**TABLE OF CONTENTS**

**CHAPTER NO. TITLE PAGE NO.**

* + - * **Cover Page** Ⅰ
      * **Certificate** Ⅱ

**Abstract** Ⅲ

* + - * **Acknowledgement** Ⅳ
      * **Table of contents** Ⅴ

**List of Figures**  Ⅷ

* + - * **List of Abbreviations** Ⅸ
* **1 INTRODUCTION 1** 
  + - * 1.1Introduction 1
      * 1.2 Need of Blood Management System 1
      * 1.3 Abstract of Blood Bank Management System 2
      * 1.4 Benefits 3
      * 1.5 Features 3
      * 1.5.1 Blood Camp Management & Reporting 3
      * 1.5.2 Donor Test Results Management and 4
      * Adverse Reaction Data Management
      * 1.6 Application 6
      * 1.7 Objectives 7
* **2**  **FRONT END**  **8**
  + - * 2.1 Architecture of Front End user 9
      * 2.1.1 Architecture and Concepts 9
      * 2.1.2 Transparent to use 9
      * 2.1.3 Flexible storage: various storage handler 10
      * 2.1.4 Built-in slam defense to avoid overloading 10
      * 2.1.5 Unique approach to caching 10
      * 2.2 Software & Tools Used 11
      * 2.2.1 Introduction of PHP 11
      * 2.2.2 HTML 13
      * 2.2.3 HTML 5 15
      * 2.2.4 CSS 16
      * 2.2.5 JAVASCRIPT 17

* **3 BACK END 19**
* 3.1 MySQL’s Logical Architecture 19
* 3.2 Software and tools used 21
* 3.2.1 My Sql 21
* 3.2.2 WAMP 24
* **4 MINIMUM HARDWARE SPECIFICATION 32**

* 4.1 Hardware Requirements 32
* 4.2 Software Requirements 32
* **5 DETAILED DESCRIPTION OF TECHNOLOGY USED 33**
* 5.1 Reasons for using PHP 33
* 5.1.1 Learning PHP is easy 33
* 5.1.2 Its Performance 33
* 5.1.3 Its Open Source, We can modify it 34
* 5.1.4 PHP Syntax 34
* 5.2 HTML 35
* 5.2.1 HTML Images-The <img>Tag and the SRC Attributes 36
* 5.2.2 HTML forms 36
* 5.2.3 Image tag(<img>) 36
* 5.2.4 HTML Lists 37
* 5.3 HTML5 37
* 5.4 CSS 37
* 5.5 JavaScripts 39
* 5.6 MySql’s Logical Architecture 40
* 5.7 Software and tools used 41
* 5.7.1 My Sql 41
* **6 SOFTWARE DEVELOPMENT PROCESS 44**

* 6.1 Life cycle used to develop this project 44
* 6.2 Overview 44
* 6.3 System Development Phases 45
* 6.3.1 initiation Phases 45
* 6.3.2 System Concept Development 46
* 6.3.3 Planning Requirement Analysis 47
* 6.3.4 Design 49
* 6.3.5 Development 50
* 6.3.6 Integration Test 50
* 6.3.7 Implementation 50
* 6.3.8 Operation & Maintenance 51
* **7 MODULE DESCRIPTION 52**
* 7.1.1 Admin 52
* 7.2 User 52
* 7.2.1 Admin Level User 52
* 7.2.2 General Level User 53
* 7.3.1 Main Window 54
* 7.3.2 Registration Phase 55
* 7.3.3 Donar Login 56
* **8 COMPLETE DFD & E-R DIAGRAM 57**

* 8.1 Introduction ER Diagram 57
* 8.1.1 Entity 57
* 8.1.2 Weak Entity 57
* 8.1.3 Attribute 57
* 8.1.4 Key Attribute 57
* 8.1.5 Multivalued Attribute 58
* 8.1.6 Derived Attribute 58
* 8.1.7 Relationships 58
* 8.1.8 Recursive Relationship 59
* 8.2 Data Flow Diagram 62
* 8.2.1 Introduction Of DFD 62
* 8.2.2 Context Diagram 62
* 8.2.3 Need Of DFD 62
* 8.2.4 Primitive symbols used for constructing DFDs 63
* 8.2.4.1 Function Symbol 63
* 8.2.4.2 External entity symbol 63
* 8.2.4.3 Data flow Symbol 64
* 8.2.4.4 Data store Symbol 64
* 8.2.5 DFD for Admin Login 64
* 8.2.6 DFD for User Login 65
* 8.2.7 DFD for Account Creation 66

* **9 SOFTWARE TESTING 68**
* 9.1 Levels of Testing 68
* 9.2 Functional Testing 69
* 9.3 Structural Testing 69
* 9.4 Unit Testing 69
* 9.4.1 Testing admin login form 70
* 9.5 Integrity Testing 70
* 9.5.1.2 White Box Testing 71
* **10 COMPLETE SNAPSHOTS 73**

* 10.1 Home Page 73
* 10.2 Registration 74
* 10.3 Request for Blood 74
* 10.4 Camps 75
* 10.5 Donar Login 76
* 10.6 Search 77
* 10.7 Contact 78
* 10.8 Donar Pannel 79
* 10.9 Change Password 80
* 10.10 Update profile 81
* 10.11 Blood Donated 81
* 10.12 Admin Pannel 82
* 10.13 Add User 83
* 10.14 Update User 84
* 10.15 Add City 85
* 10.16 Update City 85
* 10.17 Add Camps 86
* 10.18 Update Camps 87
* 10.19 Add State 88
* 10.20 Update State 89
* 10.21 View Blood Group 90
* **11 SUMMARY & CONCLUSION 92**
* **12 FUTURE RECOMMENDATION 93**
* **13 REFERENCES 94**

**LIST OF FIGURES**

**FIG NO. TITLE PAGE NO.**

* 2.1 PHP architecture 9
* 3.1 MySQL Logical architecture 20
* 3.1.1 Storage Engine API 21
* 3.2.2 WAMP5 Setup 25
* 8.1.8 E-R daigram 61
* 8.2.5 DFD for admin login 65
* 8.2.6 DFD for user login 66
* 8.2.7 DFD for Account Creation 66
* 8.2.8 data flow diagram 67
* 10.1 Home page of life saver 73
* 10.2 Registeration page of life saver 74
* 10.3 Request for Blood page 75
* 10.4 Camps of Blood group 76
* 10.5 DONAR Log In 77
* 10.6 Search For Blood 78
* 10.7 Contact Us 79
* 10.8 Welcome To Donar Pannel 80
* 10.9 Change Password 80
* 10.10 Update profile 81
* 10.11 Blood Donated 82
* 10.12. Admin Pannel 83
* 10.13 Add User 84
* 10.14 Update user 84
* 10.15 Add city 85
* 10.16 Update city 86
* 10.17 Add camp 87
* 10.18 Update camp 88
* 10.19 Add State 89
* 10.20 Update State 90
* 10.21 View Blood Group 91

**LIST OF ABBREVIATION**

**APC**  **:** Acquisition and Preservation Category

**API :** Application Program Interface

**BBMS**  **:** Blood Bank Management System

**CRT :** Cathode Ray Tube

**CSS :** Cascading Style Sheets

**DFD :** Data Flow Diagrams

**ER**  **:** Entity-Relationship

**HTML**  **:** Hypertext Markup Language

**LCD :** Liquid Crystal Display

**PHP**  **:** Personal Home Page(Hypertext Preprocessor)

**RAM :** Random Access Memory

**SDK :** Software Development Kit

**SDLC**  **:** System Development Life Cycle

**SQL**  **:** Structured Query Language

**URL :** Uniform Resource Locator

**W3C :** World Wide Web Consortium

**WAMP :** Window Apache MySQL and PHP

**WHATWG :** Web Hypertext Application Technology Working Group

**CHAPTER-1 :INTRODUCTION**

**Blood Bank Management System**

**1.1 Introduction**

The BLOOD BANK MANAGEMENT SYSTEM is great project. this project is designed for successful completion of project on blood bank management system. the basic building aim is to provide blood donation service to the city recently. Blood Bank Management System (BBMS) is a browser based system that is designed to store, process, retrieve and analyze information concerned with the administrative and inventory management within a blood bank. This project aims at maintaining all the information pertaining to blood donors, different blood groups available in each blood bank and help them manage in a better way. Aim is to provide transparency in this field, make the process of obtaining blood from a blood bank hassle free and corruption free and make the system of blood bank management effective.

The *Blood bank system project report* contain information related to blood like

* Blood type
* Date of Donation of blood
* validity of Blood s
* Available Blood group

**1.2 Need of Blood Bank Management System**

Bank blood donation system in java is planned to collect blood from many donators in short from various sources and distribute that blood to needy people who require blood. To do all this we require high quality software to manage those jobs. The government spending lot of money to develop high quality “Blood Bank management system project”. For do all those kinds of need blood bank management system project in java contain modules which are include the detail of following areas:

* Blood Donor
* Equipments
* Stick
* Blood Recipient
* Blood collection
* Camp
* Stock details
* blood bank system project Reports
* Blood issued
* Blood bank system project

**1.3 Abstract of Blood Bank Management System**

Help Line is an voluntary and non-governmental organization.It maintains Online library of blood donors in India. Sometimes Doctors and Blood bank project have to face the difficulty in finding the blood group Donors at right time. Help Line has attempted to provide the answer by taking upon itself the task of collecting Blood bank project nationwide for the cause and care of people in need.

At any point of time the people who are in need can reach the donors through our search facility. By mobilizing people and organization who desire to make a difference in the lives of people in need. On the basis of humanity, Everyone is welcome to register as a blood donor.

Blood Bank Management System (BBMS) is a browser based system that is designed to store, process, retrieve and analyze information concerned with the administrative and inventory management within a blood bank. This project aims at maintaining all the information pertaining to blood donors, different blood groups available in each blood bank and help them manage in a better way. Aim is to provide transparency in this field, make the process of obtaining blood from a blood bank hassle free and corruption free and make the system of blood bank management effective

**1.4 Benefits**

**Our Vision**

In the IT era, there are almost not any fields exist where computers are not used. Techshot would like to contribute to the total SATISFACTION to its esteemed CUSTOMERS by providing them with the high quality products.

Techshot wants to make products highly reliable, affordable, & consistent which will serve the customer domain.

Techshot concerned for its customers & serves them in precise time, with right product of right quality. By enhancing consulting and other potentials, we help move customers forward in each & every part of their businesses, from strategic planning to day-to-day operations.

Our Clients benefit from access to information solutions that help them better cope-up their business, cooperate with customers and make financial and operational decisions.

**Our Mission**

To endow with strategic and technical expertise to companies wanting to leverage the latest innovations.Our mission is to Define Quality Policy for the IT era, set new span for Services to customers.

**1.5 Features:**

**Blood Camp Management And Reporting**

|  |  |  |
| --- | --- | --- |
|  | * Provides recording of details of camp beginning from allocation of staff, details of facilities available in the camp venue. * Provides assigning of donor to a particular camp and generate camp organizer report * Automated report generation of camp details for submission to the Government * **Donor Management** * The system allows automatic component data generation based on the component selected in the blood donor form. * The system allows bulk update for serology for blood units. Serology result for 00000000many donors can be updated at once. * The system allows for either component creation before serology test or vice versa. Based on the serology test, the component created are updated automatically * The system allows bar-coded blood bag number entry * All donor related reports are excel downloadable * All Reports provides filtering over many factors like Blood Group, gender, area, blood Camp, date of donation, donor type etc. * The system provides easy link for easy edit or adding details for various sections of the donor form * During form filling, the system notifies the user how much percentage of donor data has been updated   **Donor Test Results Management and Adverse Reaction Data Management**   * Provides filterable selections for donor selections * Excel download of all reports * The reports are highly configurable and can be configured to display data as per institution requirements. * **Search based on Component ID, Donor Registration ID, Donor BloodBag Number and Donor Name**   a) The results displayed in search is highly configurable   * The search functionality also allows for site-wide search. It means a user can search for any data available in the system * Custom links can be added in the search results to allow easier navigation and accessibility * **Blood Components Management** * Automatic generation of components form donor form * Based on the date of collection, the system automatically derives the date of expiry and disallows issue of component if unit has expired * Until the serology test is done, the system marks the status of the the component as test awaited. And only after serology test is done, the component is marked for Ready for Issue * The Available components list is available and the system automatically generates the list of components that are ready for issue to be available. * **Patient Management System**      * Captures patient personal information as well as the hospital where blood is requireda) * The system allows for reserving a unit for 24 hours for a patient) * The blood component issued, the payment made as well as link to the final bill is available when the patient page is opened * The data allows reports like: Issue Register, Reserved Units and Patient Inventory Liste) * The system allows for capturing transfusion reaction data * **Blood Issue and Billing**      * Ability to provide adjustments in the final payment receipt for concession for blood unit * The system prevents blood issue if cross-match is not done or fails * Final bill gets generated only if only the payment has been accounted for * Final bill gets generated only component selected has been serology tested and is ready for issue * Auto-generated final receipt * Auto-generated Cross-matching report * **Managing Practical Solutions For Blood Bank Management**      * The system allows components to be created before serology and vice-versa. * The system takes care to automatically update the components when serology is done. |  |

**1.6 Applications:** This application is built such a way that it should suits for all type of blood banks in future. So every effort is taken to implement this project in this blood bank, on successful implementation in this blood bank, we can target other blood banks in the city.

**1.7 OBJECTIVE:**

The main objective of this application is to automate the complete operations of the blood bank. They need maintain hundreds of thousands of records. Also searching should be very faster so they can find required details instantly.

To develop a web-based portal to facilitate the co-ordination between supply and demand of blood . This system makes conveniently available good quality, safe blood and other blood components, which can be provided in a sound, ethical and acceptable manner, consistent with the long-term well being of the community. It actively encourage voluntary blood donation, motivate and maintain a well-indexed record of blood donors and educate the community on the benefits of blood donation. This will also serve as the site for interaction of best practices in reducing unnecessary utilization of blood and help the state work more efficiently towards self-sufficiency in blood.

The system will provide the user the option to look at the details of the existing Donor List, Blood Group and to add a new Donor. It also allows the user to modify the record. The administrator can alter all the system data.

**CHAPTER-2: FRONT END**

The front end is an [interface](https://en.wikipedia.org/wiki/Interface_(computer_science)) between the user and the back end. The front and back ends may be distributed amongst one or more systems.

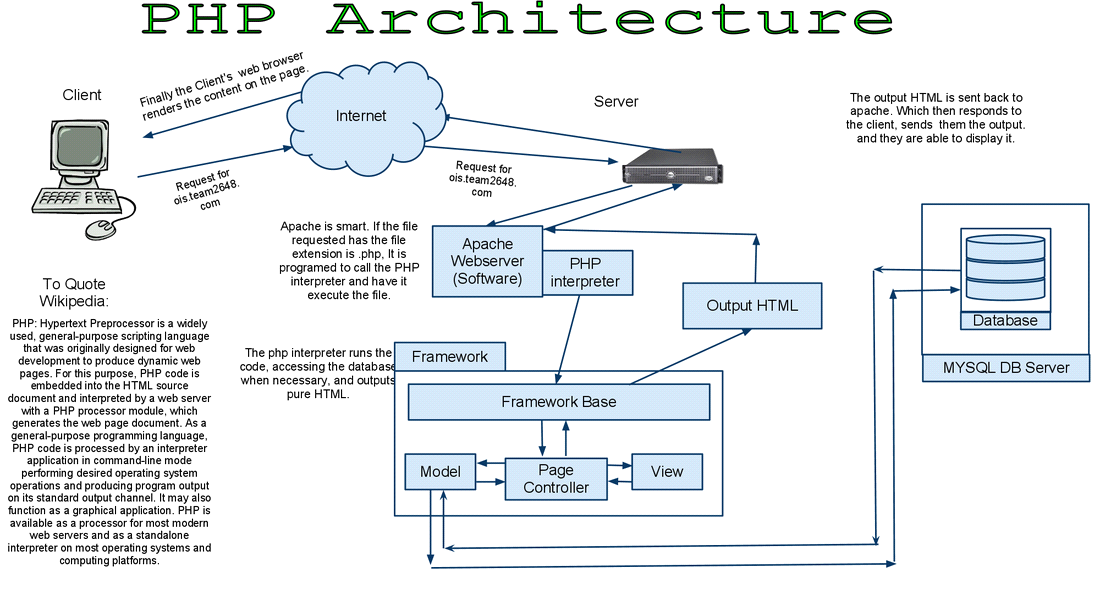
In [network computing](https://en.wikipedia.org/wiki/Computer_network), *front end* can refer to any hardware that optimizes or protects network traffic. It is called [application front-end hardware](https://en.wikipedia.org/wiki/Denial_of_service) because it is placed on the network's outward-facing front end or boundary. Network traffic passes through the front-end hardware before entering the network.

In [compilers](https://en.wikipedia.org/wiki/Compilers), the [front end](https://en.wikipedia.org/wiki/Compilers) translates a computer programming [source code](https://en.wikipedia.org/wiki/Source_code) into an [intermediate representation](https://en.wikipedia.org/wiki/Intermediate_representation), and the back end works with the intermediate representation to produce code in a computer output language. The back end usually optimizes to produce code that runs faster. The front-end/back-end distinction can separate the [parser](https://en.wikipedia.org/wiki/Parsing) section that deals with source code and the back end that [generates code and optimizes](https://en.wikipedia.org/wiki/Code_generation_(compiler)).

These days, front-end development refers to the part of the web users interact with. In the past, web development consisted of people who worked with Photoshop and those who could code HTML and CSS. Now, developers need a handle of programs like Photoshop and be able to code not only in HTML and CSS, but also JavaScript or jQuery, which is a compiled library of JavaScript.

Most of everything you see on any website is a mixture of HTML, CSS, and JavaScript, which are all controlled by the browser. For example, if you’re using Google Chrome or Firefox, the browser is what translates all of the code in a manner for you to see and with which to interact, such as fonts, colors, drop-down menus, sliders, forms, etc. In order for all of this to work, though, there has to be something to support the front-end; this is where the backend comes into play.

**2.1 Architecture of Front End user:**



**Figure 2.1:php Architecture**

**2.1.1 Architecture and Concepts**

The query cache plugin is implemented as a PHP extension. It is written in C and operates under the hood of PHP. During the startup of the PHP interpreter, it gets registered as a [mysqlnd](http://php.net/manual/en/book.mysqlnd.php) plugin to replace selected mysqlnd C methods. Hereby, it can change the behaviour of any PHP MySQL extension ([mysqli](http://php.net/manual/en/ref.mysqli.php), [PDO\_MYSQL](http://php.net/manual/en/ref.pdo-mysql.php), [mysql](http://php.net/manual/en/ref.mysql.php)) compiled to use the mysqlnd library without changing the extensions API. This makes the plugin compatible with each and every PHP MySQL application. Because existing APIs are not changed, it is almost transparent to use. Please, see the [mysqlnd plugin API description](http://php.net/manual/en/mysqlnd.plugin.php) for a discussion of the advantages of the plugin architecture and a comparison with proxy based solutions.

**2.2 Software & Tools Used:**

**1. PHP:-**

**2.2.1 INTRODUCTION OF PHP**

PHP is now officially known as “**PHP: Hypertext Preprocessor**”. It is a server-side scripting language usually written in an HTML context. Unlike an ordinary HTML page, a PHP script is not sent directly to a client by the server; instead, it is parsed by the PHP binary or module, which is server-side installed. HTML elements in the script are left alone, but PHP code is interpreted and executed. PHP code in a script can query databases, create images, read and write files, talk to remote servers – the possibilities is endless. The output from PHP code is combined with the HTML in the script and the result sent to the user’s web-browser, therefore it can never tell the user whether the web-server uses PHP or not, because the entire browser sees is HTML.

PHP’s support for Apache and MySQL further increases its popularity. Apache is now the most-used web-server in the world, and PHP can be compiled as an Apache module. MySQL is a powerful free SQL database, and PHP provides a comprehensive set of functions for working with it. The combination of Apache, MySQL and PHP is all but unbeatable.

That doesn’t mean that PHP cannot work in other environments or with other tools. In fact, PHP supports an extensive list of databases and web-servers. While in the mid-1990s it was ok to build sites, even relatively large sites, with hundreds of individual hard-coded HTML pages, today’s webmasters are making the most of the power of databases to manage their content more effectively and to personalize their sites according to individual user preferences.

**Reasons for using PHP**

There are some indisputable great reasons to work with PHP. As an open source product, PHP is well supported by a talented production team and a committed user community. Furthermore, PHP can be run on all the major operating systems with most servers.

**a)** **Learning PHP is easy**Basic is easy any interpreted language should be easy to learn. Since you are isolated from the system (no pointers to use, no memory to allocate). The other advantage that all modern interpreted languages share is good associative array constructs.

**b)** **Its Performance**

While we can build an application that serves millions of pages a day on a server, when we really look at the performance of the language it sucks. We are still orders of magnitude from real performance. Not only that, but since PHP is designed around a single process model our ability to share data structures or connection pool resources is left to native code libraries.

* **The low cost**

There are many languages which are available at very less cost. There are some languages which are available at very less cost like below:

* PHP
* C
* C++ etc

**d) It’s Open Source, We can modify it**

We can modify it if you need a hole in your head! Technically the point is that it’s an open source project and they release patches often. You’re point is that the community is actively working out the bugs. So, what any active language is doing this...

Unfortunately C, C++ and Perl have all “died” at this point and will pretty much remain static at their current functionality.

**Its Portability**

C is portable; it’s just the OS bits that aren’t. A lot PHP isn’t portable to Windows since people don’t use the OS abstractions to avoid some problems.

**It has interfaces to a large variety of database systems**

PHP supports a large variety of the database.

**Support available**

Online Support is available for using PHP.

**e)PHP Syntax**

You cannot view the PHP source code by selecting “View source” in the browser – you will only see the output from the PHP file, which is plain HTML. This is because the scripts are executed on the server before the result is sent back to the browser.

**Basic PHP Syntax**

A PHP scripting block always starts with **<?php** and ends with **?>**. A PHP scripting block can be placed anywhere in the document. On servers with shorthand support enabled you can start a scripting block with <? And end with ?>. However, for maximum compatibility, we recommend that you use the standard form (<?php) rather than the shorthand form.

A PHP file normally contains HTML tags, just like an HTML file, and some PHP scripting code.

**2.2.2 HTML**

**HTML** or **Hyper Text Markup Language** is the standard markup language used to create web pages.

HTML was created in 1991 by Tim Berners-Lee at CERN in Switzerland. It was designed to allow scientists to display and share their research.

HTML is written in the form of HTML elements consisting of *tags* enclosed in angle brackets(like <html>). HTML tags most commonly come in pairs like <h1> and </h1>, although some tags represent *empty elements* and so are unpaired, for example <img>. The first tag in a pair is the *start tag*, and the second tag is the *end tag* (they are also called *opening tags* and *closing tags*).

The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language rather than a programming language.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as Java Script which affect the behavior of HTML web pages.

HTML is descriptive markup language. Library of various markup languages is defined in various browsers.

**a) HTML Images - The <img> Tag and the Src Attribute**

In HTML, images are defined with the <img> tag.

The <img> tag is empty, which means that it contains attributes only, and has no closing tag.

To display an image on a page, you need to use the src attribute. Src stands for "source". The value of the src attribute is the URL of the image you want to display.

**Syntax for defining an image:**

<imgsrc="*url*" alt="*some\_text*">

**b) HTML FORMS**

HTML forms are used to pass data to a server.

|  |
| --- |
| The <form> tag is used to create an HTML form:  <form> . *input elements* . </form> |

An HTML form can contain input elements like text fields, checkboxes, radio-buttons, submit buttons and more. A form can also contain select lists, textarea, fieldset, legend, and label elements.

**c)** **Image tag (<img>) :**

To add an image to an HTML document, we just need to include an <IMG> tag with a

reference to the desired image. The <IMG> tag is an empty element i.e. it doesn’t require a

closing tag and we can use it to include from small icons to large images.

**Syntax: <imgsrc=”URL” alt=”alternative text”>**

**d) HTML Lists :**

|  |  |
| --- | --- |
| An ordered list:   * The first list item * The second list item * The third list item | An unordered list:   * List item * List item * List item |

**2.2.3 HTML 5**

HTML5 will be the new standard for HTML. The previous version of HTML, HTML 4.01,

came in 1999. The web has changed a lot since then. HTML5 is still a work in progress.

However, the major browsers support many of the new HTML5 elements and APIs.

HTML5 is cooperation between the World Wide Web Consortium (W3C) and the Web

Hypertext Application Technology Working Group (WHATWG).

WHATWG was working with web forms and applications, and W3C was working with

XHTML 2.0. In 2006, they decided to cooperate and create a new version of HTML.

Some rules for HTML5 were established:

a) New features should be based on HTML, CSS, DOM, and JavaScript

b) Reduce the need for external plug-ins (like Flash)

c) Better error handling

d) More markup to replace scripting

e) HTML5 should be device independent

f) The development process should be visible to the public

**2.2.4 CSS**

**CSS tutorial** or CSS 3 tutorial provides basic and advanced concepts of CSS technology. Our CSS tutorial is developed for beginners and professionals. The major points of CSS are given below:

* CSS stands for Cascading Style Sheet.
* CSS is used to design HTML tags.
* CSS is a widely used language on the web.
* HTML, CSS and JavaScript are used for web designing. It helps the web designers to apply style on HTML tags.

**Cascading Style Sheets** (**CSS**) is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to style web pages and user interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. CSS is a cornerstone specification of the web and almost all web pages use CSS style sheets to describe their presentation.

CSS is designed primarily to enable the separation of document content from document presentation, including elements 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 pages to share formatting, and reduce complexity and repetition in the structural content (such as by allowing for table less web design).

CSS can also allow the same markup page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille-based, tactile devices. It can also be used to allow the web page to display differently depending on the screen size or device on which it is being viewed. While the author of a document typically links that document to a CSS file, readers can use a different style sheet, perhaps one on their own computer, to override the one the author has specified.

With plain HTML you define the colors and sizes of text and tables throughout your pages. If

you want to change a certain element you will therefore have to work your way through the

document and change it. With CSS you define the colors and sizes in "styles". Then as you

write your documents you refer to the styles. Therefore: if you change a certain style it will

change the look of your entire site. Another big advantage is that CSS offers much more

detailed attributes than plain HTML for defining the look and feel of your site.

**2.2.5 JAVASCRIPT**

**JavaScript** (**JS**) is a dynamic computer programming language. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. It is also being used in server-side network programming (with Node.js), game development and the creation of desktop and mobile applications.

JavaScript is a prototype-based scripting language with dynamic typing and has first-class functions. Its syntax was influenced by C. JavaScript copies many names and naming conventions from Java, but the two languages are otherwise unrelated and have very different semantics. The key design principles within JavaScript are taken from the Self and Scheme programming languages. It is a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles.

The application of JavaScript in use outside of web pages—for example, in PDF documents, site-specific browsers, and desktop widgets—is also significant. Newer and faster JavaScript VMs and platforms built upon them (notably Node.js) have also increased the popularity of JavaScript for server-side web applications. On the client side, JavaScript was traditionally implemented as an interpreted language but just-in-time compilation is now performed by recent (post-2012) browsers.

JavaScript was formalized in the ECMA Script language standard and is primarily used as part of a web browser (client-side JavaScript). This enables programmatic access to objects within a host environment.

JavaScript is the most popular programming language in the world.

It is the language for HTML, for the Web, for computers, servers, laptops, tablets, smart phones, and more.

You can use JavaScript to:

a) Change HTML elements

* Delete HTML elements
* Create new HTML elements
* Copy and clone HTML elements.

**CHAPTER-3: BACK END**

In a previous blog, we talked about how web programmers are concerned with launching websites, updates, and maintenance, among other things. All of that works to support the front-end of the website. The back-end has three parts to it: server, application, and database.

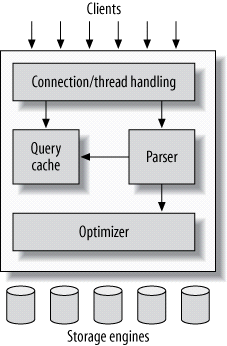
To better explain how all of this works, let’s use the example of a customer trying to purchase a plane ticket using a website. Everything that the customer sees on the webpage is the front-end, as we have explained before, but once that customer enters all of his or her information, such as their name, billing address, destination, etc, the web application stores the information in a database that was created previously on the server in which the website is calling for information.

The web application creates, deletes, changes, renames, etc items in the database. For example, when a customer purchases a ticket, that creates an item in the database, but when they have a change in their order or they wish to cancel, the item in the database is changed.

.In short, when a customer wants to buy a ticket, the backend operation is the web application communicating with the server to make a change in a database stored on said server. Technologies like PHP, Ruby, Python, and others are the ones backend programmers use to make this communication work smoothly, allowing the customer to purchase his or her ticket with ease.

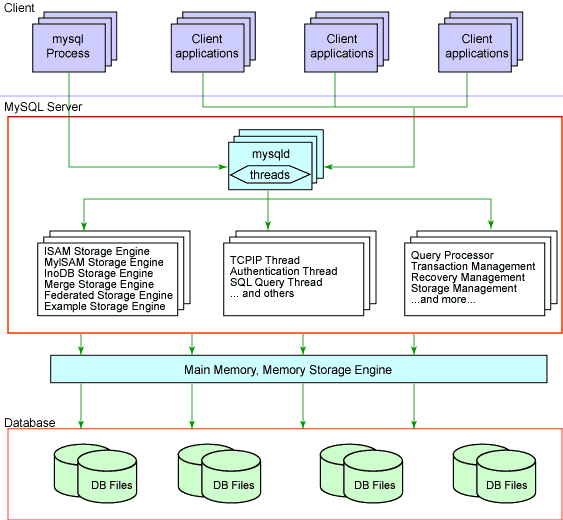
**3.1 MySQL’s Logical Architecture**

The topmost layer contains the services that aren’t unique to MySQL. They’re services most network-based client/server tools or servers need: connection handling, authentication, security, and so forth.

.

**Figure 3.1:MySQL Logical Architecture**

The third layer contains the storage engines. They are responsible for storing and retrieving all data stored “in” MySQL. Like the various filesystems available for GNU/Linux, each storage engine has its own benefits and drawbacks. The server communicates with them through the *storage engine API*. This interface hides differences between storage engines and makes them largely transparent at the query layer. The API contains a couple of dozen low-level functions that perform operations such as “begin a transaction” or “fetch the row that has this primary key.” The storage engines don’t parse SQL[[4](https://www.safaribooksonline.com/library/view/high-performance-mysql/9781449332471/ch01.html)] or communicate with each other; they simply respond to requests from the server.



**Figure 3.2: Storage Engine API**

**3.2 Softwares and tools used:**

**3.2.1 My Sql:**

**Introduction:**

The database has become an integral part of almost every human's life. Without it, many things we do would become very tedious, perhaps impossible tasks. Banks, universities, and libraries are three examples of organizations that depend heavily on some sort of database system. On the Internet, search engines, online shopping, and even the website naming convention would be impossible without the use of a database. A database that is implemented and interfaced on a computer is often termed a database server.  
 One of the fastest SQL (Structured Query Language) database servers currently on the market is the MySQL server, developed by T.c.X. DataKonsultAB. MySQL, available for download at [www.mysql.com](http://www.mysql.com/), offers the database programmer with an array of options and capabilities rarely seen in other database servers. MySQL is free of charge for those wishing to use it for private and commercial use. Those wishing to develop applications specifically using MySQL should consult MySQL's licensing section, as there is charge for licensing the product.

**These capabilities range across a number of topics, including the following:**

a) Ability to handle an unlimited number of simultaneous users.

b) Capacity to handle 50,000,000+ records.

c) Very fast command execution, perhaps the fastest to be found on the market.

d)Easy and efficient user privilege system.

However, perhaps the most interesting characteristic of all is the fact that it's free. That's right, T.c.X offers MySQL as a free product to the general public.

**Reasons to Use MySQL**

**a) Scalability and Flexibility**

The MySQL database server provides the ultimate in scalability, sporting the capacity to handle deeply embedded applications with a footprint of only 1MB to running massive data warehouses holding terabytes of information. Platform flexibility is a stalwart feature of MySQL with all flavors of Linux, UNIX, and Windows being supported.

**b) High Performance**

A unique storage-engine architecture allows database professionals to configure the MySQL database server specifically for particular applications, with the end result being amazing performance results.

**C) High Availability**

Rock-solid reliability and constant availability are hallmarks of MySQL, with customers relying on MySQL to guarantee around-the-clock uptime. MySQL offers a variety of high-availability options from high-speed master/slave replication configurations, to specialized Cluster servers offering instant failover, to third party vendors offering unique high-availability solutions for the MySQL database server.

**d) Robust Transactional Support**

MySQL offers one of the most powerful transactional database engines on the market. Features include complete ACID (atomic, consistent, isolated, durable) transaction support, unlimited row-level locking, distributed transaction capability, and multi-version transaction support where readers never block writers and vice-versa.

**e) Web and Data Warehouse Strengths**

MySQL is the de-facto standard for high-traffic web sites because of its high-performance query engine, tremendously fast data inserts capability, and strong support for specialized web functions like fast full text searches.

**f) Strong Data Protection**

Because guarding the data assets of corporations is the number one job of database professionals, MySQL offers exceptional security features that ensure absolute data protection. In terms of database authentication, MySQL provides powerful mechanisms for ensuring only authorized users have entry to the database server, with the ability to block users down to the client machine level being possible.

**g) Management Ease**

MySQL offers exceptional quick-start capability with the average time from software download to installation completion being less than fifteen minutes. This rule holds true whether the platform is Microsoft Windows, Linux, Macintosh, or UNIX.

**PHP Main Features of MySQL**

* Tested with a broad range of different compilers.
* Works on many different platforms.
* The MySQL Server design is multi-layered with independent modules.
* Fully multi-threaded using kernel threads. It can easily use multiple CPUs if they are available.
* Provides transactional and non-transactional storage engines.
* Uses very fast B-tree disk tables with index compression.
* Relatively easy to add other storage engines. This is useful if you want to provide an SQL interface for an in-house database.
* A very fast thread-based memory allocation system.
* Very fast joins using an optimized one-sweep multi-join.
* In-memory hash tables, which are used as temporary tables.
* SQL functions are implemented using a highly optimized class library and should be as fast as possible. Usually there is no memory allocation at all after query initialization.
* The server is available as a separate program for use in a client/server networked environment.

**CHAPTER-4: MINIMUM HARDWARE SPECIFICATION**

**4.1 Hardware Requirement**

Processor : Intel Core Duo 2.0 GHz or more

RAM : 1 GB or More

Harddisk : 80GB or more

Monitor : 15” CRT, or LCD monitor

Keyboard : Normal or Multimedia

Mouse : Compatible mouse

**4.2 Software Requirement**

Front End : Visual Basic 2005 Express edition

With Sql Server Compact Edition

Microsoft SDK 2.0

Or

Visual Basic 2008 Express edition

With Sql Server Compact Edition

Microsoft SDK 3.0

Back End : MS Sql Server

Operation System : Windows 7 with server pack 2

Or

Windows8.1

**CHAPTER-5.COMPLETE DFD & E-R DAIGRAM**

**5.1 INTRODUCTION ER DIAGRAM**

The entity-relationship data model is based on a perception of a real world that consists of a collection of basic objects called entities and of relationships among these objects. An entity is an “object” in the real world that is distinguishable from other objects. **For e.g.** each customer is an entity and rooms can be considered to be entities. Entities are described by a set of attributes. **For e.g.** the attributes Room no. and Room type describes a particular Room in a hotel. The set of all entities of the same type and the set of all relationships of the same type are termed as an entity set and relationship set respectively.

The logical structure of a database can be expressed graphically by an E- R diagram consists of the following major components:

**5.1.1 Entity**

***Entity***

An entity is an “object” in the real world that is distinguishable from all other objects. An entity set is a set of entities of the same type that share the same attributes.

**5.1.2 Weak Entity**

An entity set that may not have sufficient attributes to form a primary key is termed as a weak entity set.

**Entity**

**5.1.3 Attribute**

Attributes are descriptive properties possessed by each member of an entity set.

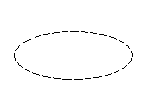
**5.1.4 Key attribute**

A key attribute is the unique, distinguishing characteristic of the entity. For example, Guest ID might be the guest’s key attribute.

**5.1.5 Multivalued attribute**

In an instance where an attribute has a set of values for a specific entity is called multivalued attribute.

**5.1.6 Derived attribute**



In these attributes the value can be derived from the values of other related attributes.

**5.1.7 Relationships**

A relationships an association among several entities.

**Relationship**

**For e.g.,** we can define a relationship that associates customer Jon with Room 142.This relationship specifies that Jon is a customer with Room No.142.

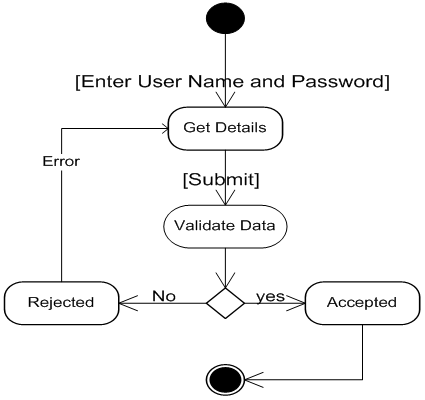
**5.1.8 Recursive Relationship**

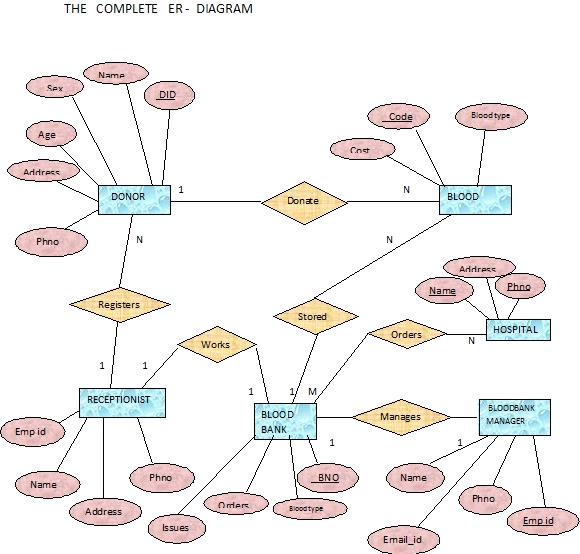
In some cases, entities can be self-linked. For example, Employees can supervise other employees in a hotel.

**Employee**

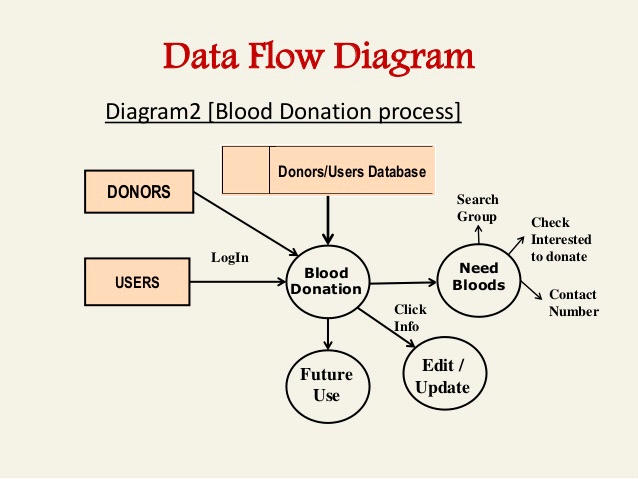
**While using E-R diagrams, we can follow certain guidelines, which are as follows:**

* Unnecessary attributes should not be introduced.
* Entities should be merged with common attributes.
* A complex entity should be simplified by decomposing a complex attribute into sub attributes.
* We should generalize or specialize wherever possible and appropriate. Generalization is the result of taking the union of several lower entity sets to produce higher- level entity set.





**Figure 5.1: E-R Diagram**



**Figure 5.2: Data Flow diagram**

**6.COMPLETE SNAPSHOTS**

**6.1 Home page:-** This is the home page or the main page of a blood bank management system. This is the main page of a client side. This page define all about related to project. This page also includes the galary of the camps.

Blood Donor

Equipments

Stick

Blood Recipient

Blood collection

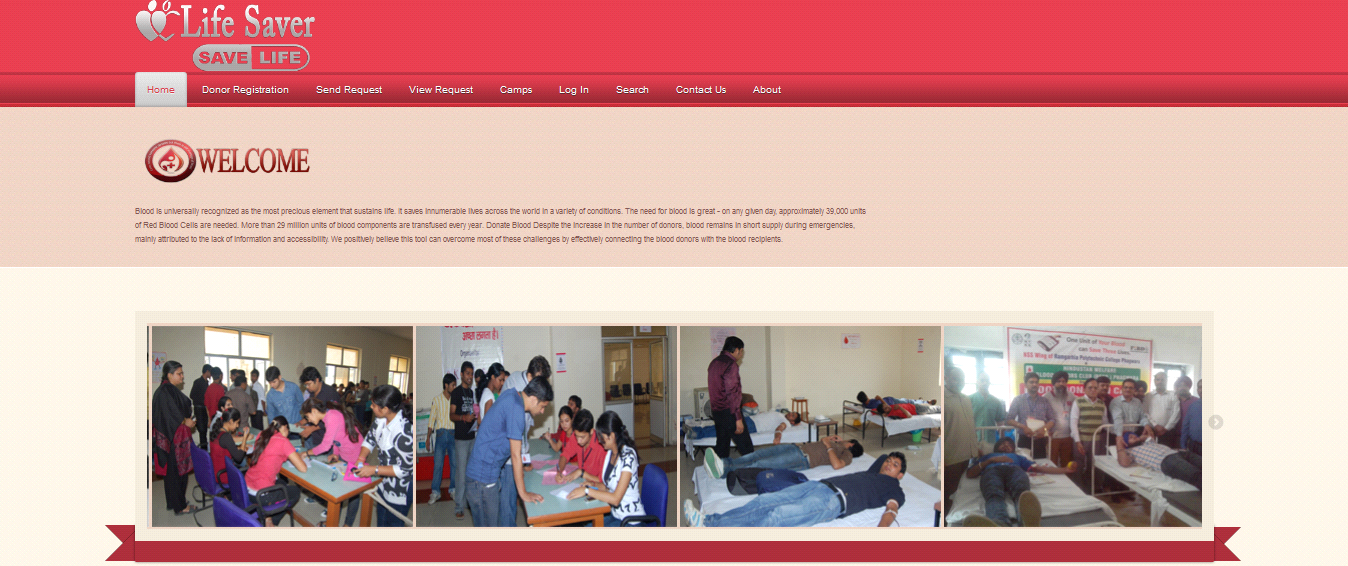
Camp

Stock details

blood bank system project Reports

Blood issued

Blood bank system project



**Figure 6.1:Home Page of life saver**

**6.2 Registration**

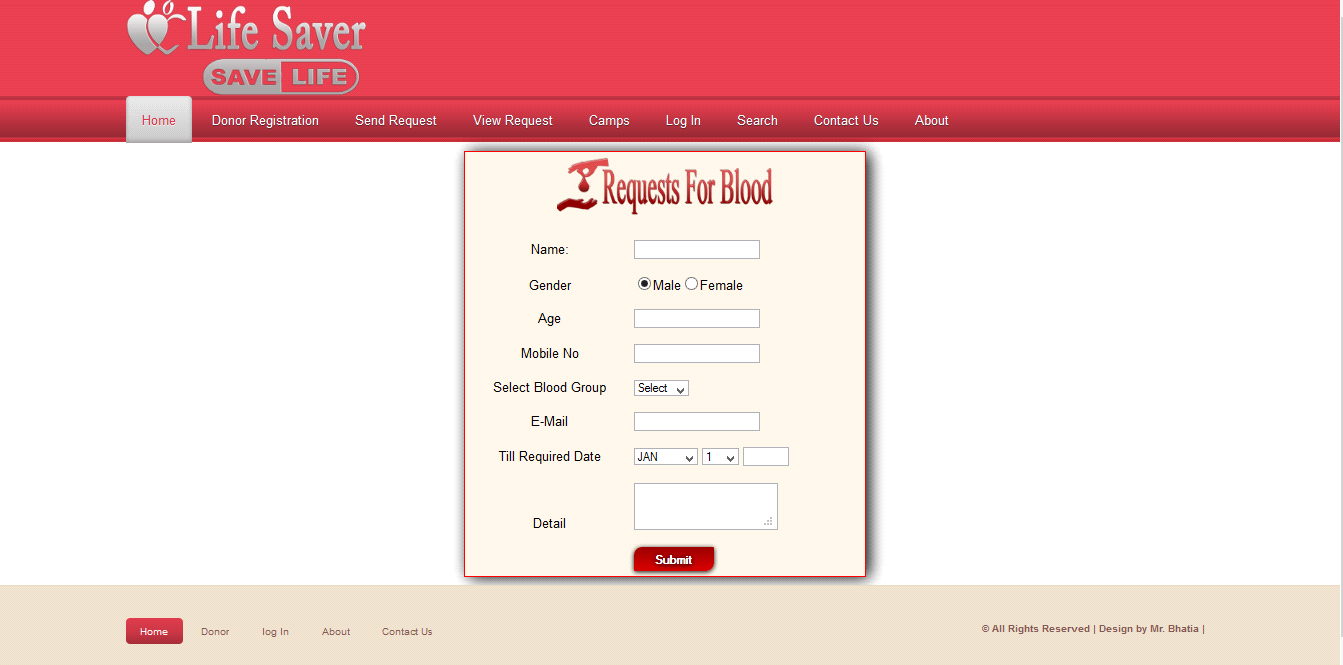
Registration page includes the information of the donor who want to register. Donor can register the account by clicking on new register. He/she can add the account for the further enquiry of the blood donation.



**Figure 6.2:Registration page of Life Saver**

**6.3 Request For Blood**

Request for blood page includes the information of the donor who want to register. Donor can register the account by clicking on new register. He/she can send requests for the further enquiry of the blood donation.

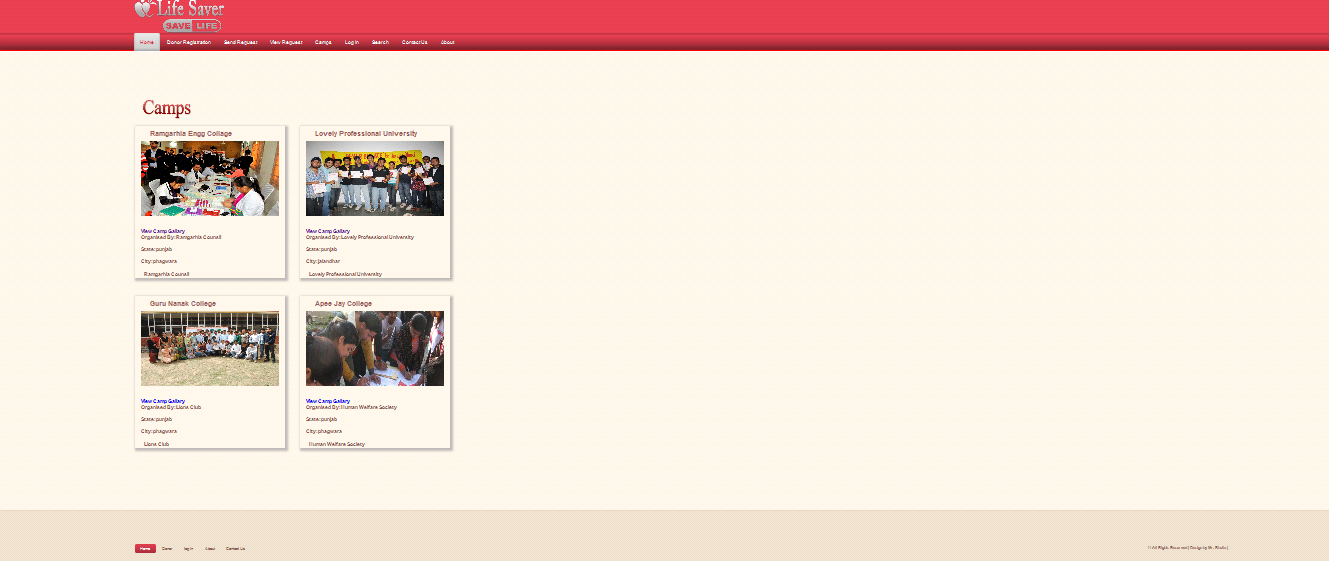


**Figure 6.3:Request for Blood page**

**6.4 Camps**

Camp page includes the information about camps, this camps organised the blood donation camp. Donor can register the account by clicking on new register.

Request for blood page includes the information of the donor who want to register. Donor can register the account by clicking on new register. He/she can send requests for the further enquiry of the blood donation.



**Figure 6.4:Camps of Blood group**

**6.5 Donor Login**

It is  login menu, I have set a default E-mail and password for it. This mail id and password are store in a binary file inside this project. For Password security I have replace the password's alpha numeric character to special symbol, Dollar ($) symbol.

If you will enter the correct mail id and password, It will allow you to move to next menu otherwise it will show you error message and ask you to invalid mail id and password.

Request for blood page includes the information of the donor who want to register. Donor can register the account by clicking on new register. He/she can send requests for the further enquiry of the blood donation.



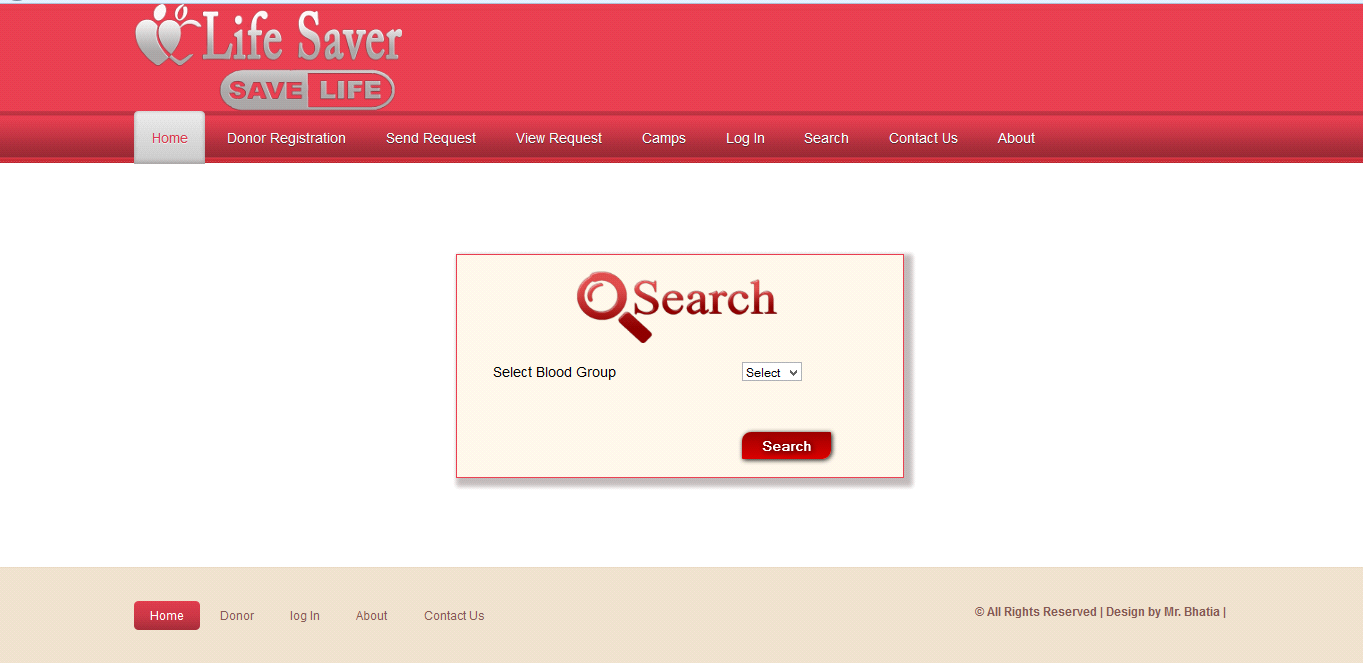
**Figure 6.5:DONOR Log In**

**6.6 Search**

Search button is used for search the donations of blood for a different defferent blood groups.

Blood groups name

* A
* A+
* B
* B+
* AB+
* AB\_
* O+
* O-



**Figure 6.6:Search for Blood**

**6.7 Contact**

In contact page any one who wants to need a blood or gaining a information about this system then he/she can contact with us.

Person will get the blood immediately he/she requested for the particular blood group he/she

Has requested.



**Figure 6.7:Contact Us**

**6.8 Donor Pannel**

This page is the the welcome page of the donor panel. In this page include all the module related to Donor like :

Change password

Update profile

Blood donated

View donation

View requested

Logout



**Figure 6.8:Welcome To Donor Pannel**

**6.9 Change Password**

Above snap short describe how the donor change our password .This is only used by a donor.



**Figure 6.9:Change Password**

**6.10 Update Profile**

Above snap short describe how the donor update our profile .This is only used by a donor.



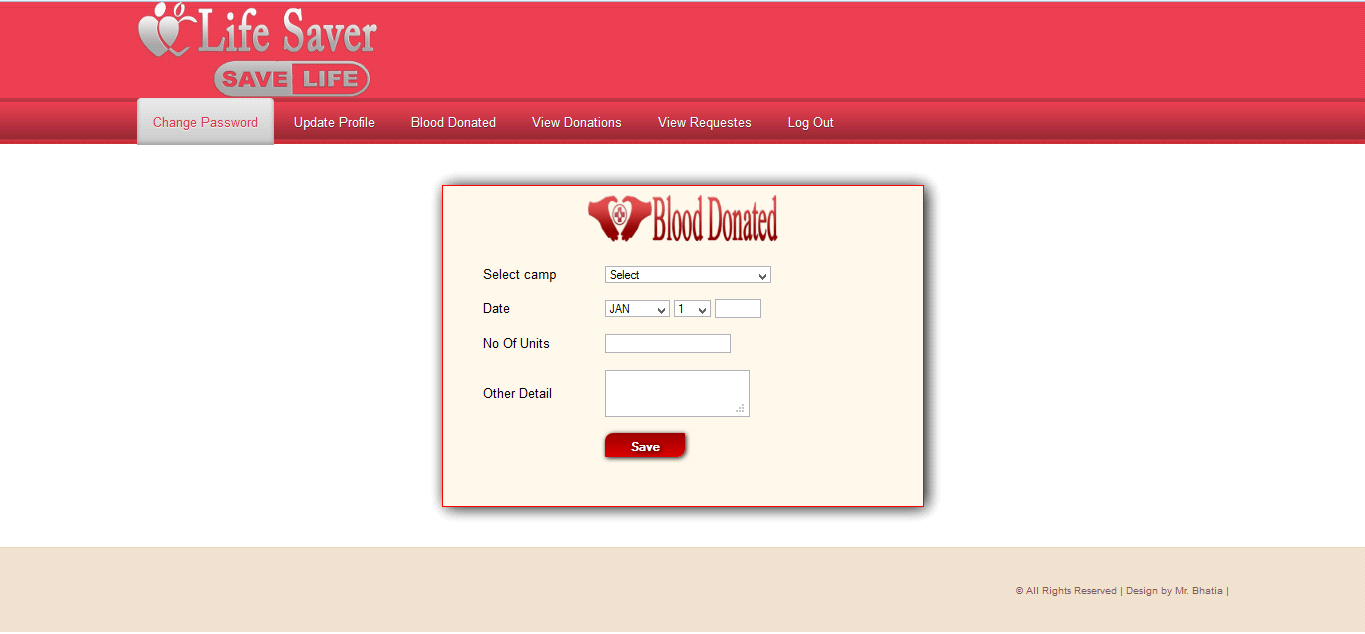
**Figure 6.10:update Profile**

**6.11 Blood Donated**

Above snap short describe about donation from donor.This is only used by a donor.

Person will get the blood immediately he/she requested for the particular blood group he/she

has requested.



**Figure 6.11:Blood Donated**

**6.12 Admin Pannel**

This is the admin side of the project, shows all the admin page like addition , updation, deletion of the user, city,state,camps etc.

Person will get the blood immediately he/she requested for the particular blood group he/she

has requested.



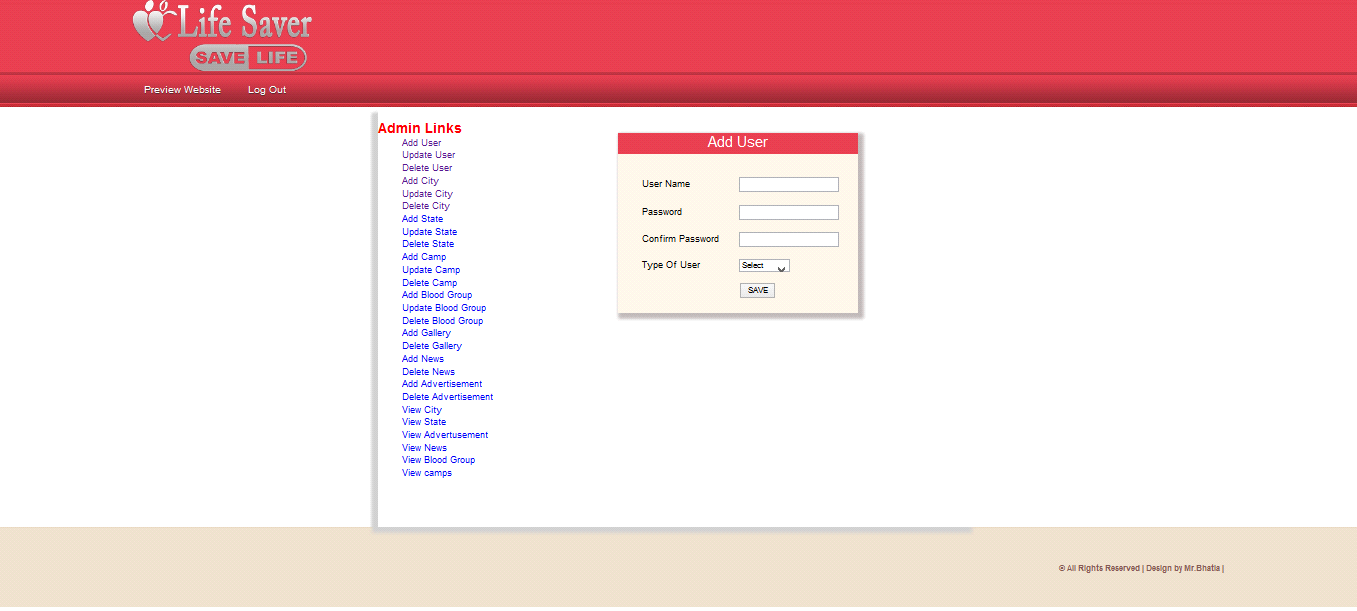
**Figure 6.12:Admin Pannel**

**6.13 Add user**

Show the addition of user.

Person will get the blood immediately he/she requested for the particular blood group he/she

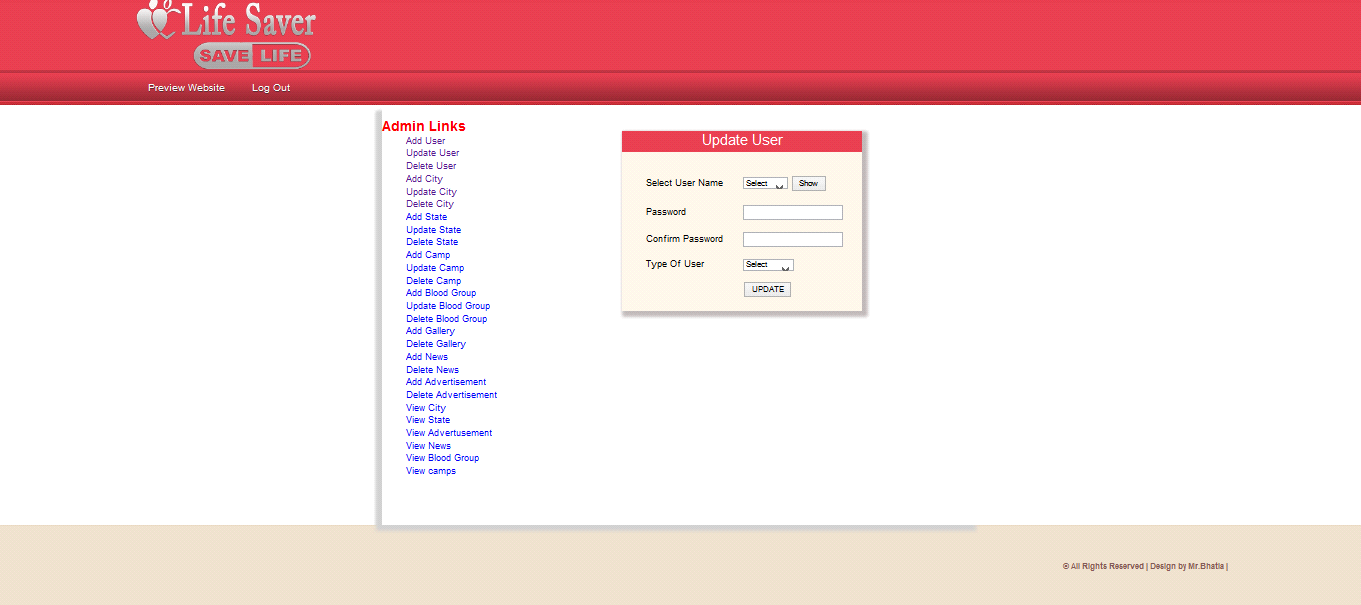
has requested.



**Figure 6.13: Add User**

**6.14 Update User**

Show the updation of user.



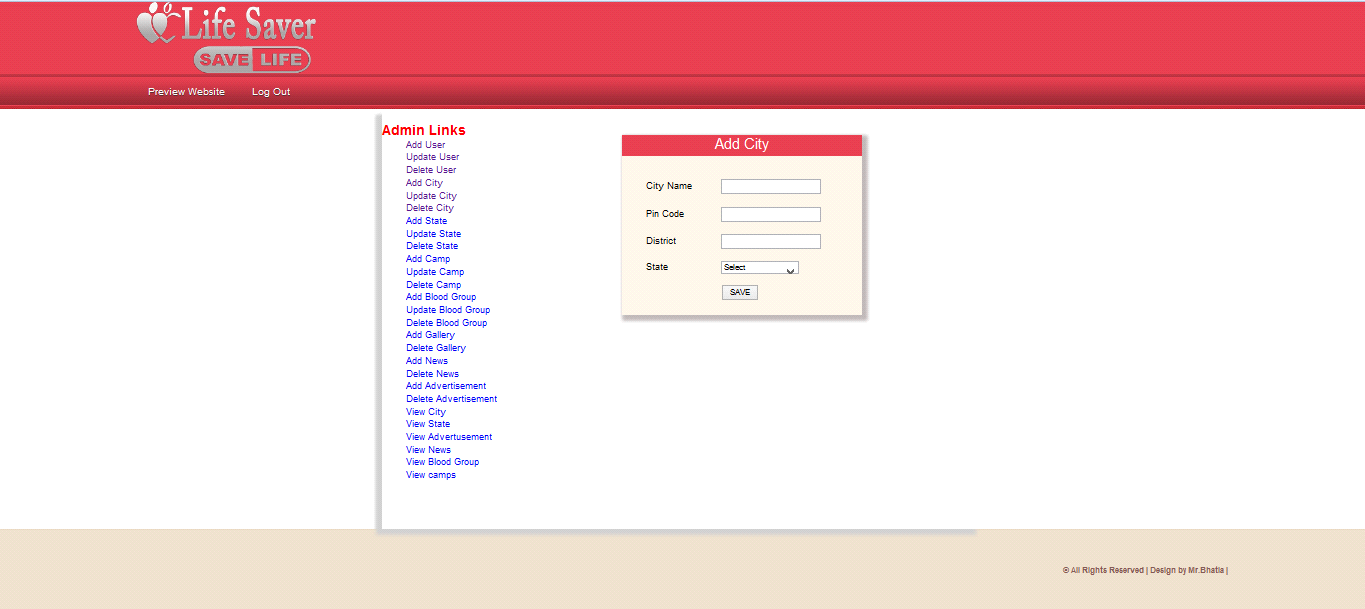
**Figure 6.14:Update User**

**6.15 Add City**

Show the addition of city.

Person will get the blood immediately he/she requested for the particular blood group he/she

has requested.



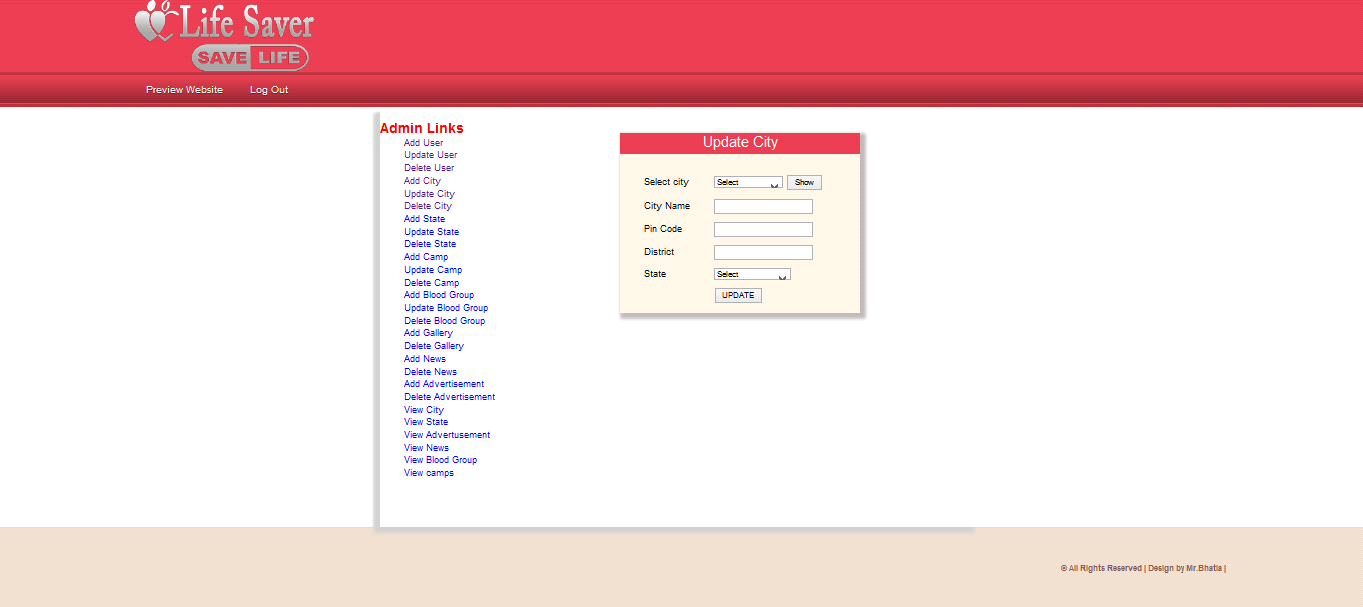
**Figure 6.15:Add City**

**6.16 Update City**

Show the updation of city.

Person will get the blood immediately he/she requested for the particular blood group he/she

has requested.



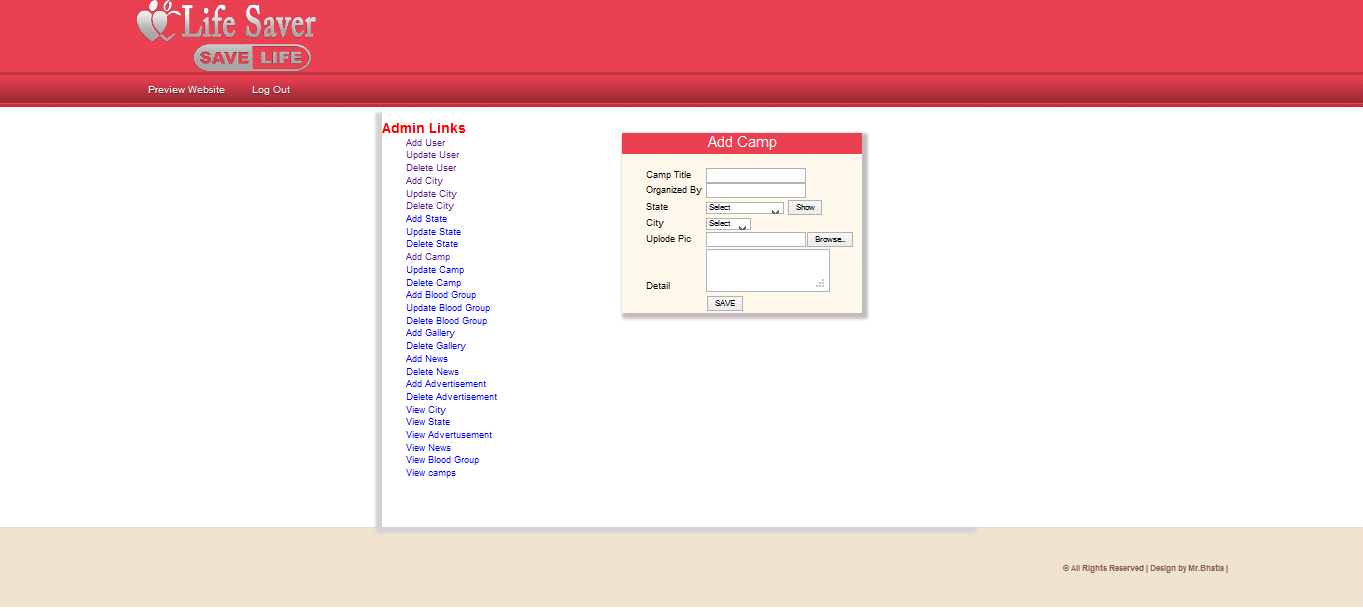
**Figure 6.16:Update City**

**6.17 Add Camp**

Show the addition of camp.

Person will get the blood immediately he/she requested for the particular blood group he/she

has requested.



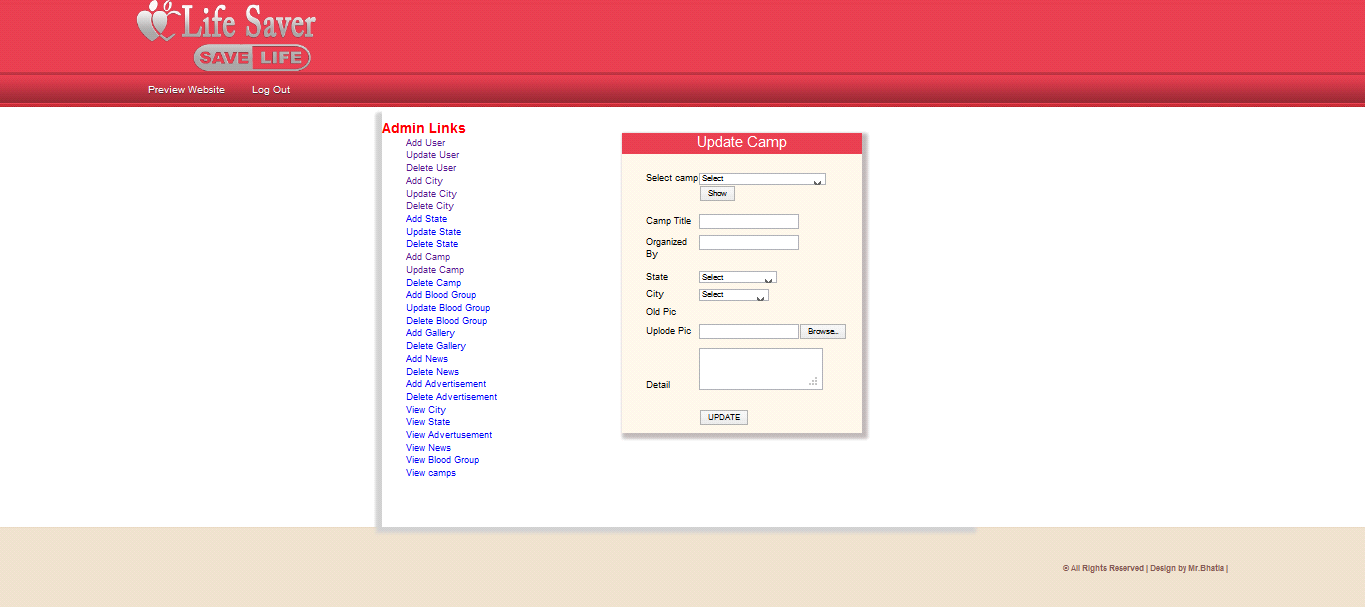
**Figure 6.17:Add Camp**

**6.18 Update Camp**

Show the updation of camp.

Person will get the blood immediately he/she requested for the particular blood group he/she

has requested.



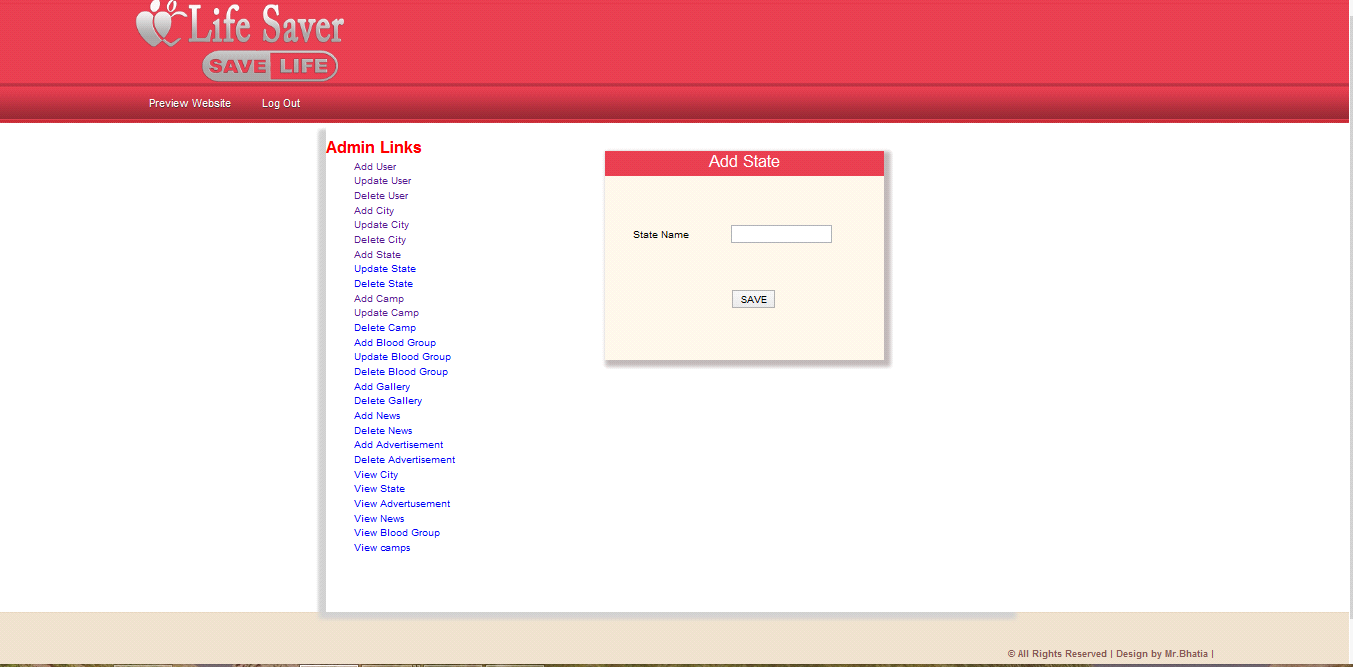
**Figure 6.18:Update Camp**

**6.19 Add State**

Show the addition of state.

Person will get the blood immediately he/she requested for the particular blood group he/she

has requested.



**Figure 6.19:Add State**

**6.20 Update State**

Show the updation of state.

Person will get the blood immediately he/she requested for the particular blood group he/she

has requested.



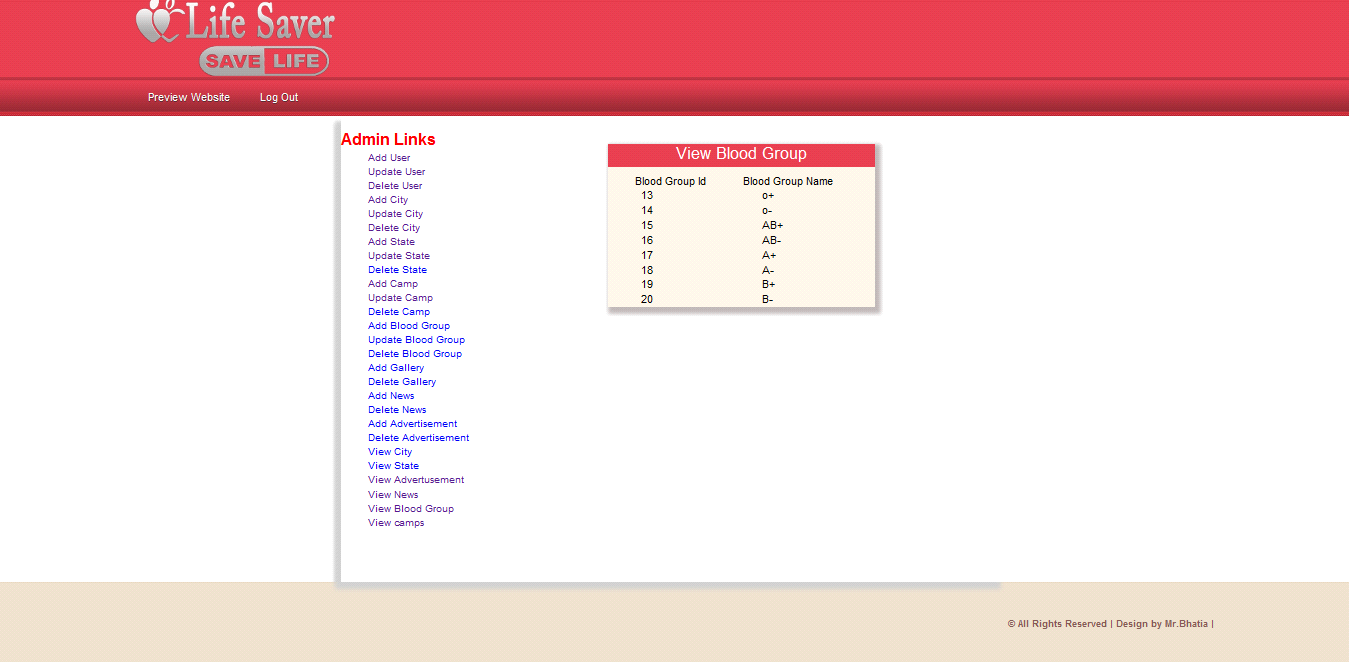
**Figure 6.20:Update State**

**6.21 View Blood Group**

This snapshort shows all the blood group. This shows blood group name and id**.**

Person will get the blood immediately he/she requested for the particular blood group he/she

has requested.



**Figure 6.21:View Blood Group**

**CHAPTER-7: SUMMARY &CONCLUSION**

With the theoretical inclination of our syllabus it becomes very essential to take the atmost advantage of any opportunity of gaining practical experience that comes along. The building blocks of this Major Project ”BLOOD BANK Management System” was one of these opportunities. It gave us the requisite practical knowledge to supplement the already taught theoretical concepts thus making us more competent as a computer engineer.

* The planning that goes into implementing a project.
* The importance of proper planning and an organized methodology.
* The key element of team spirit and co-ordination in a successful project.

The project also provided us the opportunity of interacting with our teachers and to gain from their best experience.

**CHAPTER-8 : FUTURE RECOMMENDATION**

**BLOOD BANK** MANAGEMENT is a software application to built such a way that it should suits for all type of blood banks in **future**.

One important future scope is availability of location based blood bank details and extraction of location based donor’s detail, which is very helpful to the acceptant people. All the time the network facilities cannot be use. This time donor request does not reach in proper time, this can be avoid through adding some message sending procedure this will help to find proper blood donor in time. This will provide availability of blood in time.

**CHAPTER-9 : REFERENCES**

**PHP Manual** www.php.net/

https://www.google.com

http://www.w3schools.com

http://www.indianbloodgroup.com

.