

REFERENCES

Books and Websites

1. Author name:

[1] Fundamentals Database Systems by Ramez Elmasri & Navathe.

[2] Database Management Systems Raghu Ramakrishnan

2. Website name:

<https://www.apachefriends.org/>

<http://code-projects.org/c/app-projects>

<http://code-projects.org/c/languages>

<https://www.udemy.com/php-for-complete-beginners-includes-mysql-object-oriented/>

CHAPTER 4

IMPLEMENTATION

4.1 Introduction

The project is implemented using MySQL database along with HTML, CSS and PHP. In this project, we use the below mentioned concepts for implementing different type of effects like HTML pages to demonstrate on a web page as a stand-alone or web-based application.

ZAMPP: In this project XAMPP is used to connect the server. XAMPP is a free and opensource cross-platform web server solution stack package, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.

PHP: PHP is a general-purpose scripting language especially suited to web development. PHP code is usually processed on a web server by a PHP interpreter. Any PHP code in a requested file is executed by the PHP runtime, usually to create dynamic web page content or dynamic images used on websites or elsewhere. It can also be used for command-line scripting and client-side graphical user interface (GUI) applications.

HTML: Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. Web browsers receive HTML documents from a web server or from a local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included queues for the appearance of the document.

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CHAPTER 1

INTRODUCTION

A blood donation is a process whereby a person voluntarily has blood drawn to be used for future transfusions when in need at hospitals for treatment procedures that require them. Donation may be of whole blood (blood drawn directly from the body) or of specific components of the blood; such as red blood cells, white blood cells, plasma, and platelets. Blood banks often participate in the process of collecting blood and other procedures such as managing stocks, approving blood requests and updating donation information. The inspiration of this project is to improve blood banks in Pakistan and to develop a blood bank information system which focuses on making an online system that is accessible for both donors and administrators. Donors can directly receive information regarding their previous blood donations, including their blood results and donation history, in order to easily schedule their next donations. They can also update the personal information through the system, without having to contact the blood bank registry. information if necessary. The administrator is also responsible for responding to the hospital's blood requests and checking the stocks in the blood bank's inventory.

1.1 Objective

The goal of the project is to develop a web application for blood banks to manage information about their donors and blood stock. The main objectives of this website development can be defined as follows:

- 1. To develop a system that provides functions to support donors to view and manage their information conveniently.
- 2. To maintain records of blood donors, blood donation information and blood stocks in a centralized database system.
- 3. To inform donors of their blood result after their donation.
- 4. To support searching, matching and requesting for blood convenient for administrators.

```
PRIMARY KEY (`blood_contact_id`),
KEY `contact_fk` (`contact_fk`),
KEY `blood_fk` (`blood_fk`),
CONSTRAINT `blood_contact_ibfk_1` FOREIGN KEY (`contact_fk`)
REFERENCES `contact` (`contact_id`) ON UPDATE CASCADE,
CONSTRAINT `blood_contact_ibfk_2` FOREIGN KEY (`blood_fk`)
REFERENCES `blood_group` (`blood_id`) ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
/*Data for the table `blood_contact` */
/*Table structure for table `blood_donor_group` */
DROP TABLE IF EXISTS `blood_donor_group`;
CREATE TABLE `blood_donor_group` (
  `blood_donor_group_id` int(100) NOT NULL AUTO_INCREMENT,
  `donor_fk` int(100) DEFAULT NULL,
  `blood_fk` int(100) DEFAULT NULL,
  PRIMARY KEY (`blood_donor_group_id`),
  KEY `donor_fk` (`donor_fk`),
  KEY `blood_fk` (`blood_fk`),
  CONSTRAINT `blood_donor_group_ibfk_1` FOREIGN KEY (`donor_fk`)
REFERENCES `donor` (`donor_id`) ON UPDATE CASCADE,
  CONSTRAINT `blood_donor_group_ibfk_2` FOREIGN KEY (`blood_fk`)
REFERENCES `blood_group` (`blood_id`) ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
/*Data for the table `blood_donor_group` */
/*Table structure for table `blood_group` */
DROP TABLE IF EXISTS `blood_group`;
CREATE TABLE `blood_group` (
  `blood_id` int(100) NOT NULL AUTO_INCREMENT,
  `blood_group` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`blood_id`)
```

```
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=latin1;

/*Data for the table `blood_group` */

insert into `blood_group`(`blood_id`,`blood_group`) values
(3,'sd'),
(5,'D#');

/*Table structure for table `blood_request_group` */
DROP TABLE IF EXISTS `blood_request_group`;
CREATE TABLE `blood_request_group` (
  `blood_request_group_id` int(100) NOT NULL AUTO_INCREMENT,
  `requester_fk` int(100) DEFAULT NULL,
  `blood_fk` int(100) DEFAULT NULL,
  PRIMARY KEY (`blood_request_group_id`),
  KEY `requester_fk` (`requester_fk`),
  KEY `blood_fk` (`blood_fk`),
  CONSTRAINT `blood_request_group_ibfk_1` FOREIGN KEY (`requester_fk`)
REFERENCES `requester` (`requester_id`) ON UPDATE CASCADE,
  CONSTRAINT `blood_request_group_ibfk_2` FOREIGN KEY (`blood_fk`)
REFERENCES `blood_group` (`blood_id`) ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/*Data for the table `blood_request_group` */

/*Table structure for table `city` */
DROP TABLE IF EXISTS `city`;
CREATE TABLE `city` (
  `city_id` int(100) NOT NULL AUTO_INCREMENT,
  `city_code` varchar(100) DEFAULT NULL,
  `city_name` varchar(100) DEFAULT NULL,
  `description` varchar(120) DEFAULT NULL,
  `donor_fk` int(100) DEFAULT NULL,
  `requester_fk` int(100) DEFAULT NULL,
  `state_fk` int(100) DEFAULT NULL,
```

CONCLUSION

This proposed Blood Bank donation Management System gives a reliable platform for both donors and acceptors. The BBDMS is a web-based application that helps to minimize human errors and problems pertaining to data redundancy.

The organization of blood donation should be best and safe to the patient/recipients and to the blood donor also. With increase in population and development of more advanced medical and surgical procedures, the need for blood is ever increasing.

```
PRIMARY KEY (`city_id`),
KEY `donor_fk` (`donor_fk`),
KEY `requester_fk` (`requester_fk`),
KEY `state_fk` (`state_fk`),
CONSTRAINT `city_ibfk_1` FOREIGN KEY (`donor_fk`) REFERENCES `donor`
(`donor_id`) ON UPDATE CASCADE,
CONSTRAINT `city_ibfk_2` FOREIGN KEY (`requester_fk`) REFERENCES
`requester` (`requester_id`) ON UPDATE CASCADE,
CONSTRAINT `city_ibfk_3` FOREIGN KEY (`state_fk`) REFERENCES `state`
(`state_id`) ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=latin1;
/*Data for the table `city` */
insert                                     into
`city`(`city_id`,`city_code`,`city_name`,`description`,`donor_fk`,`requester_fk`,`state
_fk`) values
(3,'CLSs','Thihariyass','Thiahriya town shop',NULL,NULL,36);
/*Table structure for table `contact` */
DROP TABLE IF EXISTS `contact`;
CREATE TABLE `contact` (
  `contact_id` int(100) NOT NULL AUTO_INCREMENT,
  `name` varchar(100) DEFAULT NULL,
  `email` varchar(100) DEFAULT NULL,
  `address` varchar(200) DEFAULT NULL,
  `phone` varchar(100) DEFAULT NULL,
  `message` varchar(200) DEFAULT NULL,
  `member_fk` int(100) DEFAULT NULL,
  PRIMARY KEY (`contact_id`),
  KEY `member_fk` (`member_fk`),
  CONSTRAINT `contact_ibfk_1` FOREIGN KEY (`member_fk`) REFERENCES
`member` (`member_id`) ON UPDATE CASCADE
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
/*Data for the table `contact` */
/*Table structure for table `donor` */
DROP TABLE IF EXISTS `donor`;
CREATE TABLE `donor` (
  `donor_id` int(100) NOT NULL AUTO_INCREMENT,
  `name` varchar(100) DEFAULT NULL,
  `father_name` varchar(100) DEFAULT NULL,
  `gender` varchar(100) DEFAULT NULL,
  `dob` varchar(100) DEFAULT NULL,
  `body_weight` varchar(100) DEFAULT NULL,
  `email` varchar(100) DEFAULT NULL,
  `blood_group` varchar(40) DEFAULT NULL,
  `state` varchar(100) DEFAULT NULL,
  `city` varchar(100) DEFAULT NULL,
  `address` varchar(200) DEFAULT NULL,
  `pincode` varchar(100) DEFAULT NULL,
  `phone` varchar(100) DEFAULT NULL,
  `image` varchar(255) DEFAULT NULL,
  `username_fk` varchar(100) NOT NULL,
  `status` int(12) DEFAULT NULL,
  PRIMARY KEY (`donor_id`,`username_fk`),
  KEY `member` (`username_fk`)
) ENGINE=InnoDB AUTO_INCREMENT=18 DEFAULT CHARSET=latin1;
/*Data for the table `donor` */
insert into
`donor`(`donor_id`,`name`,`father_name`,`gender`,`dob`,`body_weight`,`email`,`blood_group`,`state`,`city`,`address`,`pincode`,`phone`,`image`,`username_fk`,`status`)
values
```


- 5. To provide a function to send an e-mail directly to the donor for their user account and the hospital, the availability of the blood bag.

1.2 aim

The blood bank management system aims to fulfill the gap between blood donors and doctors. When a blood donor gets an easy and suitable platform to register and donate blood, it becomes easier for many people who wish to donate blood but cannot find a proper platform for it due to their busy schedules.

1.3 scope

Blood Bank Donation Management System is the process of the managing and controlling the activities of a blood. It includes maintaining and managing the records of Donor & Member available in the Blood Bank Management System. It also maintains the list of donor, blood requirement, and approval status held in the blood bank.

1.4 limitations

- Lack of immediate retrievals: -The information is very difficult to retrieve and to find particular information like- E.g. – To find out about the donor's history, the user has to go through various registers. This results in inconvenience and wastage of time.
- Does not meet emergency requirements.
- Searching required blood group and donor is tedious job.

1.5 Problem statement

The percentage of people donating blood is increasing day by day due to awareness to donate blood for those needed. The blood received must be managed thoroughly so that there will be no negative effect on the blood receiver once they received blood.

CHAPTER 2

ANALYSIS

There are two types of process in the existing system.

Blood Donation Process by Donors

When a new donor comes to donate blood, they are required to fill out their personal information during the registration process before making a donation only Blood bank administrators have the authority to access the donor's records.

Blood Request Process by Hospitals

Hospitals can request for blood by calling in or e-mailing the blood bank the type of blood and the quantity that is in need. The administrator is responsible in checking the availability of the blood type according to the request.

SYSTEM REQUIREMENTS SPECIFICATION

2.1 Hardware Requirements

- **Microprocessor:** i5 core processor
- **Main memory:** 8 GB RAM
- **Hard Disk** : 1 TB
- **Keyboard: Mouse:** Qwerty
- **Monitor:** 14'CRT

2.2 Software Requirements

- **Operating system:** Windows 10 64bit processor
- **IDE Used:** Visual Studio Code 1.52
- **Database:** MySQL
- **Front End:** HTML CSS
- **Back End:** PHP

CHAPTER 3

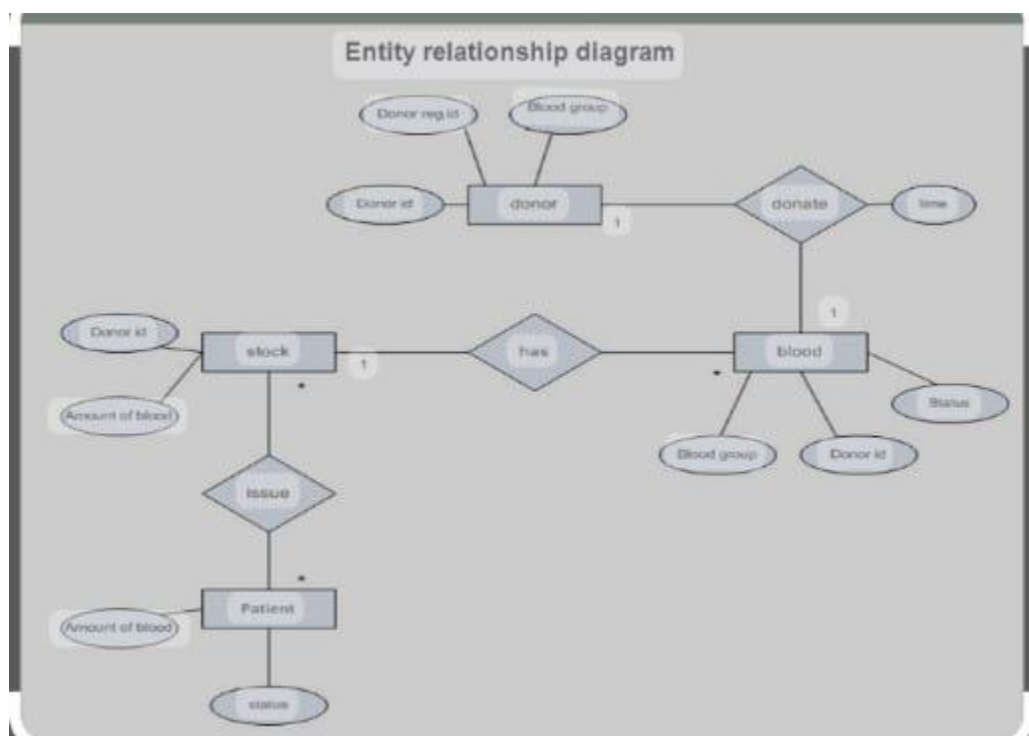
DESIGN

3.1 System Design

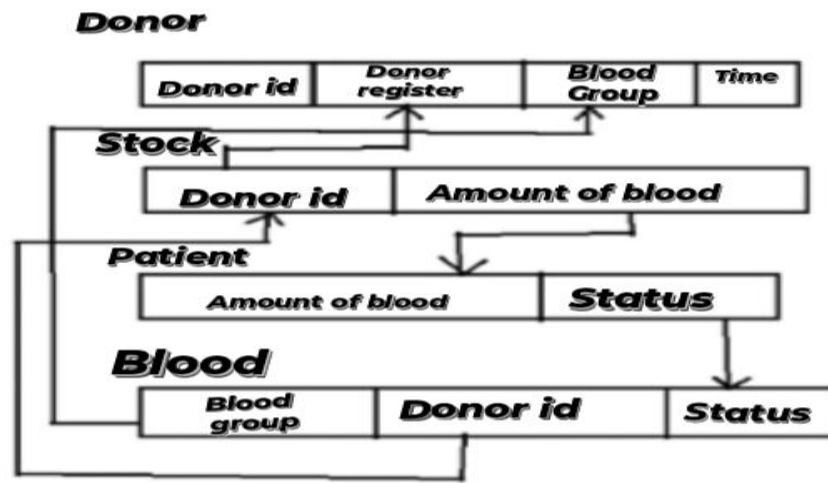
The aim of the project is to develop a website using HTML, CSS and MYSQL The Blood Bank Donor Management System is a web application that will be used to manage the details of the donors and the requestors. Basically, the system will be designed to suit the need of the blood.

The database implemented captures the most essential features. It also stores information about the donors, this data can be used to contact the donors and view the projects awards and credit points.

3.2 ER Diagram



3.3 Schema Diagram



3.4 Database Design

Table	Action	Rows	Type	Collation	Size
<input type="checkbox"/> blood_contact	★ Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	48.0 KiB
<input type="checkbox"/> blood_donor_group	★ Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	48.0 KiB
<input type="checkbox"/> blood_group	★ Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16.0 KiB
<input type="checkbox"/> blood_request_group	★ Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	48.0 KiB
<input type="checkbox"/> city	★ Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	64.0 KiB
<input type="checkbox"/> contact	★ Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	32.0 KiB
<input type="checkbox"/> donor	★ Browse Structure Search Insert Empty Drop	6	InnoDB	latin1_swedish_ci	32.0 KiB
<input type="checkbox"/> email_subs	★ Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	32.0 KiB
<input type="checkbox"/> member	★ Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16.0 KiB
<input type="checkbox"/> requester	★ Browse Structure Search Insert Empty Drop	5	InnoDB	latin1_swedish_ci	32.0 KiB
<input type="checkbox"/> state	★ Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16.0 KiB
11 tables	Sum	19	InnoDB	utf8mb4_general_ci	384.0 KiB

Table 1: blood bank

MySQL, pronounced either "MySQL" or "My Sequel," is an open-source relational database management system. It is based on the structure query language (SQL), which is used for adding, removing, and modifying information in the database. Standard SQL commands, such as ADD, DROP, INSERT, and UPDATE can be used with MySQL

1.2Methodology

1. Project Identification and Selection

In this project, we aimed to develop an blood bank system which will focus mainly on managing the donor's blood information. Anyone who is interested in blood donation can donate the blood at the hospital or blood donation centers.

2. Project Initiation and Planning

To begin the project, we have gathered user requirement of this system and prepare the scope and objective. The results from this phase are scope and limitation, objectives, cost and benefits, feature of the proposed system and user interface design.

3. Analyzing System needs

We have studied and identified problems of existing system, then we develop data flow diagram for the existing system. We also develop entity relation diagram (E-R diagram) for the proposed system

4.3Module Description

1) Module login:

```
<?php  
  
include('connection.php');  
  
session_start();  
  
$username = $_POST['username'];
```

```
$password = $_POST['password'];

$login = $connection->query("SELECT * FROM member WHERE
username='$username' AND password='$password'");

$fetch = $login->fetch_array();

if($login->num_rows == 1){

    if($fetch['usertype'] == 'admin'){

        $_SESSION['member_id'] = $fetch['member_id'];

        $_SESSION['username'] = $fetch['username'];

        header('location:admin_dashboard/admin_dashboard.php');

    }elseif ($fetch['usertype'] == 'user') {

        $_SESSION['userid'] = $fetch['user_id'];

        $_SESSION['membername'] = $fetch['username'];

        header('location:user_dashboard/user_dashboard.php');

    }

}

else {

    $_SESSION['error'] = "";

}

?>
```

2) logout module:

```
<?php

include('connection.php');

session_start();

session_unset($_SESSION['username']);

session_unset($_SESSION['member_id']);

header('location:index.php');
```

 ?>
3) connection module:

<?php

\$connection = new mysqli('localhost', 'root', '', 'blood_bank');

?>

4) database module:

/*

SQLyog Community v12.4.3 (64 bit)

MySQL - 5.6.17 : Database - blood_bank

*/

/*!40101 SET NAMES utf8 */;

/*!40101 SET SQL_MODE="*/;

 /*!40014 SET @OLD_UNIQUE_CHECKS=@ @UNIQUE_CHECKS,
 UNIQUE_CHECKS=0 */;

 /*!40014 SET
 @OLD_FOREIGN_KEY_CHECKS=@ @FOREIGN_KEY_CHECKS,
 FOREIGN_KEY_CHECKS=0 */;

 /*!40101 SET @OLD_SQL_MODE=@ @SQL_MODE,
 SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;

/*!40111 SET @OLD_SQL_NOTES=@ @SQL_NOTES, SQL_NOTES=0 */;

 CREATE DATABASE /*!32312 IF NOT EXISTS*/`blood_bank` /*!40100
 DEFAULT CHARACTER SET latin1 */;

USE `blood_bank`;

/*Table structure for table `blood_contact` */

DROP TABLE IF EXISTS `blood_contact`;

 CREATE TABLE `blood_contact` (
 `blood_contact_id` int(100) NOT NULL AUTO_INCREMENT,
 `contact_fk` int(100) DEFAULT NULL,
 `blood_fk` int(100) DEFAULT NULL,

CHAPTER 5

SAMPLE OUTPUT

5.1 Main page: This page shows the main page of this mini project. This page has login page.

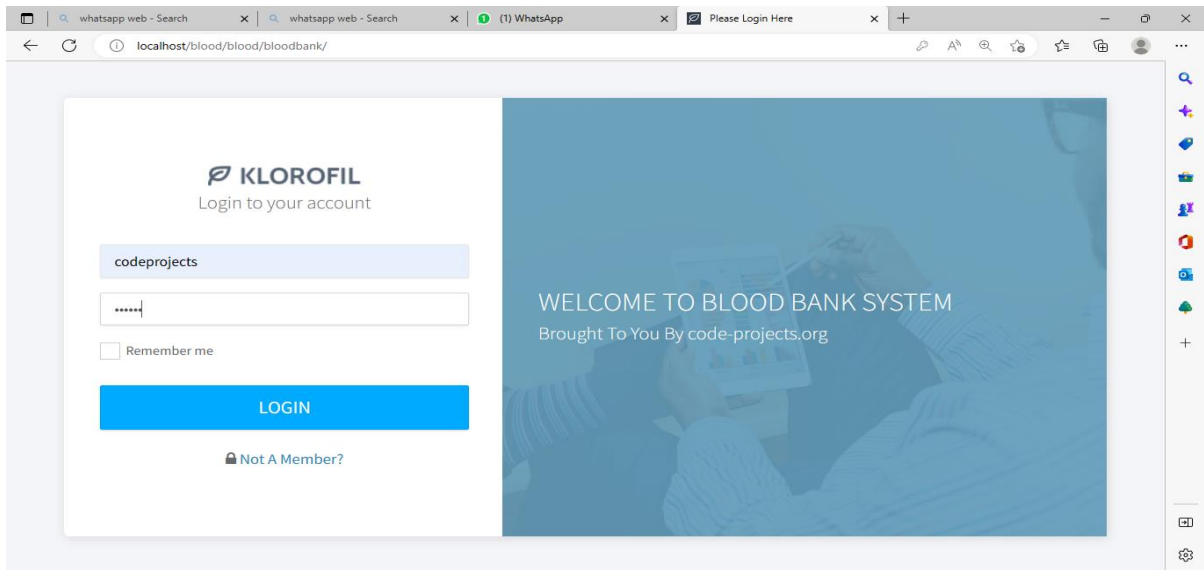


Figure 5.1 login page

5.2 Dashboard: This page shows the weekly overview and no: of downloads, Sales visits conversations.

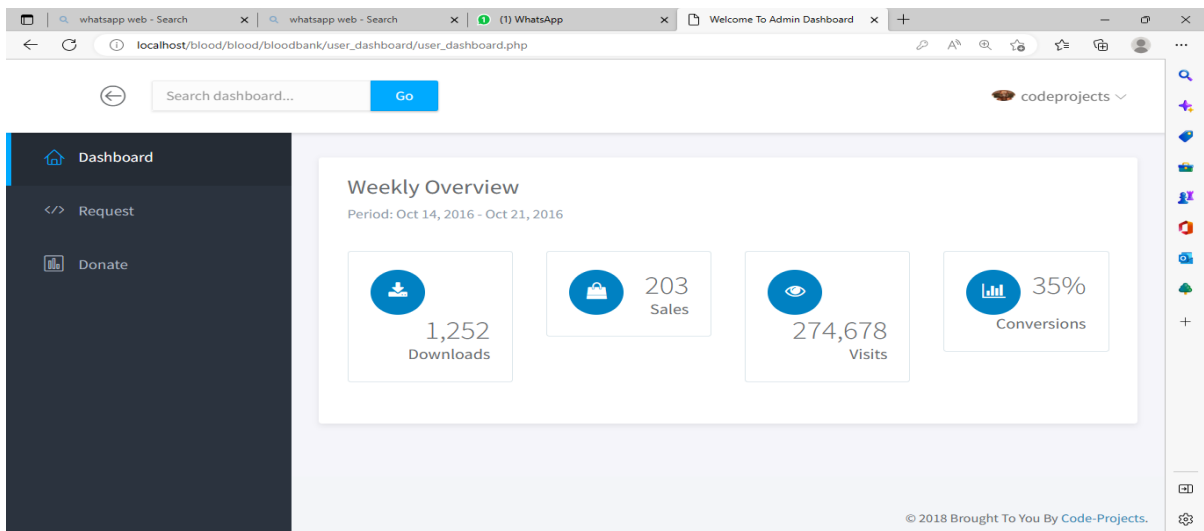


Figure 5.2 Dashboard

5.3 Request for blood: this page helps us to request for the type of the blood ,units of blood etc.

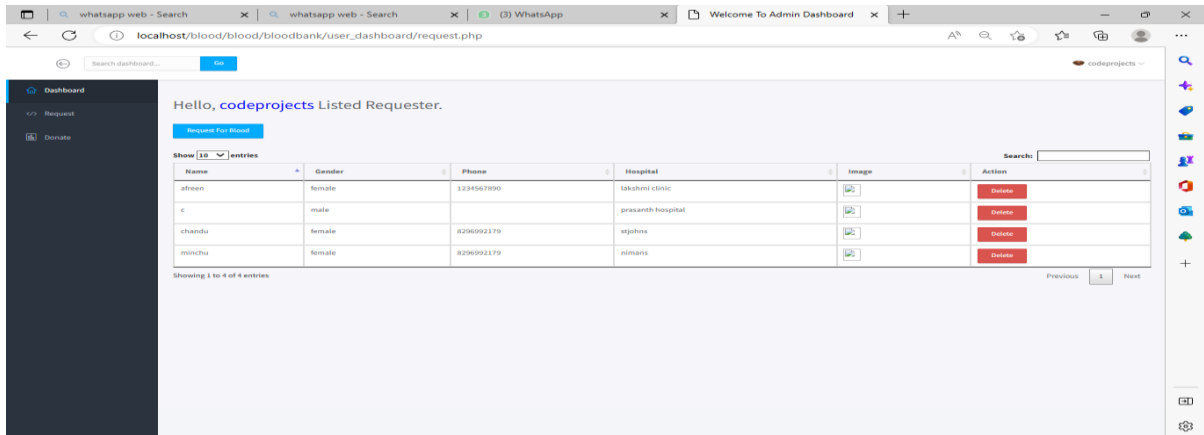


Figure 5.3 Request for blood

5.4 Donate blood: this page allows you to donate the blood by filling up your details and also can view the existing donor's information

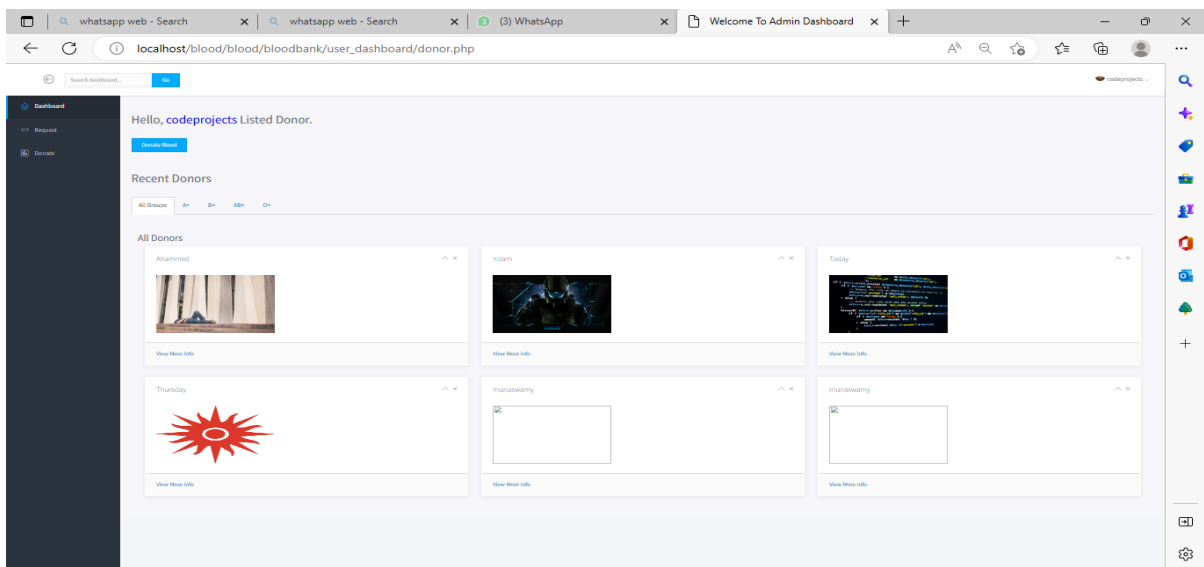


Figure 5.4 Donate blood

5.5 manage state: this page allows to manage the states where we can add and delete states.

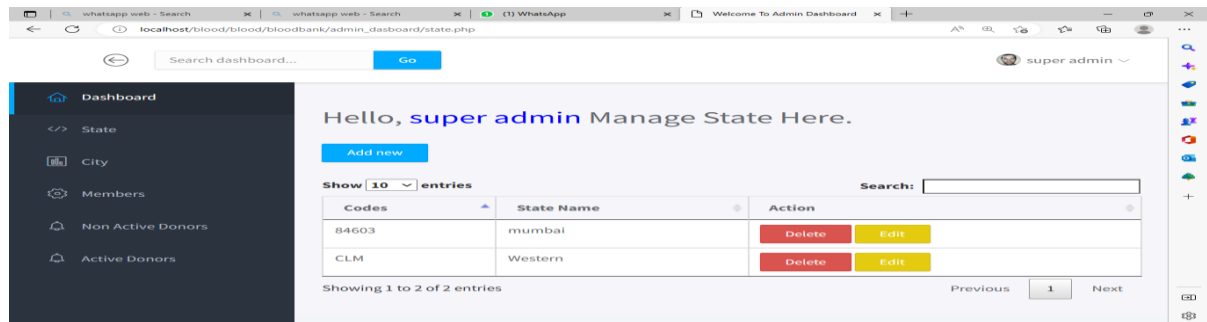


Figure 5.5 manage state

5.6 manage city: this page allows to manage the city where we can add and delete the cities.

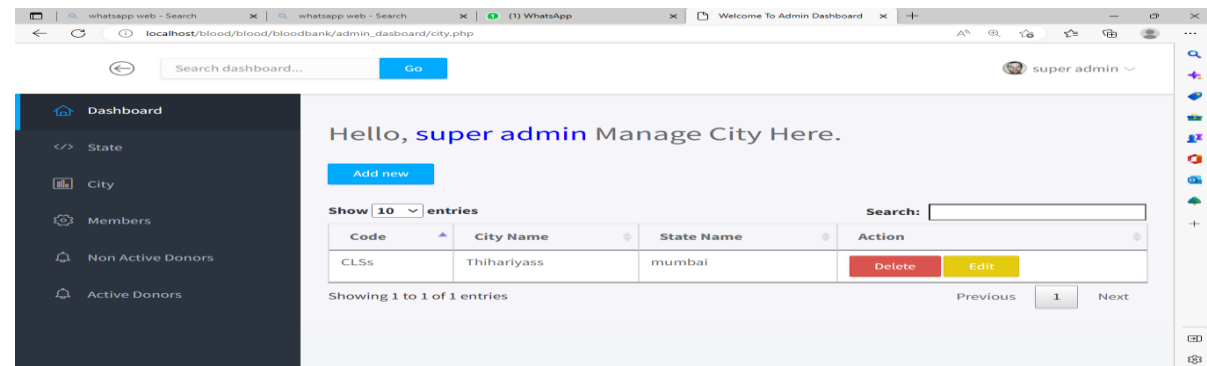


Figure 5.6 manage city

5.7 manage members: this page allows to manage the members.

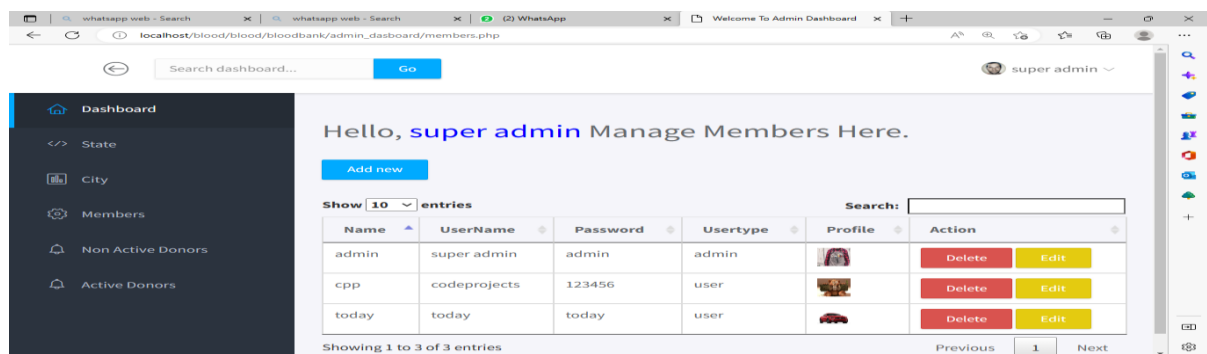


Figure 5.7 manage members

5.8 manage active donors: this page manages the active donors who r ready to donate their blood.

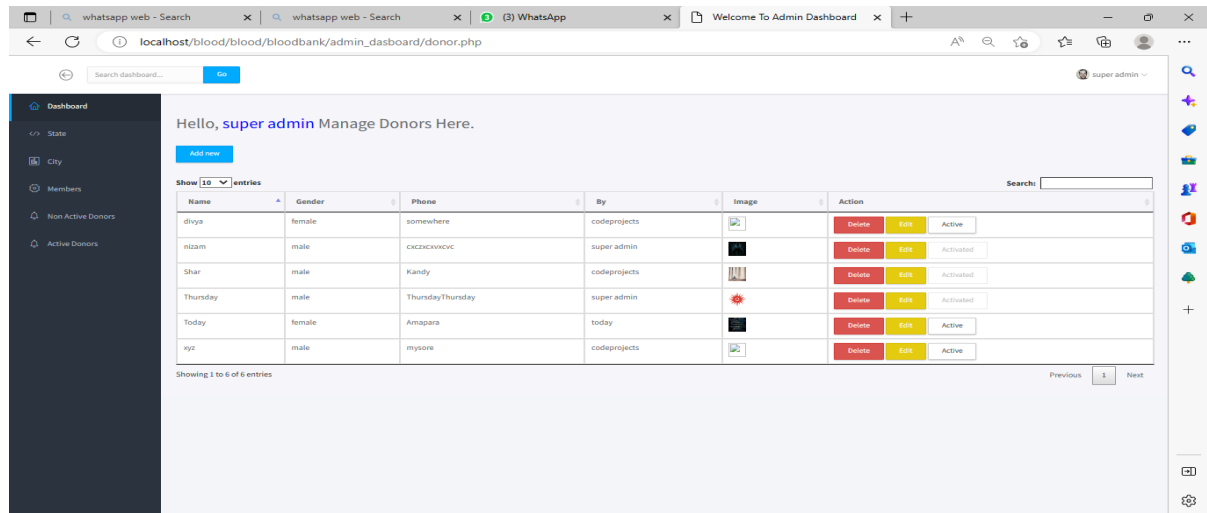


Figure 5.8 manage active donors

5.9 manage non-active donors: this page manages the non-active donors who are not ready to donate their blood.

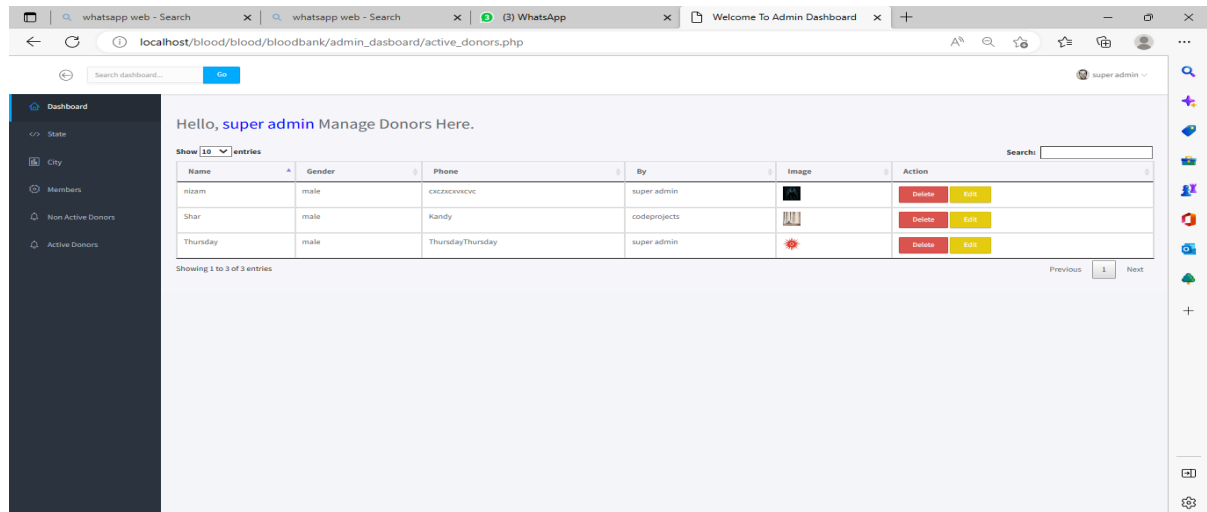


Figure 5.9 manage non-active donors