# VISVESVARAYA INSTITUTE OF TECHNOLOGY
## Jnana Sangama, Belagavi-590010



MINI PROJECT REPORT

ON
## "BLOOD DONATION MANAGEMENT SYSTEM"

Submitted in partial fulfillment for the requirements for the fifth semester curriculum

## BACHELOR OF ENGINEERING
## IN
## COMPUTER SCIENCE AND ENGINEERING

For the Academic year 2018-2019

Submitted by:

**ABHIJNA S HEBBAR ( 1MV16CS002)**
**NAMRATHA A ( 1MV16CS056)**

Project carried out at:
**Sir M. Visvesvaraya Institute of Technology**
Bengaluru-562157

Under the guidance of:
**Mrs. Sapna R**
Asst. Professor, Department of CSE
Sir M. Visvesvaraya Institute of Technology, Bengaluru



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**
**SIR. M VISVESVARAYA INSTITUTE OF TECHNOLOGY**
HUNASAMARANAHALLI, BENGALURU-562157

# SIR M. VISVESVARAYA INSTITUTE OF TECHNOLOGY
## BENGALURU -562157

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



## <u>CERTIFICATE</u>

It is certified that the project work entitled "**BLOOD DONATION MANAGEMENT SYSTEM** " is a bonafide work carried out **by Abhijna S Hebbar (1MV16CS002) , Namratha A (1MV16CS056)** in partial fulfilment for the requirements of mini project for the V semester curriculum  Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belagavi during the year 2018-2019. It is certified that all corrections and suggestions indicated for Internal Assessment have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the course of Bachelor of Engineering.

Name & Signature of Guide                                   Name & Signature of HOD

**Mrs. Sapna R**                                                    **Dr.G.C. Bhanuprakash**
Asst. Prof & Internal Guide                              Prof and HOD Dept of CSE
Dept. Of CSE, Sir MVIT                                    Sir MVIT
Bengaluru - 562157                                            Bengaluru -562157

External Examination:

Name of the Examiners                                      Signature with date

1)

2)

# ACKNOWLEDGEMENT

# **DECLARATION**

       We hereby declare that the entire mini project work embodied in this dissertation has been carried out by us and no part has been submitted for any degree or diploma of any institution previously.

Place: Bengaluru

Date:

Signature of Students:

ABHIJNA S HEBBAR (1MV16CS002)

NAMRATHA A (1MV16CS056)

# ABSTRACT

Blood Donation Management System (BDMS) 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.

# TABLE OF CONTENTS

# Chapter 1: <u>Introduction:</u>

## 1.1 <u>Project Profile</u>

- **Name of the project:** Blood Donation Management System

- **Object Description:** The project generally refers to a system which maintains all the information pertaining to blood donors, different blood groups available in each blood bank and help them manage in a better way.

- **Front End:** PHP

- **Back End:** MySQL Database

## 1.2 <u>Purpose</u>

The main objective of the Project on Blood Donation Management System is to manage the details of Blood Camps, Blood Group, Blood. It manages all the information about Blood Camps, Users, Donors. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the Blood Camps, Blood, Blood Group. It tracks all the details about the Blood Group, Blood Cells, Blood Stock.

## 1.3 <u>Project Scope</u>

Without an Online registration system, managing and maintaining the details of all the users, donors, blood and blood camps is a tedious job for any organization.

This system will store all the details of the users like their profile picture, blood type including their personal information and the contact details of all the users.

**<u>Signup module:</u>** Signup module will helps the users who are signing up for the first time. If they have already signed in with valid information, they can directly login.

**<u>Login Module:</u>** Login module will help in authentication of user accounts. Users who have valid login id and password can only login and register.

**Auto populate Module:** Auto populate module will help in getting the proper count of blood units in each blood type and the count of units of blood donated by each donor .

## What contribution would the project make?

This in an era of information technology where automation of each and every activity is gaining importance. This is better than offline struggles that the people involved in the events must undergo.

## Computerized vs. Manual registration System:

- ➢ Time saving
- ➢ Avoidance of paper
- ➢ Allows neat handling of data rather than error prone records
- ➢ Accuracy
- ➢ Easy and fast retrieval of information.
- ➢ Access of any information individually.
- ➢ Work becomes very speedy
- ➢ Easy to update information.

## 1.4  Definitions, Acronyms and Abbreviations:

**Personal details:** Details of the users such as name, phone number, email id, address.

**Contact details:** Details of contact associated with the users and donors.

**WWW:** World Wide Web

**BDMS:** Blood Donation Management System

**MySQL:** A RDBMS based on SQL which is used for adding, removing and modifying information in the database.

**RDBMS:** Relational Database Management System

## 1.5 <u>Overview</u>

The purpose of this document is to present a detailed description of the BBMS. It will explain the purpose and features of the software, the interfaces of the software, what actually the software does, the constraints under which the software works and how it reacts to external stimuli. This document is intended for the end users.

This is based on the fact that there is a need to upgrade the system with a computer based information system which is **Blood Donation Management System**.

# Chapter 2: <u>About the System:</u>

## 2.1    <u>Software Requirement Specification</u>

### ❖ **Introduction**

This Software Requirements Specification (SRS) document is intended to give a complete overview of Blood Donation Management System Project(working title). The SRS document details all features upon which the system has currently decided with reference to the manner and importance of their implementation.

### ❖ **Product Perspective**

This product and application is newer which provides the user a new utility in their role as a user or donor or both. The users will get the details of all the donors by just logging to the application and streaming on the desired blood type.

## 2.2 <u>Performance Requirements</u>

Better performance will lead to better operating environment. For better environment the user needs a high speed internet so that the uploading of details and registration will be done better.

## 2.3  <u>Security Requirements</u>

The login details must be kept confidential so that other user may not login using other's id and password.

## 2.4    <u>Feasibility Study</u>

## **Feasibility Analysis:**

- o A feasibility study is short focused, which aims to answer a number of questions:
- o Does the system contribute to the overall objective of the organization.

- o Can the system be implemented using the current technology and within given schedule constraints?
- o Can the system be integrated with system which is already in place?

## Economic Feasibility:

The project is economically feasible as it only requires a desktop with it's operating system. The users must be able to connect to internet and this would be the only cost incurred on the project.

## Technical Feasibility:

To develop this desktop application, an internet connection, a database server, a web server and software are required. The current project is technically feasible as this desktop application was successfully deployed on our desktop.

## Behavioral Feasibility:

The desktop application is behaviorally feasible since it requires no technical guidance, all the modules are user friendly and execute in a manner they were designed to.

## 2.5    Project Plan

At the beginning of the project, we scheduled meeting time for the group to discuss on the design and implementation of the software and what language to use in writing the software. We had several meetings to this effect. We then developed a time-line for the project and the estimated time. We also pondered on a suitable name to give to the project.

The workload was then divided among us and we worked on the parts of the code. We kept in touch with each other and whenever we had difficulties, we asked each other questions. On some occasions, we had to pretend we were the customer so as to try to figure out some of the things that user would desire, such as the friendliness of the user interface and ease of navigation through the software.

# Chapter 3: <u>DESIGN</u>

### 3.1  E-R Diagram

Entity-Relationship diagram is a detail & logical representation of entities and data elements for an organization. This technique is used in database that helps in an enterprise are related to each other. There are 3 types of E-R diagram:

1. **one to one :**

    It is a one to one relationship is an association between 2 entities.
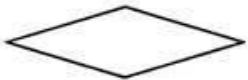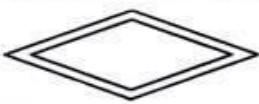
2. **one to many:**

    One-to-many relationship exists when one entity related to one or more entity.

3. **Many to many:**

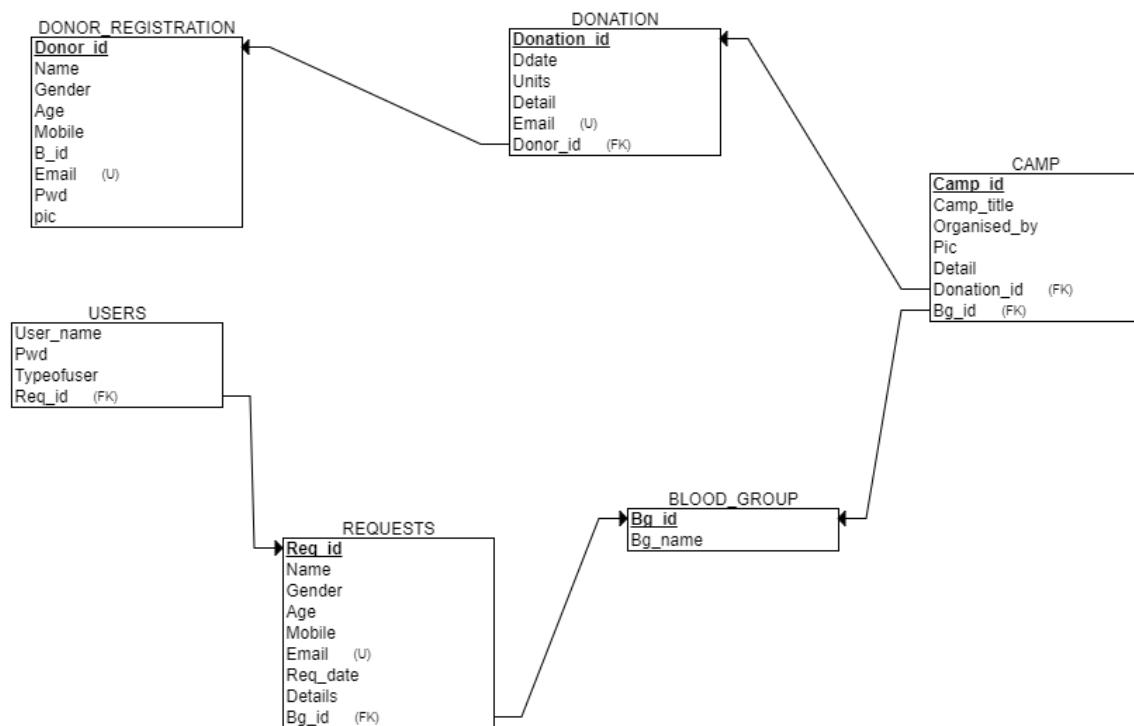    It describes entities that may have many relationships among each other.

The basic symbols for E-R diagram are as described below:

| SYMBOLS | MEANING |
| --- | --- |
|  | ENTITY TYPE |
|  | WEAK ENTITY TYPE |
|  | RELATIONSHIP TYPE |
|  | IDENTIFYING RELATIONSHIP TYPE |
|  | ATTRIBUTE |

## The E-R diagram **for blood donation management system is:**
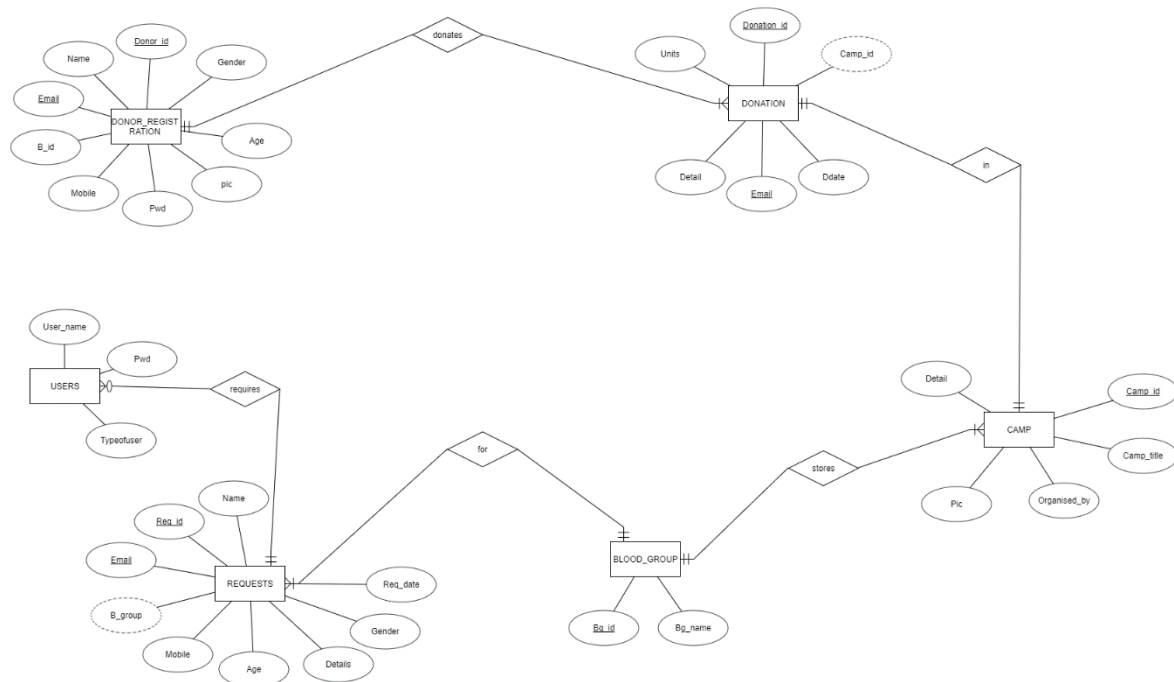


## 3.2 <u>Class Diagram</u>

## 3.3 Data Dictionary

### TABLE 1: Donor Registration

| Field Name | Data Type | Size | Constraint |
|---|---|---|---|
| Donor_id | int | 100 | Primary Key, Auto Increment |
| Name | varchar | 100 | Not Null |
| Gender | varchar | 100 | Not Null |
| Age | varchar | 100 | Not Null |
| Mobile | varchar | 100 | Not Null |
| B_id | int | 100 | Not Null |
| email | varchar | 100 | Not Null |
| pwd | int | 100 | Not Null |
| pic | varchar | 1000 | Not Null |

### TABLE 2: Donation

| Field Name | Data Type | Size | Constraints |
|---|---|---|---|
| Donation_id | int | 100 | Primary key, Auto Increment |
| Camp_id | int | 100 | Not null |
| ddate | datetime | | Not null |
| units | int | 100 | Not null |
| detail | varchar | 800 | Not null |
| email | varchar | 100 | Not null |

### TABLE 3: <u>Requests</u>

| Field Name | Data Type | Size | Constraints |
|---|---|---|---|
| Req_id | int | 100 | Primary key, auto increment |
| name | varchar | 100 | Not null |
| gender | varchar | 100 | Not null |
| age | varchar | 100 | Not null |
| mobile | varchar | 100 | Not null |
| email | varchar | 100 | Not null |
| bgroup | int | 100 | Not null |
| reqdate | datetime | | Not null |
| detail | varchar | 500 | Not null |

### TABLE 4: <u>Users</u>

| Field Type | Data Type | Size | Constraints |
|---|---|---|---|
| username | varchar | 100 | Not null |
| pwd | varchar | 100 | Not null |
| typeofuser | varchar | 100 | Not null |

### TABLE 5: <u>Camp</u>

| Field Name | Data Type | Size | Constraints |
|---|---|---|---|
| Camp_id | int | 100 | Primary key, auto increment |
| Camp_title | varchar | 500 | Not null |
| Organized_by | varchar | 500 | Not null |
| pic | varchar | 900 | Not null |
| detail | varchar | 1000 | Not null |

### TABLE 6:<u>Blood group</u>

| Field Name | Data Type | Size | Constraints |
|---|---|---|---|
| Bg_id | int | 100 | Primary key, Auto increment |
| Bg_name | varchar | 100 | Not null |

## 3.4  <u>SYSTEM FLOW DIAGRAM:</u>

```
                              ┌──────────┐
                              │  Blood   │
                              │  Bank    │
                              └──────────┘
        ┌──────────────────────────┼──────────────────────────┐
        ▼                          │                          ▼
  ┌──────────┐                     │                   ┌──────────┐
  │  Blood   │                     │                   │  Blood   │
  │  Donors  │                     │                   │ Seekers  │
  └──────────┘                     │          ┌────────┴────────┬──────────┐
        ▼                          │          ▼        ▼                   ▼
  ┌──────────────┐                 │                ┌──────────┐
  │ Registration │                 │                │  Donor   │
  └──────────────┘                 │                │  Search  │
        ▼                          │                └──────────┘
  ┌──────────────┐                 │     ┌──────────┐           ┌──────────┐
  │ Login Check  │                 │     │   Life   │           │ Mobilink │
  └──────────────┘                 │     │  Saving  │           │  Paging  │
        ▼                          │     │ Contacts │           │ Services │
  ┌──────────────┐                 │     └──────────┘           └──────────┘
  │   Details    │                 │
  │ Modification │                 │
  └──────────────┘                 │
                       ┌───────────┼───────────┐
                       ▼           ▼           ▼
                 ┌──────────┐ ┌────────┐ ┌──────────┐
                 │ General  │ │ FAQ's  │ │ Terms Of │
                 │Information│ │        │ │ Service  │
                 └──────────┘ └────────┘ └──────────┘
```
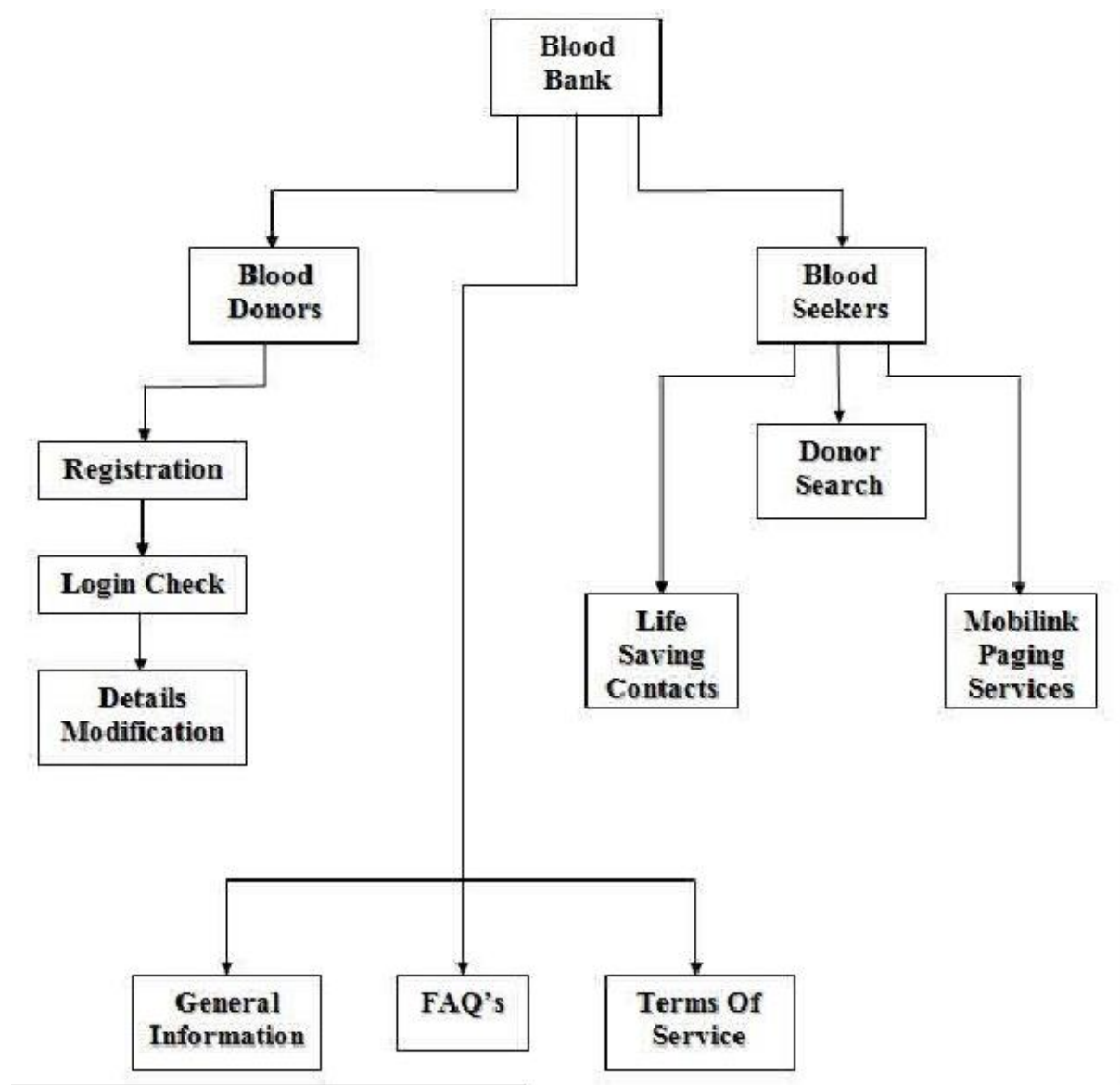
# Chapter 4 : <u>TESTING:</u>

### 4.1 <u>Testing Plan</u>

### 4.1.1  <u>Testing Process</u>:

We have tested the software process activities such as design, implementation and requirement engineering because design errors are very costly to repair once system has been designed and once it has been started to operate. Therefore, it is quite obvious to repair at an early stage of the system. So analysis is the most important process of any project.

### 4.1.2 <u>Requirement Traceability</u>:

As the most interested portion is whether the system is meeting its requirements or not, for that testing should be planned so that all requirements are individually tested. We have to check out that output of certain combinations of inputs gives the desirable results or not. Your requirement specification gives us the path to get the desirable result.

### 4.1.3 <u>Testing Schedule</u>:

We have tested each procedure back to back so that errors and omissions can be found as early as possible.

### 4.2 <u>Testing Strategy</u>

A strategy for the software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. The strategy provides a road map that describes the steps to be conducted as part of testing. When these steps are planned and then undertaken, much efforts, time and resources are required. A software testing strategy should be flexible enough to promote a customized testing approach.

At that same time it must be rigid enough to promote reasonable planning and management tracking as the project progresses.

- A software testing strategy has following characteristics:
- Testing begins at the component level and works outward towards the integration of the entire computer based system.
- Different testing techniques are appropriate at different points in time.
- Testing & Debugging are different activities but debugging must be accommodated in any testing strategy. We checked entire project thoroughly so not even a single mistake would be there.

## 4.3 SOFTWARE TESTING

Testing is the process of executing then programs with the intention of finding out errors. During the process, the project is executed with set of test and the output of the website is evaluated to determine if the project is performing as expected. Testing makes a logical assumption that if all the parts of the module are correct then goal will be successfully achieved. Testing includes after the completion of the coding phase. The project was tested from the very beginning and also at each step by entering different type of data. In the testing phase some mistakes were found, which did not come to knowledge at the time of coding the project. Then changes were made in the project coding so that it may take all the relevant data and gives the required result. All the forms were tested one by one and made all the required changes.

Testing is vital to the success of the system. Testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved. A small system error can conceivably explode into a much larger problem. Effective testing early in the process translates directly into long-term cost savings from a reduced number of errors. For the verification and validation of data various-nesting task are performed. Testing is itself capable of finding the syntactical mistakes in the system but user need to test the system for logical checking.

## 4.3.1 LEVELS OF TESTING:

The aim of the testing process is to identify all the defects in the website. It is not practical to test the website with respect to each value that the input request data may assume. Testing provides a practical way of reducing defects in the website and increasing the user's confidence in a developed system. Testing consists of subjecting the website to a set of test inputs and observing if the program behaves as expected. If the program fails to Testing behave as expected then conditions under which failure occurs are noted for later debugging and correction. The following things are associated with testing:

Failure is a manifestation of an error. But the mere presence of an error may not necessarily lead to a failure. A test case is the triplet [I, S, O] where I am data input to the system. S is the

state of the state of the system at which the data is input, O is the expected output of the system A test suite is the set of all test cases with which a given software product is to be tested.

### 4.3.2 FUCTIONAL TESTING:

Here the system is a black box whose behavior is determined by studying its inputs and related outputs. The key problem is to select the inputs that have a huge probability of being members of a set in many case; the selection of these test cases is based on the previous studies.

### 4.3.3 STRUCTURAL TESTING:

A great deal can be learnt about the strength and the limitation of the application by examinee the manner in which the system breaks. This type of testing has two limitations .It tests failure behavior of the system circumstances may arise through an unexpected combination of events where the node placed on the system exceeds the maximum anticipated load. The structure of the each module was checked at every step. Some structures were firstly wrong, which came to notice at the time of the connectivity.

### 4.3.4 UNIT TESTING:

In unit testing the entire individual functions and modules were tested independently. By following this strategy all the error in coding were identified and corrected. This method was applied in combination with the white and black box testing techniques to find the errors in each module.

Unit testing is normally considered an adjunct to the coding step. Unit test case design was started after source level code had been developed, reviewed, and verified for correct syntax. A review of design information provides guidance for establishing test cases that were likely to uncover errors in each of the categories discussed above. Each test case was coupled with a set of expected results.

# Chapter 5: <u>RESULTS/ SNAPSHOTS</u>

## 5.1 <u>Home page</u> :

This is the home page or the main page of a blood donation management system. This is the main page of a client side. This page define all about related to project. This page also includes the gallery of the camps**.**
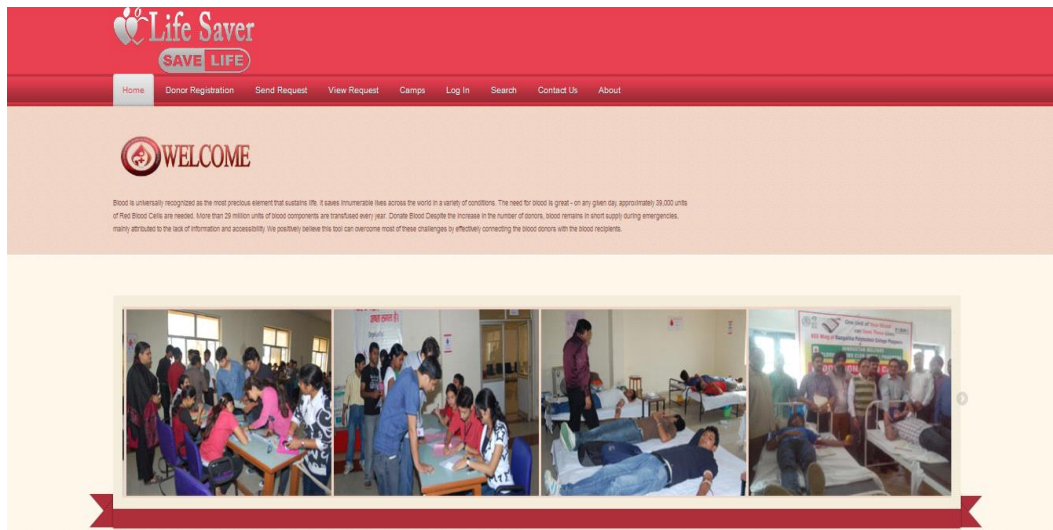


## 5.2 <u>Registration</u> :

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.

## 5.3 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.



## 5.4 Camps:

Camp page includes the information about camps, this camps organized the blood donation camp. Donor can register the account by clicking on new register.
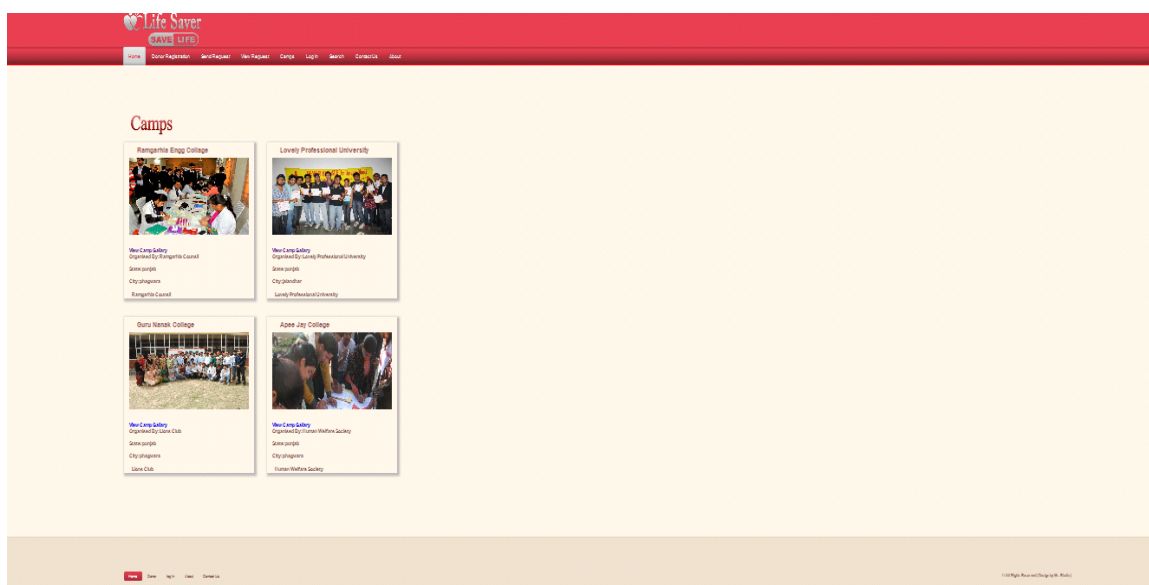
## 5.5 **Donor Login:**

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.
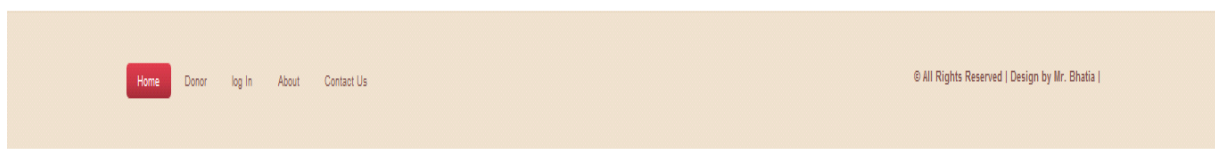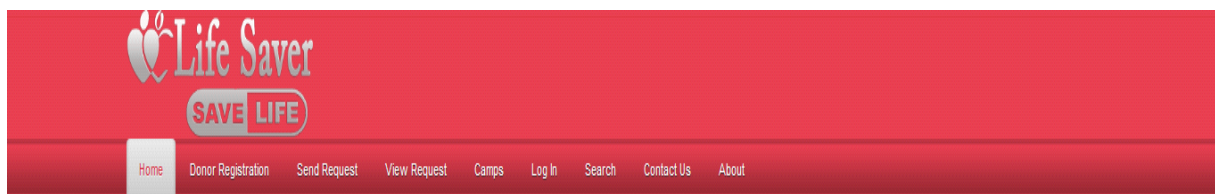
## 5.6 Search:

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

## Blood groups name

- **A**
- **A+**
- **B**
- **B+**
- **AB+**
- **AB_**
- **O+**
- **O-**
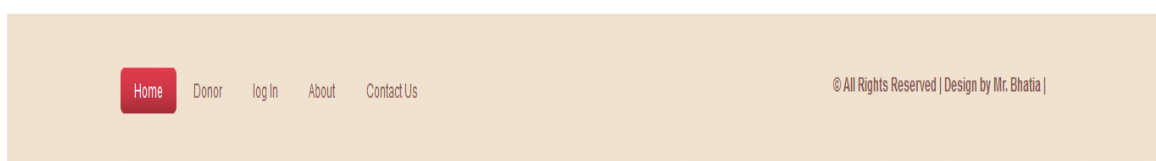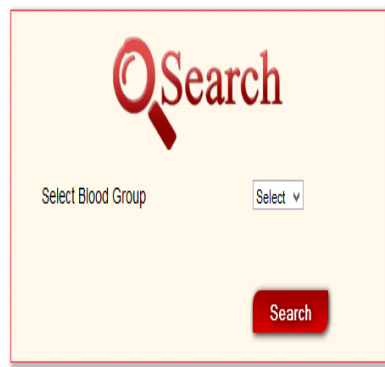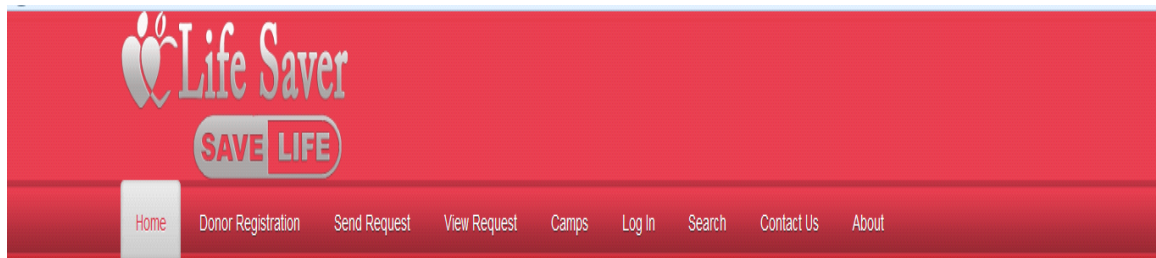
## 5.7 **Donor Panel:**

This page is 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
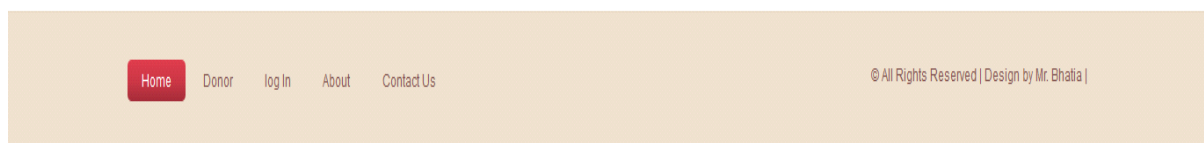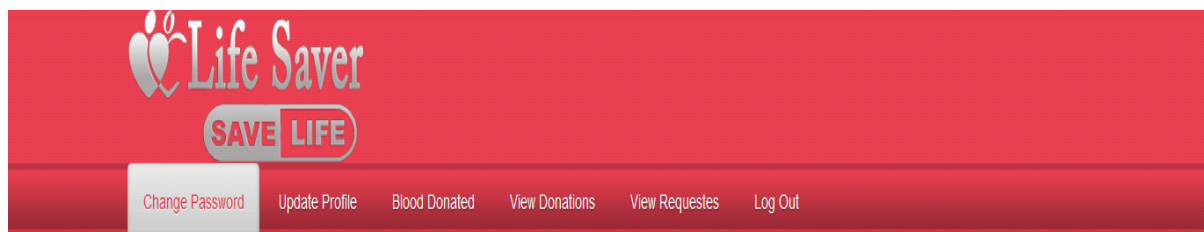
## 5.8 Change Password:
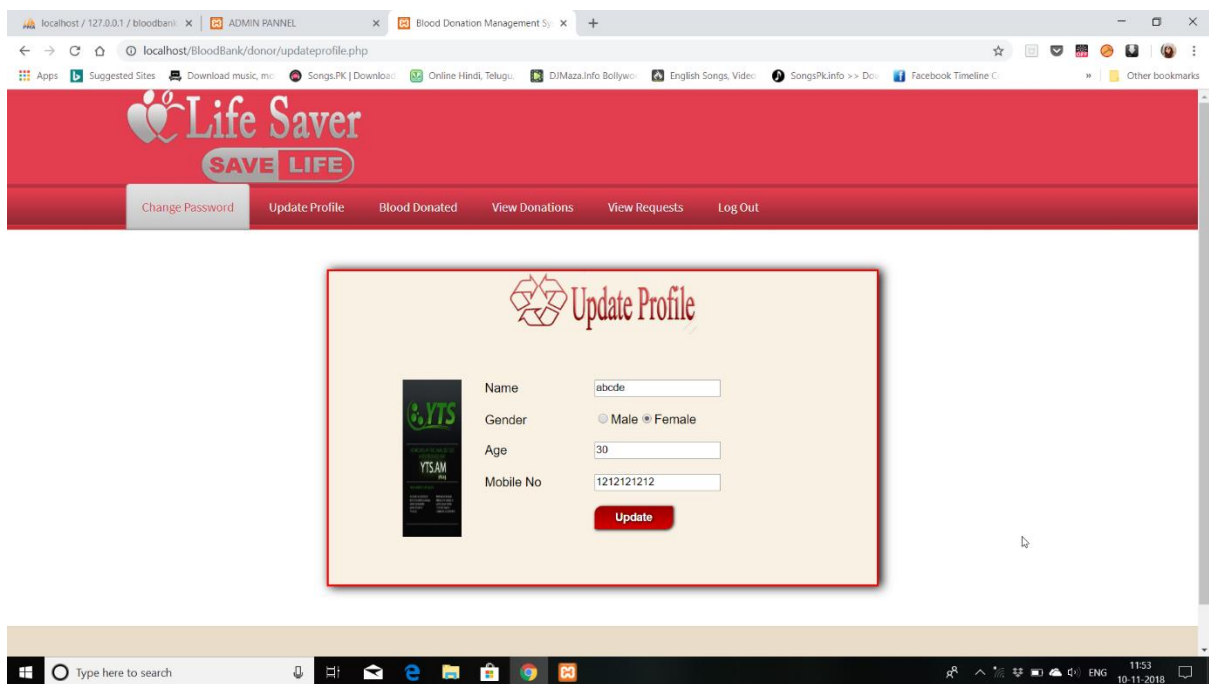
Above snap short describe how the donor change our password. This is only used by a donor.



## 5.9 Update Profile:

It describe how the donor update our profile .This is only used by a donor.

## 5.10 <u>Blood Donated:</u>

It describes 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.



## 5.11 <u>Admin Panel:</u>

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.
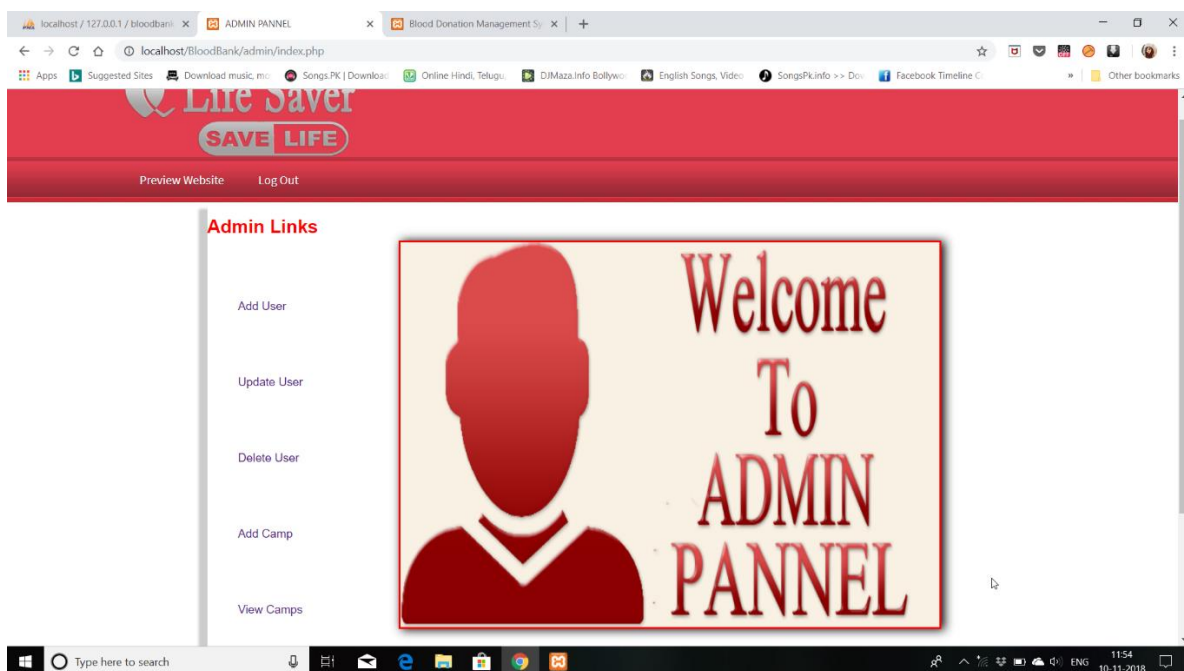
## 5.12 <u>Add user:</u>

Show the addition of user. Person will get the blood immediately he/she requested for the particular blood group he/she has requested<u>.</u>



## 5.13 <u>Update User</u>
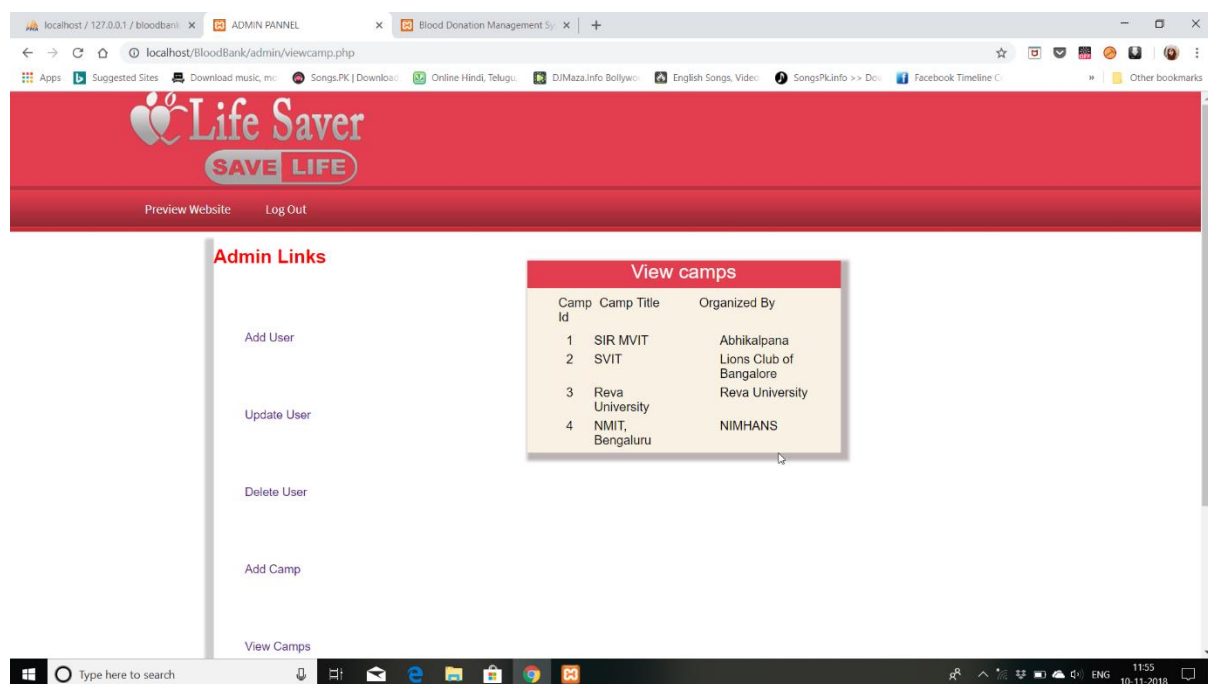
Show the updation of user.

## 5.14 <u>Add Camp</u>

Show the addition of camp. Person will get the blood immediately he/she requested for the particular blood group he/she has requested.
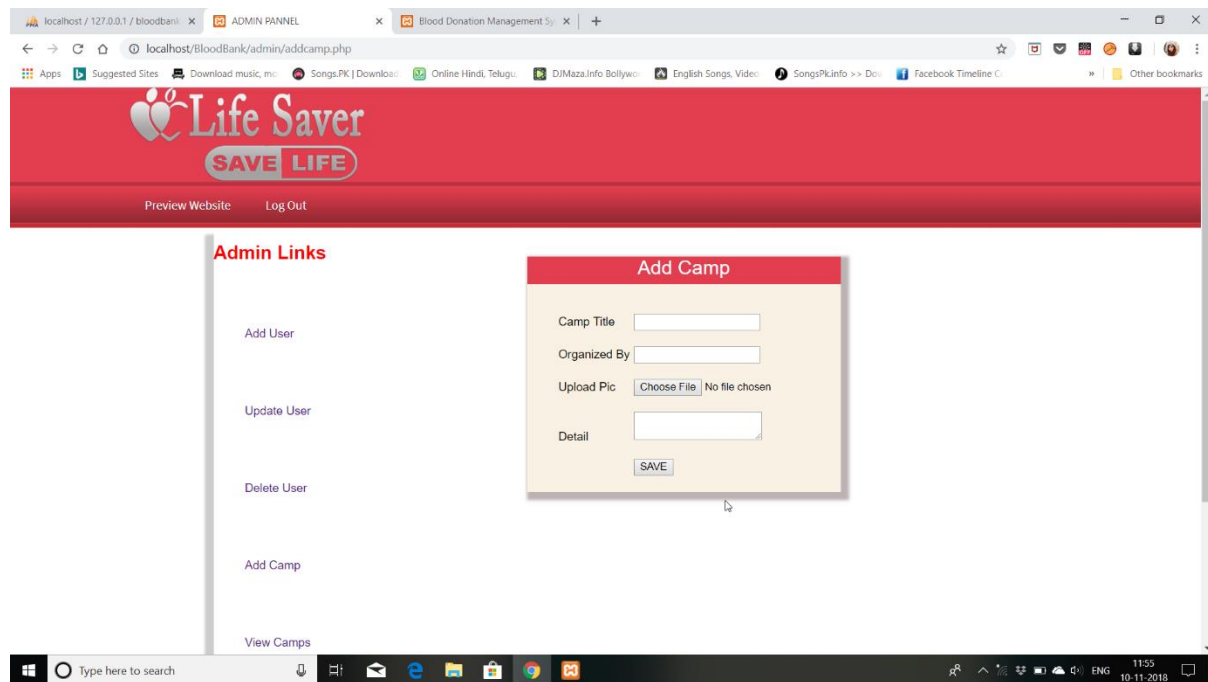


## 5.15 <u>View Camp</u>

Show the details of camp.

# Chapter 6: <u>CONCLUSION</u>

Blood Donation Management System (BDMS) is a browser based system that is designed to store, process, retrieve and analyse 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.

# Chapter 7: <u>FUTURE ENCHANCEMENT</u>

❖ This system will be linked by internet, so that the other hospitals can use this data.

❖ This system will be linked with nearby blood bank options which provides the user with google maps using which a real estimate of the distance of the desired blood bank can be made by the user.

# Chapter 8:  <u>REFERENCES</u>

- Software requirement specifications from Internet
- www.google.com
- www.wikipedia.com
- www.slideshare.com
- www.w3schools.com

## <u>MYSQL (Structured Query Language):</u>

**MySQL** is an open source Relational Database Management System (RDBMS). MySQL is written in c and c++. 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 python ".

## <u>PHP(Hypertext Preprocessor):</u>

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.