don't have an account?

<a href="sign-up.php" >Create one now</a>

</p>

</form>

</div>

</div>

</section>

<!-- //about -->

<?php include('includes/footer.php');?>

<!-- Js files -->

<!-- JavaScript -->

<script src="js/jquery-2.2.3.min.js"></script>

<!-- Default-JavaScript-File -->

<!-- banner slider -->

<script src="js/responsiveslides.min.js"></script>

<script>

$(function () {

$("#slider4").responsiveSlides({

auto: true,

pager: true,

nav: true,

speed: 1000,

namespace: "callbacks",

before: function () {

$('.events').append("<li>before event fired.</li>");

},

after: function () {

$('.events').append("<li>after event fired.</li>");

}

});

});

</script>

<!-- //banner slider -->

<!-- fixed navigation -->

<script src="js/fixed-nav.js"></script>

<!-- //fixed navigation -->

<!-- smooth scrolling -->

<script src="js/SmoothScroll.min.js"></script>

<!-- move-top -->

<script src="js/move-top.js"></script>

<!-- easing -->

<script src="js/easing.js"></script>

<!-- necessary snippets for few javascript files -->

<script src="js/medic.js"></script>

<script src="js/bootstrap.js"></script>

<!-- Necessary-JavaScript-File-For-Bootstrap -->

<!-- //Js files -->

</body>

</html>

**SEARCH DONORS**

<?php

//error\_reporting(0);

include('includes/config.php');

?>

<!DOCTYPE html>

<html lang="zxx">

<head>

<title>Blood Bank Donar Management System | Blood Donor List </title>

<!-- Meta tag Keywords -->

<script>

addEventListener("load", function () {

setTimeout(hideURLbar, 0);

}, false);

function hideURLbar() {

window.scrollTo(0, 1);

}

</script>

<!--// Meta tag Keywords -->

<!-- Custom-Files -->

<link rel="stylesheet" href="css/bootstrap.css">

<!-- Bootstrap-Core-CSS -->

<link rel="stylesheet" href="css/style.css" type="text/css" media="all" />

<!-- Style-CSS -->

<link rel="stylesheet" href="css/fontawesome-all.css">

<!-- Font-Awesome-Icons-CSS -->

<!-- //Custom-Files -->

<!-- Web-Fonts -->

<link href="//fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,700,700i,800,800i&amp;subset=cyrillic,cyrillic-ext,greek,greek-ext,latin-ext,vietnamese"

rel="stylesheet">

<link href="//fonts.googleapis.com/css?family=Roboto+Condensed:300,300i,400,400i,700,700i&amp;subset=cyrillic,cyrillic-ext,greek,greek-ext,latin-ext,vietnamese"

rel="stylesheet">

<!-- //Web-Fonts -->

</head>

<body>

<?php include('includes/header.php');?>

<!-- banner 2 -->

<div class="inner-banner-w3ls">

<div class="container">

</div>

<!-- //banner 2 -->

</div>

<!-- page details -->

<!-- <div class="breadcrumb-agile">

<div aria-label="breadcrumb">

<ol class="breadcrumb">

<li class="breadcrumb-item">

<a href="index.php">Home</a>

</li>

<li class="breadcrumb-item active" aria-current="page">Blood Donar List</li>

</ol>

</div>

</div> -->

<!-- //page details -->

<!-- contact -->

<div class="agileits-contact py-5">

<div class="py-xl-5 py-lg-3">

<form name="donar" method="post" style="padding-left: 30px;">

<div class="row">

<div class="col-lg-4 mb-4">

<div class="font-italic">Blood Group<span style="color:red">\*</span> </div>

<div><select name="bloodgroup" class="form-control" required>

<?php $sql = "SELECT \* from tblbloodgroup ";

$query = $dbh -> prepare($sql);

$query->execute();

$results=$query->fetchAll(PDO::FETCH\_OBJ);

$cnt=1;

if($query->rowCount() > 0)

{

foreach($results as $result)

{ ?>

<option value="<?php echo htmlentities($result->BloodGroup);?>"><?php echo htmlentities($result->BloodGroup);?></option>

<?php }} ?>

</select>

</div>

</div>

<div class="col-lg-4 mb-4">

<div class="font-italic">Location </div>

<div><textarea class="form-control" name="location"></textarea></div>

</div>

</div>

<div class="row">

<div class="col-lg-4 mb-4">

<div><input type="submit" name="sub" class="btn btn-primary" value="submit" style="cursor:pointer"></div>

</div>

</div>

<!-- /.row -->

</form>

<div class="agileits-contact py-5">

<div class="py-xl-5 py-lg-3">

<div class="w3ls-titles text-center mb-5">

<h3 class="title">Blood Donar List</h3>

<span>

<i class="fas fa-user-md"></i>

</span>

<p class="mt-2">Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>

</div>

<div class="d-flex">

<div class="row package-grids mt-5" style="padding-left: 50px;">

<?php

if(isset($\_POST['sub']))

{

$status=1;

$bloodgroup=$\_POST['bloodgroup'];

$location=$\_POST['location'];

$sql = "SELECT \* from tblblooddonars where (status=:status and BloodGroup=:bloodgroup) || (Address=:location)";

$query = $dbh -> prepare($sql);

$query->bindParam(':status',$status,PDO::PARAM\_STR);

$query->bindParam(':bloodgroup',$bloodgroup,PDO::PARAM\_STR);

$query->bindParam(':location',$location,PDO::PARAM\_STR);

$query->execute();

$results=$query->fetchAll(PDO::FETCH\_OBJ);

$cnt=1;

if($query->rowCount() > 0)

{

foreach($results as $result)

{ ?>

<div class="col-md-4 pricing">

<div class="price-top">

<a href="single.html">

<img src="images/blood-donor.jpg" alt="" class="img-fluid" />

</a>

<h3><?php echo htmlentities($result->FullName);?>

</h3>

</div>

<div class="price-bottom p-4">

<h4 class="text-dark mb-3">Gender: <?php echo htmlentities($result->Gender);?></h4>

<p class="card-text"><b>Blood Group :</b> <?php echo htmlentities($result->BloodGroup);?></p>

<p class="card-text"><b>Mobile No :</b> <?php echo htmlentities($result->MobileNumber);?></p>

<p class="card-text"><b>Email ID :</b> <?php echo htmlentities($result->EmailId);?></p>

<p class="card-text"><b>Age :</b> <?php echo htmlentities($result->Age);?></p>

<p class="card-text"><b>Address :</b> <?php echo htmlentities($result->Address);?></p>

<p class="card-text"><b>Message :</b> <?php echo htmlentities($result->Message);?></p>

<a class="w3ls-button-agile" href="contact-blood.php?cid=<?php echo $result->id;?>">Request</a>

</div>

</div> <?php }}

else

{

echo htmlentities("No Record Found");

}

} ?>

</div>

</div>

</div>

</div>

<!-- //contact -->

<?php include('includes/footer.php');?>

<!-- Js files -->

<!-- JavaScript -->

<script src="js/jquery-2.2.3.min.js"></script>

<!-- Default-JavaScript-File -->

<!-- banner slider -->

<script src="js/responsiveslides.min.js"></script>

<script>

$(function () {

$("#slider4").responsiveSlides({

auto: true,

pager: true,

nav: true,

speed: 1000,

namespace: "callbacks",

before: function () {

$('.events').append("<li>before event fired.</li>");

},

after: function () {

$('.events').append("<li>after event fired.</li>");

}

});

});

</script>

<!-- //banner slider -->

<!-- fixed navigation -->

<script src="js/fixed-nav.js"></script>

<!-- //fixed navigation -->

<!-- smooth scrolling -->

<script src="js/SmoothScroll.min.js"></script>

<!-- move-top -->

<script src="js/move-top.js"></script>

<!-- easing -->

<script src="js/easing.js"></script>

<!-- necessary snippets for few javascript files -->

<script src="js/medic.js"></script>

<script src="js/bootstrap.js"></script>

<!-- Necessary-JavaScript-File-For-Bootstrap -->

<!-- //Js files -->

</body>

</html>

**ADMIN DASHBOARD**

<?php

session\_start();

error\_reporting(0);

include('includes/config.php');

if(strlen($\_SESSION['alogin'])==0)

{

header('location:index.php');

}

else{

?>

<!doctype html>

<html lang="en" class="no-js">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1, minimum-scale=1, maximum-scale=1">

<meta name="description" content="">

<meta name="author" content="">

<meta name="theme-color" content="#3e454c">

<title>BBDMS | Admin Dashboard</title>

<!-- Font awesome -->

<link rel="stylesheet" href="css/font-awesome.min.css">

<!-- Sandstone Bootstrap CSS -->

<link rel="stylesheet" href="css/bootstrap.min.css">

<!-- Bootstrap Datatables -->

<link rel="stylesheet" href="css/dataTables.bootstrap.min.css">

<!-- Bootstrap social button library -->

<link rel="stylesheet" href="css/bootstrap-social.css">

<!-- Bootstrap select -->

<link rel="stylesheet" href="css/bootstrap-select.css">

<!-- Bootstrap file input -->

<link rel="stylesheet" href="css/fileinput.min.css">

<!-- Awesome Bootstrap checkbox -->

<link rel="stylesheet" href="css/awesome-bootstrap-checkbox.css">

<!-- Admin Stye -->

<link rel="stylesheet" href="css/style.css">

</head>

<body>

<?php include('includes/header.php');?>

<div class="ts-main-content">

<?php include('includes/leftbar.php');?>

<div class="content-wrapper">

<div class="container-fluid">

<div class="row">

<div class="col-md-12">

<h2 class="page-title">Dashboard</h2>

<div class="row">

<div class="col-md-12">

<div class="row">

<div class="col-md-4">

<div class="panel panel-default">

<div class="panel-body bk-primary text-light">

<div class="stat-panel text-center">

<?php

$sql ="SELECT id from tblbloodgroup ";

$query = $dbh -> prepare($sql);

$query->execute();

$results=$query->fetchAll(PDO::FETCH\_OBJ);

$bg=$query->rowCount();

?>

<div class="stat-panel-number h1 "><?php echo htmlentities($bg);?></div>

<div class="stat-panel-title text-uppercase">Listed Blood Groups</div>

</div>

</div>

<a href="manage-bloodgroup.php" class="block-anchor panel-footer">Full Detail <i class="fa fa-arrow-right"></i></a>

</div>

</div>

<div class="col-md-4">

<div class="panel panel-default">

<div class="panel-body bk-success text-light">

<div class="stat-panel text-center">

<?php

$sql1 ="SELECT id from tblblooddonars ";

$query1 = $dbh -> prepare($sql1);;

$query1->execute();

$results1=$query1->fetchAll(PDO::FETCH\_OBJ);

$regbd=$query1->rowCount();

?>

<div class="stat-panel-number h1 "><?php echo htmlentities($regbd);?></div>

<div class="stat-panel-title text-uppercase">Registered Blood Group</div>

</div>

</div>

<a href="donor-list.php" class="block-anchor panel-footer text-center">Full Detail &nbsp; <i class="fa fa-arrow-right"></i></a>

</div>

</div>

<!-- <div class="col-md-4">

<div class="panel panel-default">

<div class="panel-body bk-info text-light">

<div class="stat-panel text-center">

<?php

$sql6 ="SELECT id from tblcontactusquery ";

$query6 = $dbh -> prepare($sql6);;

$query6->execute();

$results6=$query6->fetchAll(PDO::FETCH\_OBJ);

$query=$query6->rowCount();

?>

<div class="stat-panel-number h1 "><?php echo htmlentities($query);?></div>

<div class="stat-panel-title text-uppercase">Total Quries</div>

</div>

</div>

<a href="manage-conactusquery.php" class="block-anchor panel-footer text-center">Full Detail &nbsp; <i class="fa fa-arrow-right"></i></a>

</div>

</div> -->

<!------------------------>

<div class="col-md-4">

<div class="panel panel-danger">

<div class="panel-body bk-info text-light">

<div class="stat-panel text-center">

<?php

$sql6 ="SELECT ID from tblbloodrequirer ";

$query6 = $dbh -> prepare($sql6);;

$query6->execute();

$results6=$query6->fetchAll(PDO::FETCH\_OBJ);

$totalreuqests=$query6->rowCount();

?>

<div class="stat-panel-number h1 "><?php echo htmlentities($totalreuqests);?></div>

<div class="stat-panel-title text-uppercase">Total Blood Request Received</div>

</div>

</div>

<a href="requests-received.php" class="block-anchor panel-footer text-center">Full Detail &nbsp; <i class="fa fa-arrow-right"></i></a>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

<!-- Loading Scripts -->

<script src="js/jquery.min.js"></script>

<script src="js/bootstrap-select.min.js"></script>

<script src="js/bootstrap.min.js"></script>

<script src="js/jquery.dataTables.min.js"></script>

<script src="js/dataTables.bootstrap.min.js"></script>

<script src="js/Chart.min.js"></script>

<script src="js/fileinput.js"></script>

<script src="js/chartData.js"></script>

<script src="js/main.js"></script>

<script>

window.onload = function(){

// Line chart from swirlData for dashReport

var ctx = document.getElementById("dashReport").getContext("2d");

window.myLine = new Chart(ctx).Line(swirlData, {

responsive: true,

scaleShowVerticalLines: false,

scaleBeginAtZero : true,

multiTooltipTemplate: "<%if (label){%><%=label%>: <%}%><%= value %>",

});

// Pie Chart from doughutData

var doctx = document.getElementById("chart-area3").getContext("2d");

window.myDoughnut = new Chart(doctx).Pie(doughnutData, {responsive : true});

// Dougnut Chart from doughnutData

var doctx = document.getElementById("chart-area4").getContext("2d");

window.myDoughnut = new Chart(doctx).Doughnut(doughnutData, {responsive : true});

}

</script>

</body>

</html>

<?php } ?>

**ADMIN ADD DONOR**

<?php

session\_start();

error\_reporting(0);

include('includes/config.php');

if(strlen($\_SESSION['alogin'])==0)

{

header('location:index.php');

}

else{

if(isset($\_POST['submit']))

{

$fullname=$\_POST['fullname'];

$mobile=$\_POST['mobileno'];

$email=$\_POST['emailid'];

$age=$\_POST['age'];

$gender=$\_POST['gender'];

$blodgroup=$\_POST['bloodgroup'];

$address=$\_POST['address'];

$message=$\_POST['message'];

$status=1;

$sql="INSERT INTO tblblooddonars(FullName,MobileNumber,EmailId,Age,Gender,BloodGroup,Address,Message,status) VALUES(:fullname,:mobile,:email,:age,:gender,:blodgroup,:address,:message,:status)";

$query = $dbh->prepare($sql);

$query->bindParam(':fullname',$fullname,PDO::PARAM\_STR);

$query->bindParam(':mobile',$mobile,PDO::PARAM\_STR);

$query->bindParam(':email',$email,PDO::PARAM\_STR);

$query->bindParam(':age',$age,PDO::PARAM\_STR);

$query->bindParam(':gender',$gender,PDO::PARAM\_STR);

$query->bindParam(':blodgroup',$blodgroup,PDO::PARAM\_STR);

$query->bindParam(':address',$address,PDO::PARAM\_STR);

$query->bindParam(':message',$message,PDO::PARAM\_STR);

$query->bindParam(':status',$status,PDO::PARAM\_STR);

$query->execute();

$lastInsertId = $dbh->lastInsertId();

if($lastInsertId)

{

$msg="Your info submitted successfully";

}

else

{

$error="Something went wrong. Please try again";

}

}

?>

<!doctype html>

<html lang="en" class="no-js">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1, minimum-scale=1, maximum-scale=1">

<meta name="description" content="">

<meta name="author" content="">

<meta name="theme-color" content="#3e454c">

<title>BBDMS| Admin Add Donor</title>

<!-- Font awesome -->

<link rel="stylesheet" href="css/font-awesome.min.css">

<!-- Sandstone Bootstrap CSS -->

<link rel="stylesheet" href="css/bootstrap.min.css">

<!-- Bootstrap Datatables -->

<link rel="stylesheet" href="css/dataTables.bootstrap.min.css">

<!-- Bootstrap social button library -->

<link rel="stylesheet" href="css/bootstrap-social.css">

<!-- Bootstrap select -->

<link rel="stylesheet" href="css/bootstrap-select.css">

<!-- Bootstrap file input -->

<link rel="stylesheet" href="css/fileinput.min.css">

<!-- Awesome Bootstrap checkbox -->

<link rel="stylesheet" href="css/awesome-bootstrap-checkbox.css">

<!-- Admin Stye -->

<link rel="stylesheet" href="css/style.css">

<style>

.errorWrap {

padding: 10px;

margin: 0 0 20px 0;

background: #fff;

border-left: 4px solid #dd3d36;

-webkit-box-shadow: 0 1px 1px 0 rgba(0,0,0,.1);

box-shadow: 0 1px 1px 0 rgba(0,0,0,.1);

}

.succWrap{

padding: 10px;

margin: 0 0 20px 0;

background: #fff;

border-left: 4px solid #5cb85c;

-webkit-box-shadow: 0 1px 1px 0 rgba(0,0,0,.1);

box-shadow: 0 1px 1px 0 rgba(0,0,0,.1);

}

</style>

<script language="javascript">

function isNumberKey(evt)

{

var charCode = (evt.which) ? evt.which : event.keyCode

if (charCode > 31 && (charCode < 48 || charCode > 57) && charCode!=46)

return false;

return true;

}

</script>

</head>

<body>

<?php include('includes/header.php');?>

<div class="ts-main-content">

<?php include('includes/leftbar.php');?>

<div class="content-wrapper">

<div class="container-fluid">

<div class="row">

<div class="col-md-12">

<h2 class="page-title">Add Donor</h2>

<div class="row">

<div class="col-md-12">

<div class="panel panel-default">

<div class="panel-heading">Basic Info</div>

<?php if($error){?><div class="errorWrap"><strong>ERROR</strong>:<?php echo htmlentities($error); ?> </div><?php }

else if($msg){?><div class="succWrap"><strong>SUCCESS</strong>:<?php echo htmlentities($msg); ?> </div><?php }?>

<div class="panel-body">

<form method="post" class="form-horizontal" enctype="multipart/form-data">

<div class="form-group">

<label class="col-sm-2 control-label">Full Name<span style="color:red">\*</span></label>

<div class="col-sm-4">

<input type="text" name="fullname" class="form-control" required>

</div>

<label class="col-sm-2 control-label">Mobile No<span style="color:red">\*</span></label>

<div class="col-sm-4">

<input type="text" name="mobileno" onKeyPress="return isNumberKey(event)" maxlength="10" class="form-control" required>

</div>

</div>

<div class="form-group">

<label class="col-sm-2 control-label">Email id </label>

<div class="col-sm-4">

<input type="email" name="emailid" class="form-control">

</div>

<label class="col-sm-2 control-label">Age<span style="color:red">\*</span></label>

<div class="col-sm-4">

<input type="text" name="age" class="form-control" required>

</div>

</div>

<div class="form-group">

<label class="col-sm-2 control-label">Gender <span style="color:red">\*</span></label>

<div class="col-sm-4">

<select name="gender" class="form-control" required>

<option value="">Select</option>

<option value="Male">Male</option>

<option value="Female">Female</option>

</select>

</div>

<label class="col-sm-2 control-label">Blood Group<span style="color:red">\*</span></label>

<div class="col-sm-4">

<select name="bloodgroup" class="form-control" required>

<option value="">Select</option>

<?php $sql = "SELECT \* from tblbloodgroup ";

$query = $dbh -> prepare($sql);

$query->execute();

$results=$query->fetchAll(PDO::FETCH\_OBJ);

$cnt=1;

if($query->rowCount() > 0)

{

foreach($results as $result)

{ ?>

<option value="<?php echo htmlentities($result->BloodGroup);?>"><?php echo htmlentities($result->BloodGroup);?></option>

<?php }} ?>

</select>

</div>

</div>

<div class="hr-dashed"></div>

<div class="form-group">

<label class="col-sm-2 control-label">Address</label>

<div class="col-sm-10">

<textarea class="form-control" name="address" ></textarea>

</div>

</div>

<div class="hr-dashed"></div>

<div class="form-group">

<label class="col-sm-2 control-label">Message<span style="color:red">\*</span></label>

<div class="col-sm-10">

<textarea class="form-control" name="message" required> </textarea>

</div>

</div>

<div class="form-group">

<div class="col-sm-8 col-sm-offset-2">

<button class="btn btn-default" type="reset">Cancel</button>

<button class="btn btn-primary" name="submit" type="submit">Save changes</button>

</div>

</div>

</form>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

<!-- Loading Scripts -->

<script src="js/jquery.min.js"></script>

<script src="js/bootstrap-select.min.js"></script>

<script src="js/bootstrap.min.js"></script>

<script src="js/jquery.dataTables.min.js"></script>

<script src="js/dataTables.bootstrap.min.js"></script>

<script src="js/Chart.min.js"></script>

<script src="js/fileinput.js"></script>

<script src="js/chartData.js"></script>

<script src="js/main.js"></script>

</body>

</html>

<?php } ?>

**ABOUT AS ADMIN**

<?php

error\_reporting(0);

include('includes/config.php');

?>

<!DOCTYPE html>

<html lang="zxx">

<head>

<title>Blood Bank Donar Management System | About Us </title>

<!-- Meta tag Keywords -->

<script>

addEventListener("load", function () {

setTimeout(hideURLbar, 0);

}, false);

function hideURLbar() {

window.scrollTo(0, 1);

}

</script>

<!--// Meta tag Keywords -->

<!-- Custom-Files -->

<link rel="stylesheet" href="css/bootstrap.css">

<!-- Bootstrap-Core-CSS -->

<link rel="stylesheet" href="css/style.css" type="text/css" media="all" />

<!-- Style-CSS -->

<link rel="stylesheet" href="css/fontawesome-all.css">

<!-- Font-Awesome-Icons-CSS -->

<!-- //Custom-Files -->

<!-- Web-Fonts -->

<link href="//fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,700,700i,800,800i&amp;subset=cyrillic,cyrillic-ext,greek,greek-ext,latin-ext,vietnamese"

rel="stylesheet">

<link href="//fonts.googleapis.com/css?family=Roboto+Condensed:300,300i,400,400i,700,700i&amp;subset=cyrillic,cyrillic-ext,greek,greek-ext,latin-ext,vietnamese"

rel="stylesheet">

<!-- //Web-Fonts -->

</head>

<body>

<?php include('includes/header.php');?>

<!-- banner 2 -->

<div class="inner-banner-w3ls">

<div class="container">

</div>

<!-- //banner 2 -->

</div>

<!-- page details -->

<!-- <div class="breadcrumb-agile">

<div aria-label="breadcrumb">

<ol class="breadcrumb">

<li class="breadcrumb-item">

<a href="index.php">Home</a>

</li>

<li class="breadcrumb-item active" aria-current="page">About Us</li>

</ol>

</div>

</div> -->

<!-- //page details -->

<!-- about -->

<section class="about">

<div class="container py-xl-2 py-lg-2">

<div class="w3ls-titles text-center mb-md-5 mb-4">

<h3 class="title">Who We Are?</h3>

<span>

<i class="fas fa-user-md"></i>

</span>

<p class="py-3">Our mission is to create a blood donors with those who need it, makes the process of blood donation is more efficient and easy accessible. Through our website, our aim to awareness and educate the people,to know importance of regular blood donation. By using technology to streamline the process and connect donors with recipients in real-time, our hope to make a positive impact on the lives of those who need of life-saving blood transfusions.</p>

</div>

</div>

</section>

<!-- our teams -->

<section class="p-5">

<h3 class="title py-xl-2 py-lg-2">Behind This Project?</h3>

<div class="card-deck">

<div class="card">

<img src="images\me.jpg" class="card-img-top team-img" alt="Abdul Basith" style="height:50%">

<div class="card-body">

<h4 class="card-title">Abdul Basith</h4>

<p class="card-text"> i'm Basith who create this project,i'm a final year Computer Science student at Ananda College. With a passion for staying current and continuously learning,i'm always seeing out for new opportunities to expand my knowledge and skills.</p>

<div class="row p-2">

<ul class="top-right-info">

</ul>

</div>

</div>

</div>

<div class="card">

<img src="images\Aathiyen.jpg" class="card-img-top team-img" alt="Aathiyan" style="height:50%">

<div class="card-body">

<h4 class="card-title">Aathiyan</h4>

<p class="card-text">Aathiyan is a valuable team member. His dedication to growth and continuous learning makes him a valuable asset to any project. With a strong foundation in these areas, Aathiyan is well-equipped to make valuable contributions to the team.</p>

<div class="row p-2">

<ul class="top-right-info">

</ul>

</div>

</div>

</div>

</div>

</section>

<!-- //about -->

<?php include('includes/footer.php');?>

<!-- Js files -->

<!-- JavaScript -->

<script src="js/jquery-2.2.3.min.js"></script>

<!-- Default-JavaScript-File -->

<!-- banner slider -->

<script src="js/responsiveslides.min.js"></script>

<script>

$(function () {

$("#slider4").responsiveSlides({

auto: true,

pager: true,

nav: true,

speed: 1000,

namespace: "callbacks",

before: function () {

$('.events').append("<li>before event fired.</li>");

},

after: function () {

$('.events').append("<li>after event fired.</li>");

}

});

});

</script>

<!-- //banner slider -->

<!-- fixed navigation -->

<script src="js/fixed-nav.js"></script>

<!-- //fixed navigation -->

<!-- smooth scrolling -->

<script src="js/SmoothScroll.min.js"></script>

<!-- move-top -->

<script src="js/move-top.js"></script>

<!-- easing -->

<script src="js/easing.js"></script>

<!-- necessary snippets for few javascript files -->

<script src="js/medic.js"></script>

<script src="js/bootstrap.js"></script>

<!-- Necessary-JavaScript-File-For-Bootstrap -->

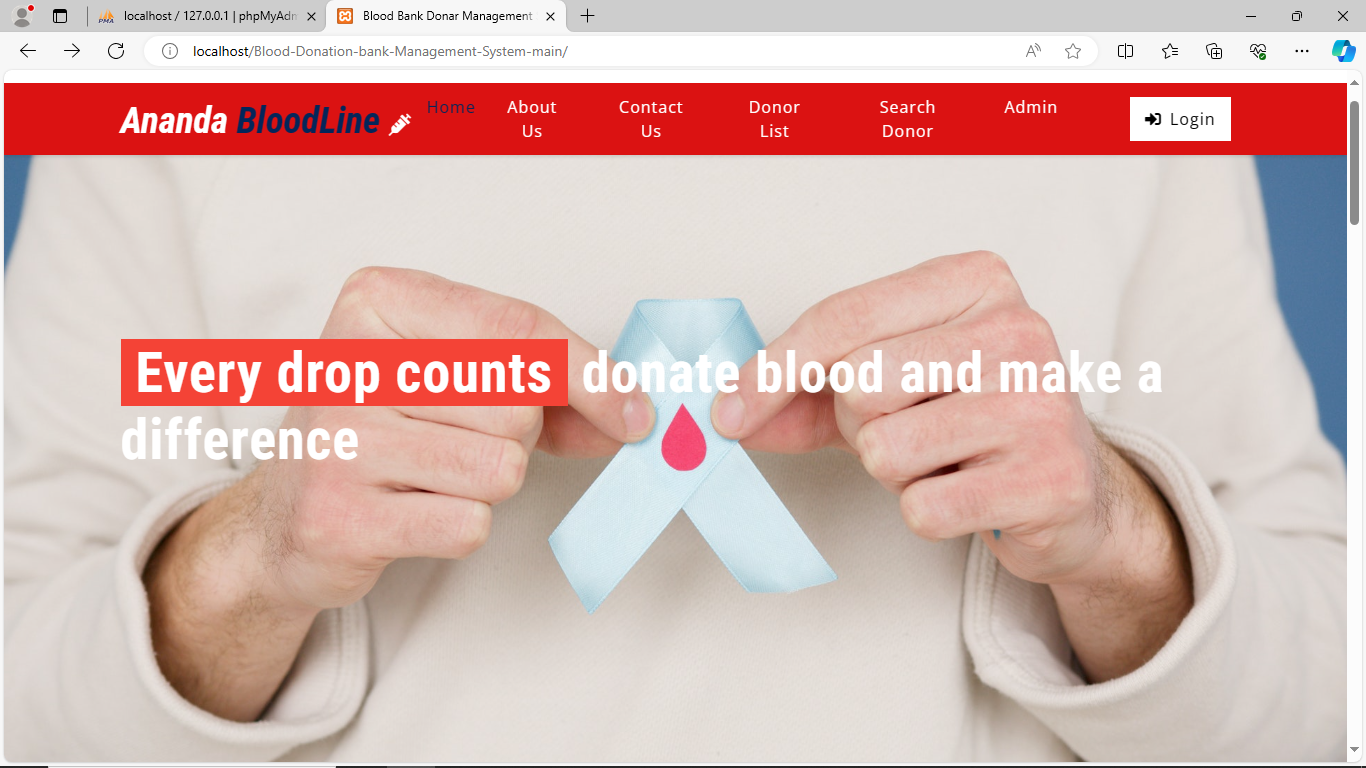
<!-- //Js files -->

</body>

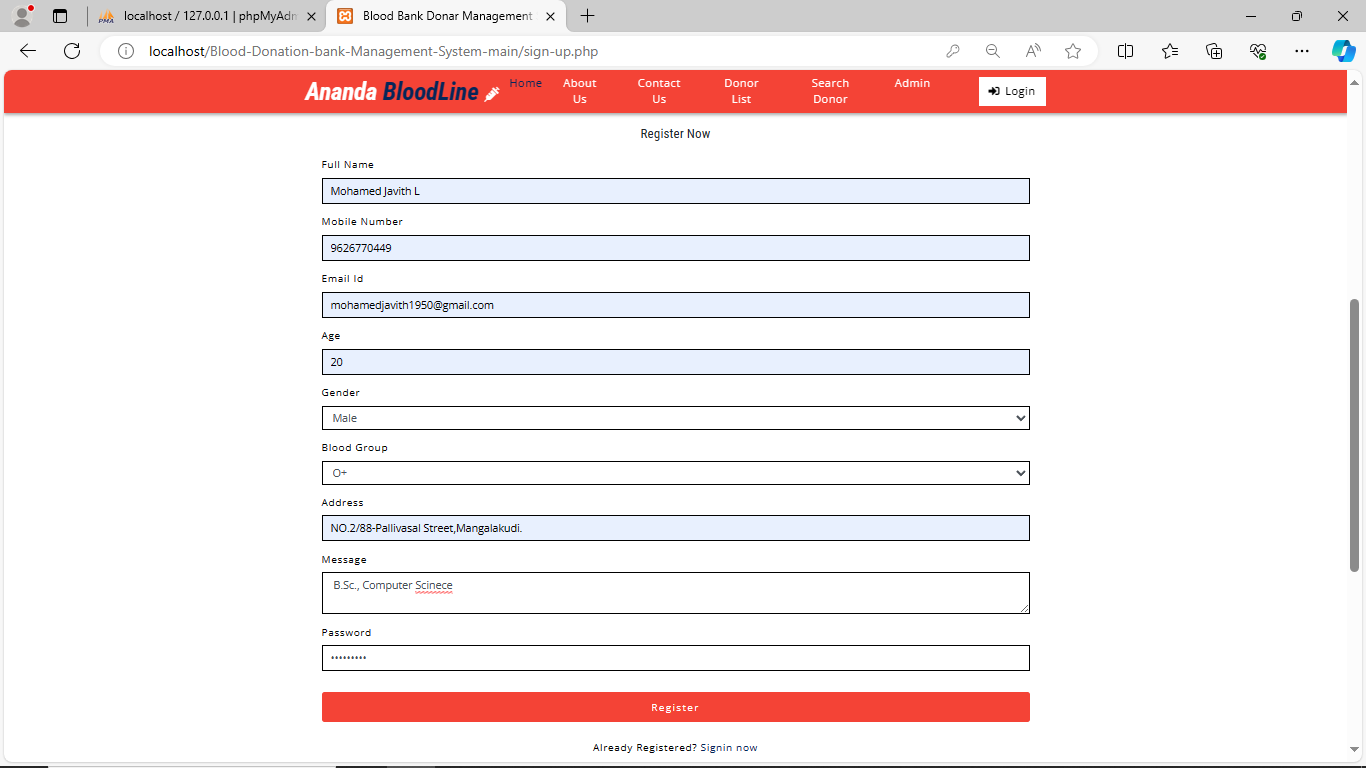
</html>

1. **SCREEN SHOTS**

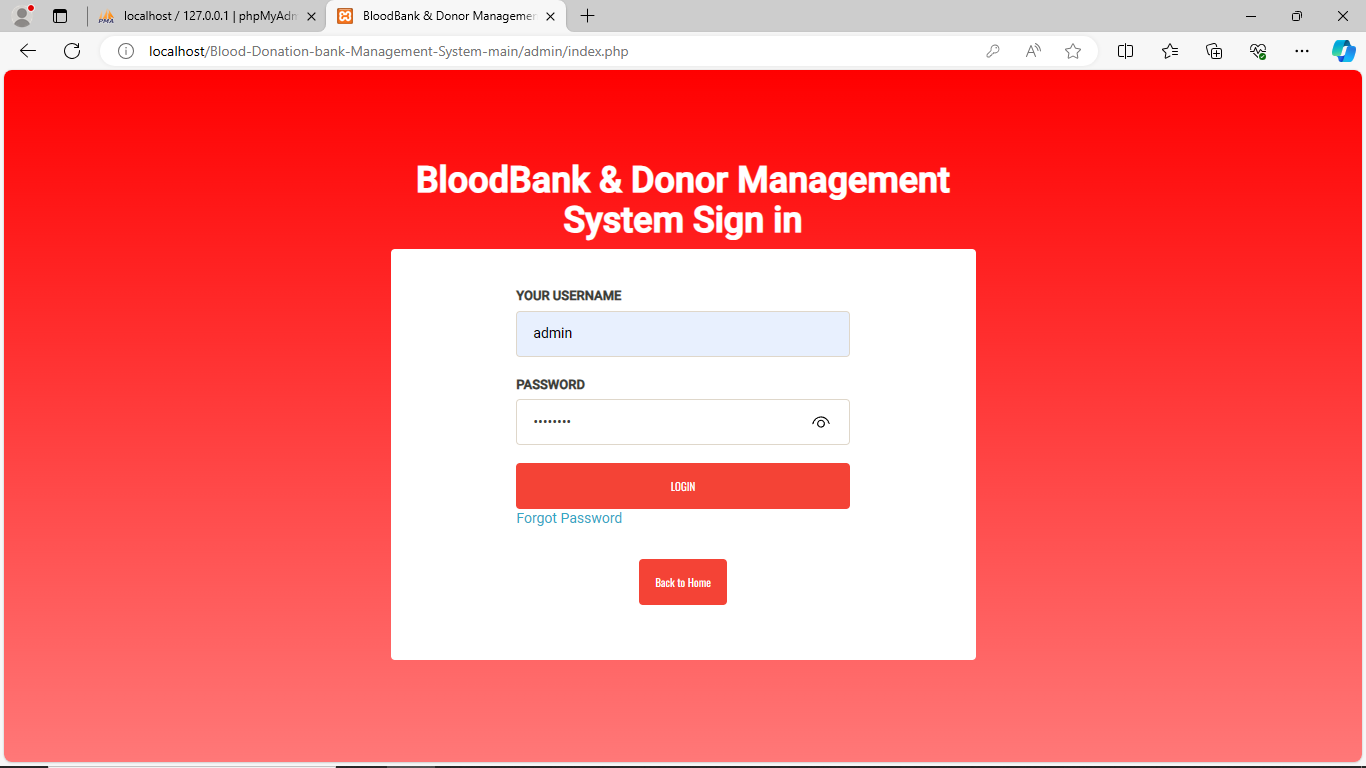
**HOME PAGE**

****

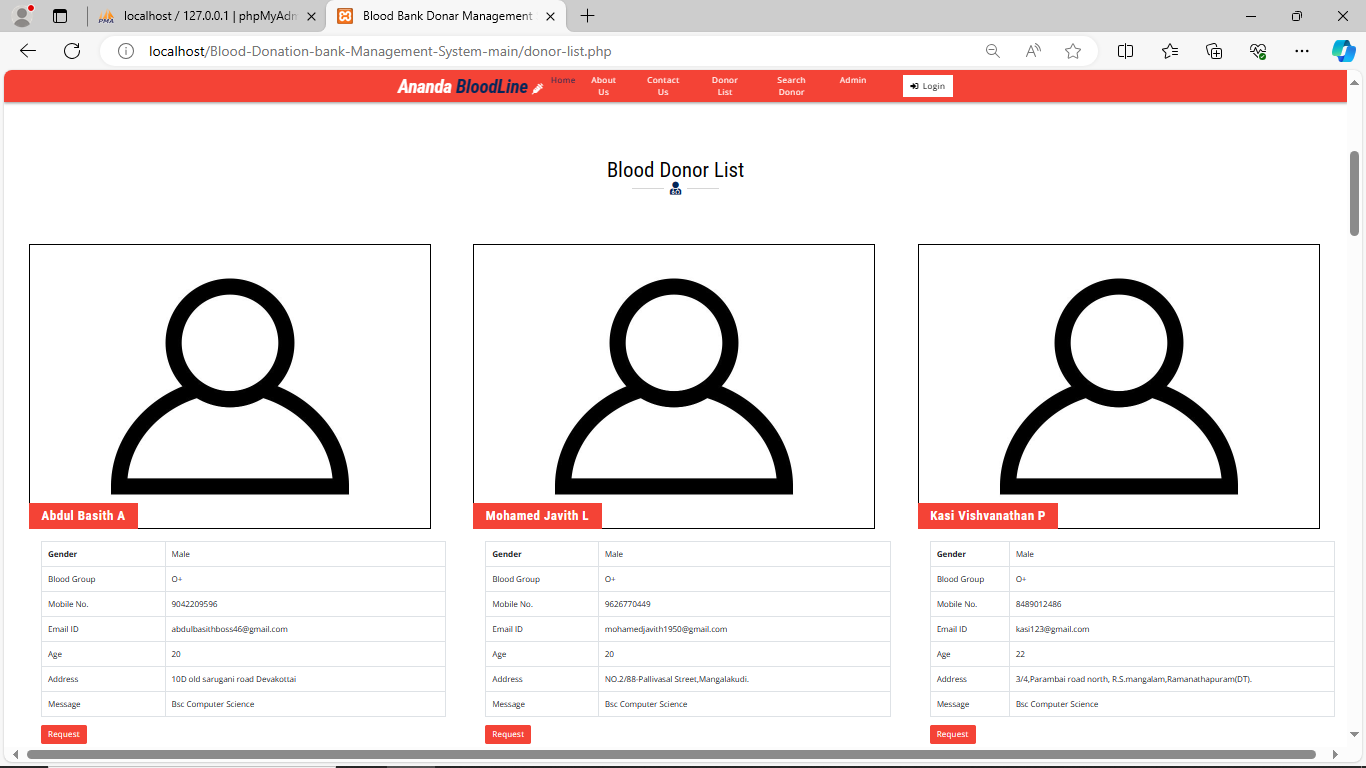
**SIGN UP PAGE**

****

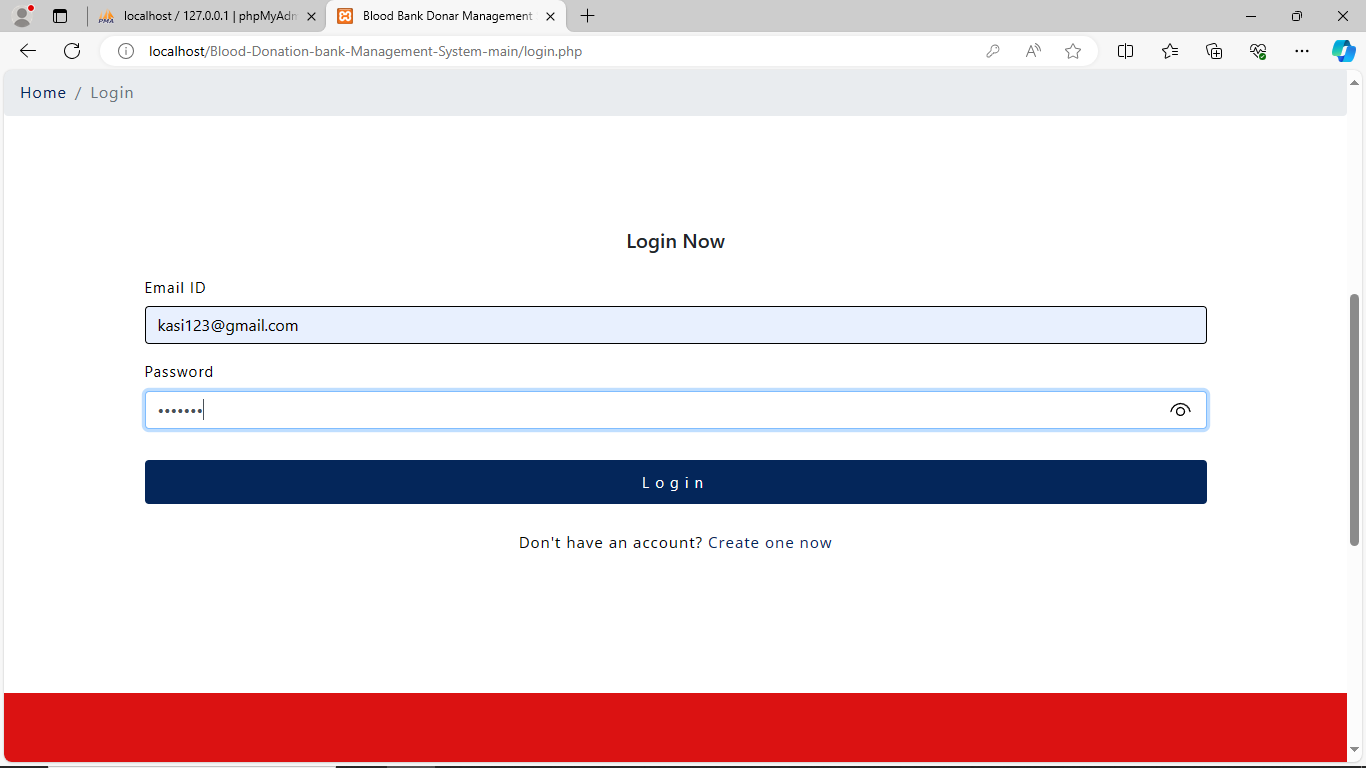
**ADMIN LOGIN PAGE**

****

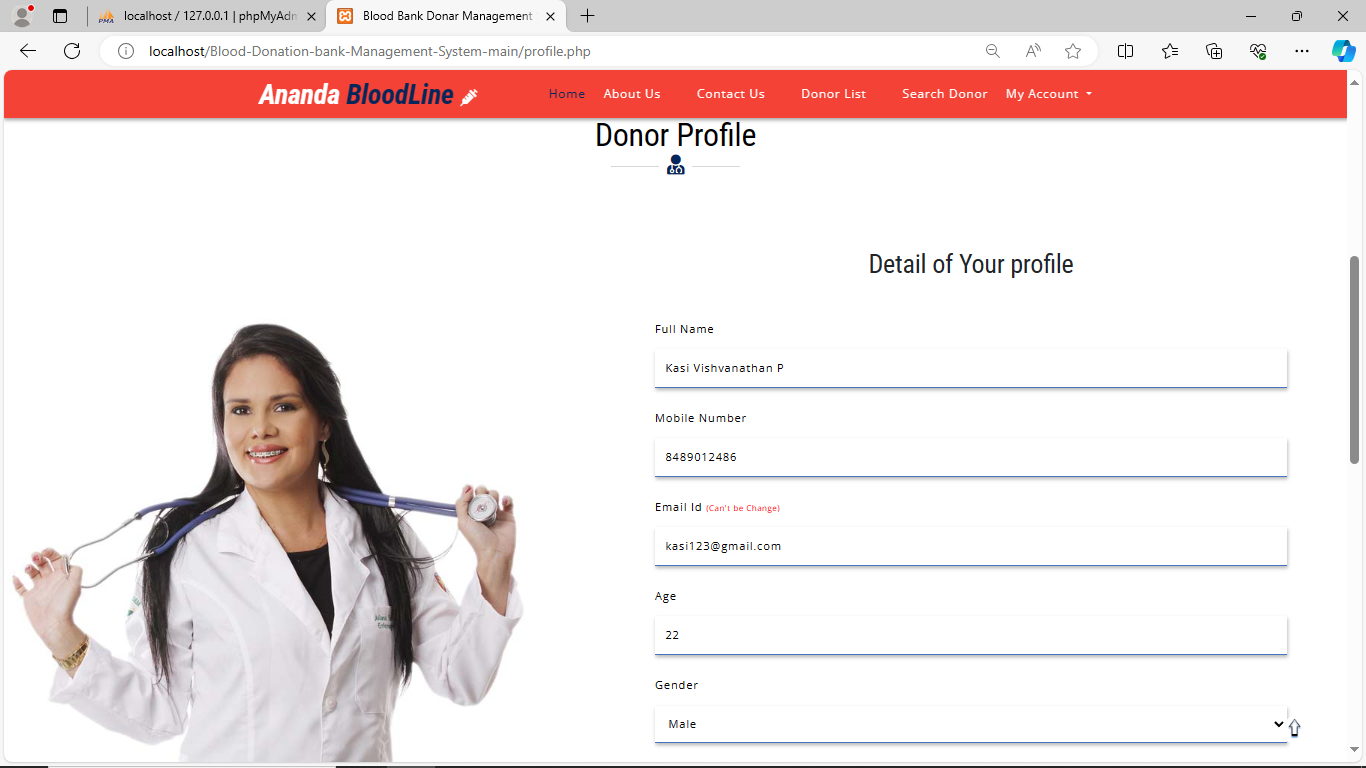
**SEARCH DONORS**

****

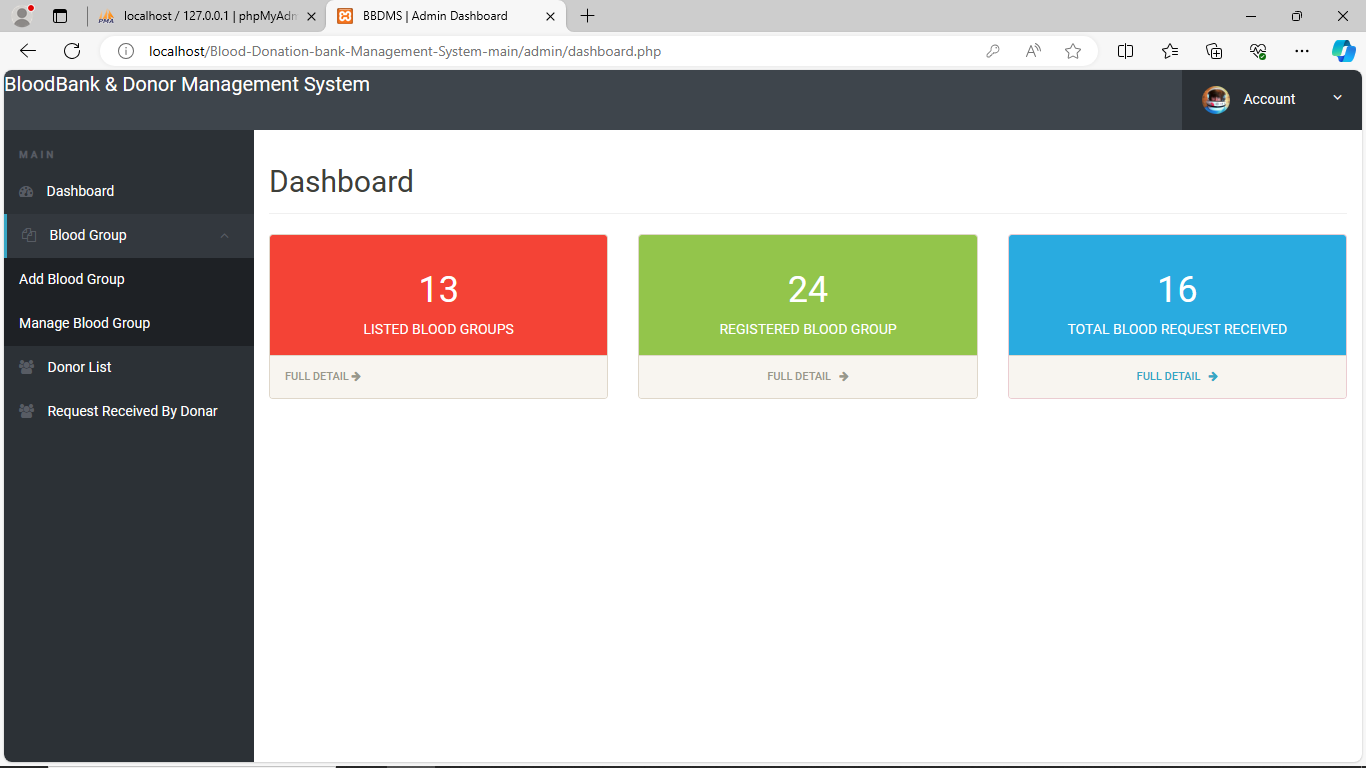
**USER LOGIN**

****

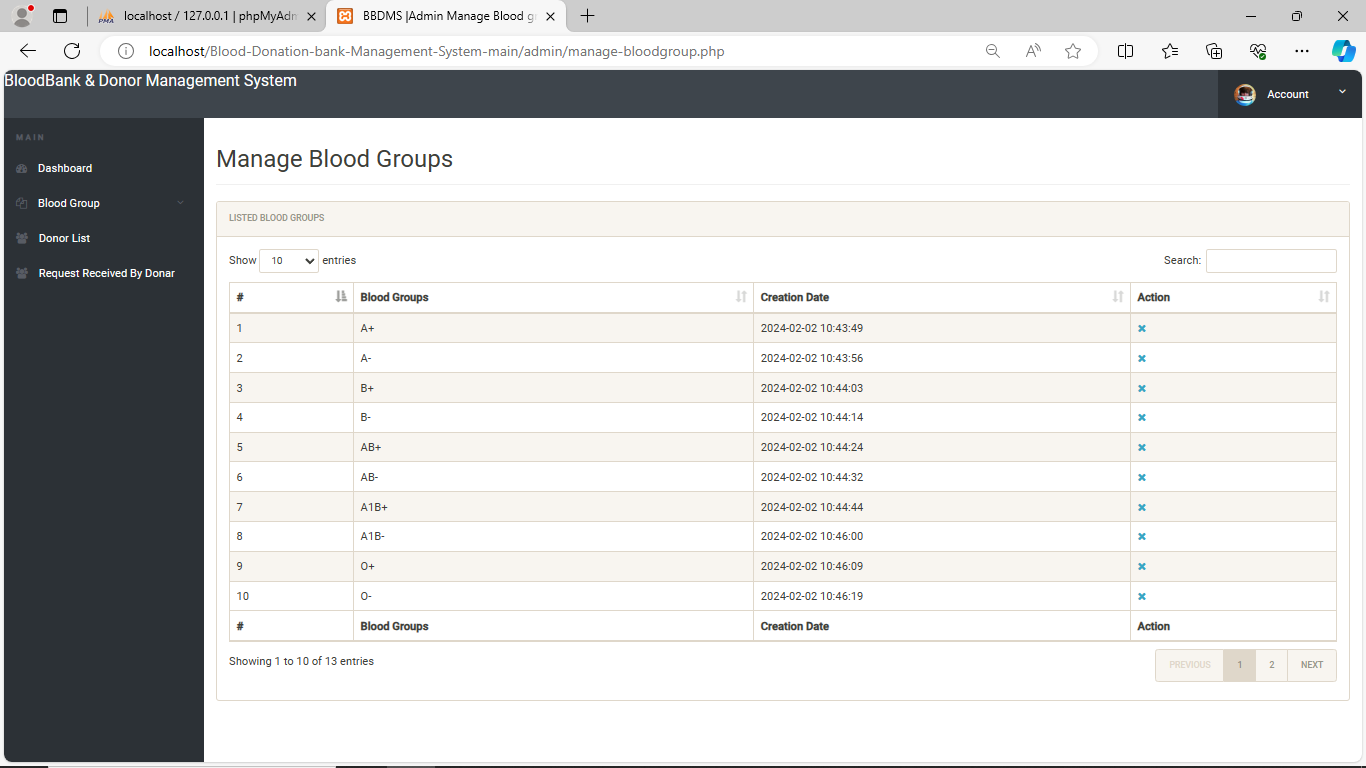
**USER PROFILE**

****

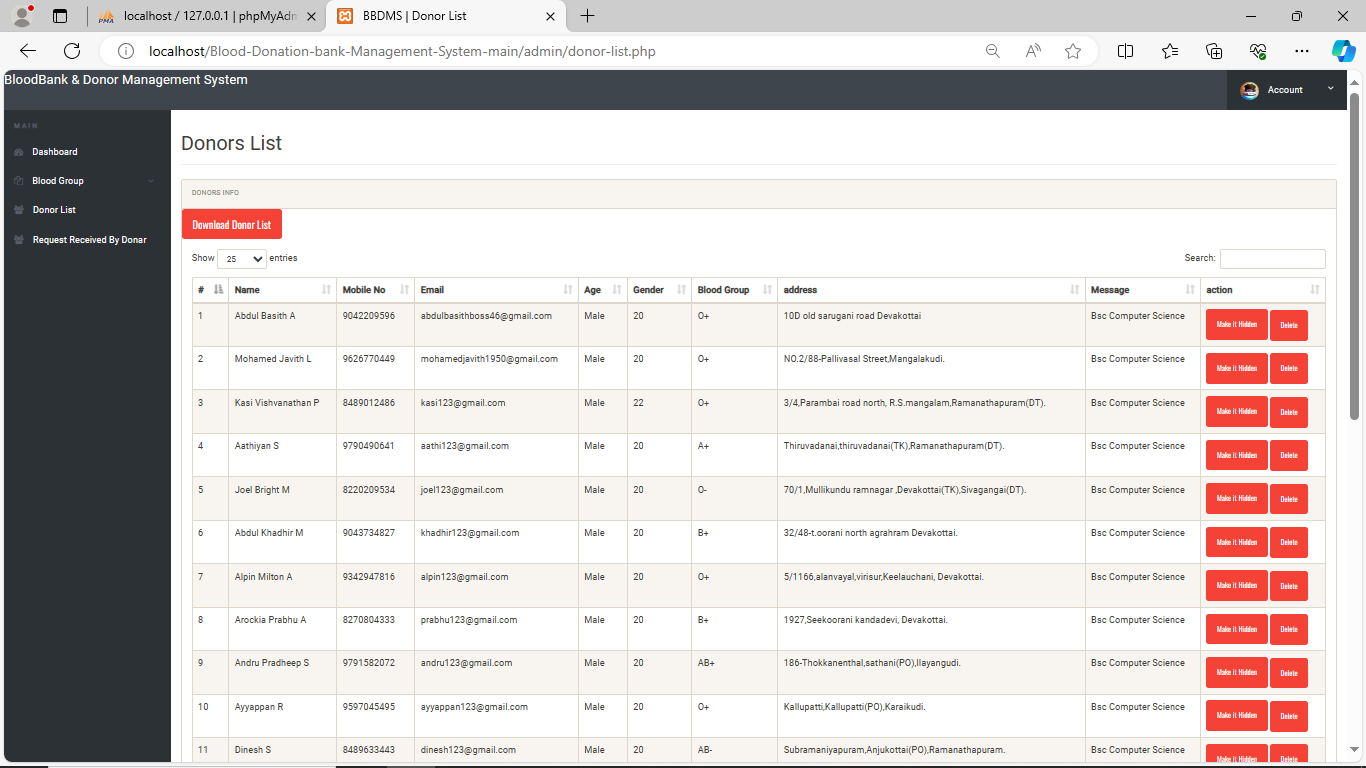
**ADMIN DASHBOARD**

****

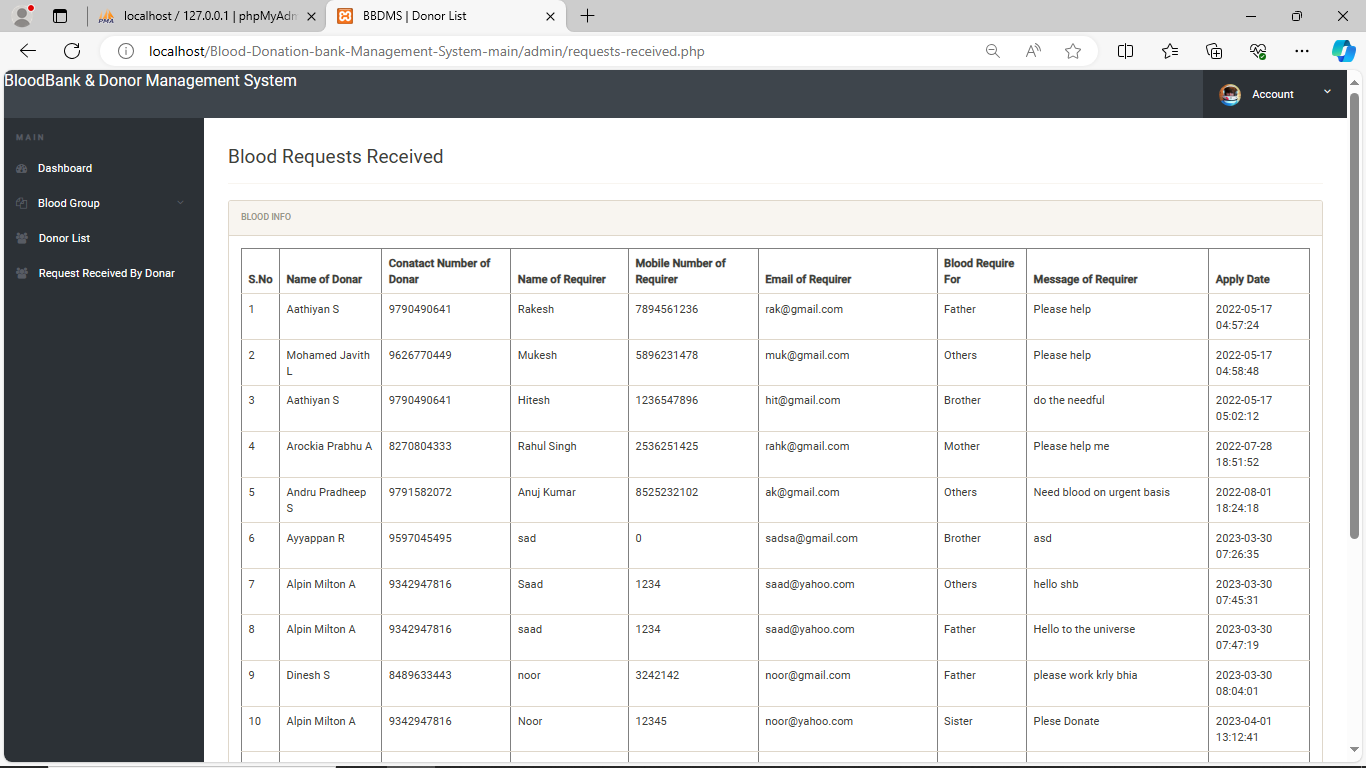
**MANAGE BLOOD GROUP**

****

**ADMIN MANAGE DONOR LIST**

****

**BLOOD REQUESTS RESIVE LIST**

****