1. INTRODUCTION

A Blood bank is a center where blood gathered as a result of blood donation is stored and preserved for later use in blood transfusion. The term "blood bank" typically refers to a division of a hospital where the storage of blood product occurs and where proper testing is performed. However, it sometimes refers to a collection center, and indeed some hospitals also perform collection.

The numbers of persons who are in need of blood are increasing in large number day by day. In order to help people who are in need of blood, Blood Bank Management System can be used effectively for getting the details of blood donors having the same blood group and within the same city.

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.

Blood Bank Management System is available to everyone easily. A person who likes to donate blood gives his entire details i.e., fill in the registration form and can create a username with a password by which he can modify his details if at all there are any changes in his information given before.

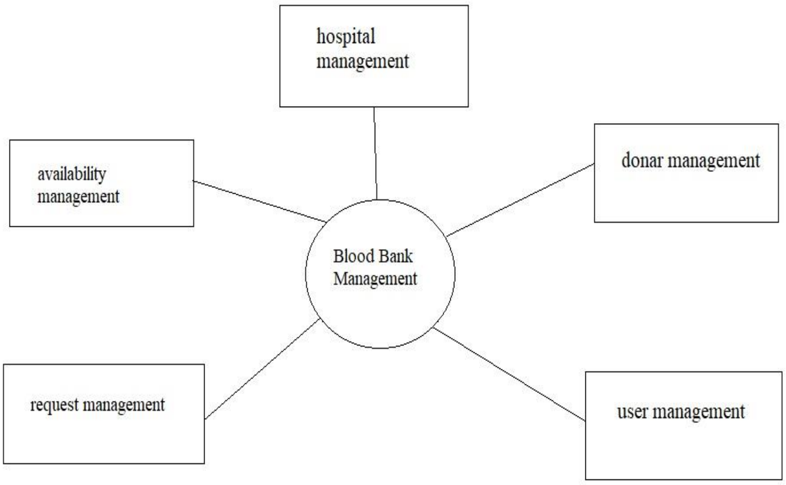
1.1 PURPOSE OF THE PROJECT

To design a blood bank management system which can be used to simplify and automate the process of searching for blood in case of emergency and maintain the records of blood donors, recipients, blood donation programs and blood stocks in the bank.

1.2 OBJECTIVES

- The main objective of the project is to design and develop a user friendly system
- Easy to use and an efficient computerized system.
- To make a software fast in processing, with good user interface.
- To make software with good user interface so that user can change it and it should be used for a long time without error and maintenance.
- To provide synchronized and centralized donor and blood stock database.
- Provides Security to the data by using login and password method.
- To provide immediate storage and retrieval of data and information.
- Reducing paperwork.

1.3 SYSTEM ANALYSIS AND FLOW OF DATA



(a)

Analysis may be defined as the process of dividing the problems into parts identifying each part and establishing relationship in the parts. Analysis is detailed study of various operations performed by system and the relationships within and outside the system.

Admin page is used to manage all the user login information .admin is allowed to delete users.

User log-in to the user login page and enter the information and user can also check availability of blood.

Hospitals are associated with blood bank where the user can check the availability of blood here also.

Donor can fill his/her information in the donor page about his donated information.

2. REQUIREMENTS SPECIFICATION

2.1 SOFTWARE REQUIREMENTS

Frontend-HTML, CSS, PHP, JS, Bootstrap

Backend-PHP/MYSQL

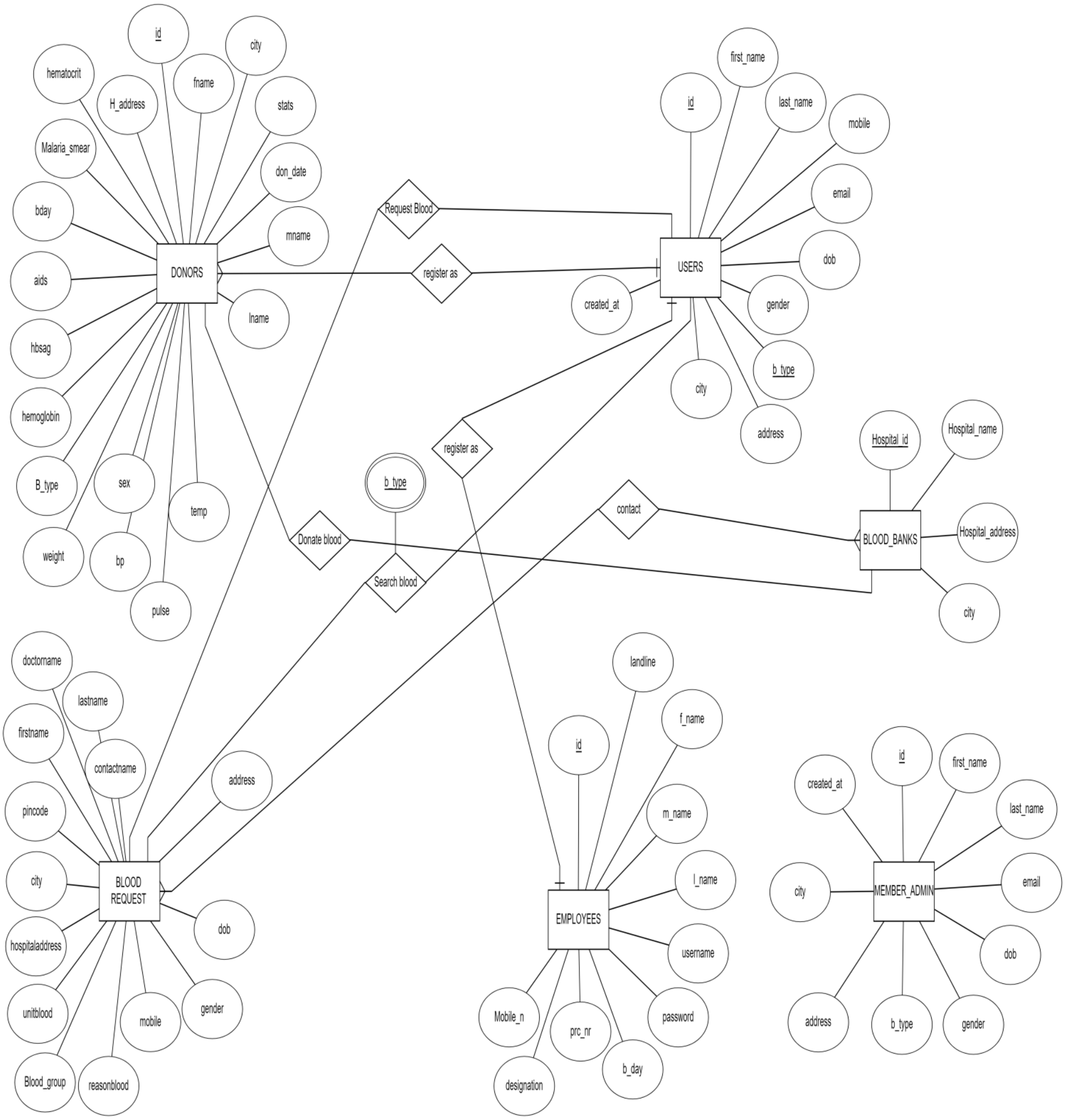
1. Operating System: Windows 10
2. Google Chrome
3. WAMP (Version-3.1.9)
4. PHP editor: Sublime text 3

2.2 HARDWARE REQUIREMENTS

1. Computer with a 2.3 GHz or faster processor
2. Minimum 4GB of RAM or more
3. 1000 GB of available hard-disk space
4. System Type 64bit operating system
5. 1366 × 768 or higher-resolution display
6. DVD-ROM drive

3. PROJECT DESIGN

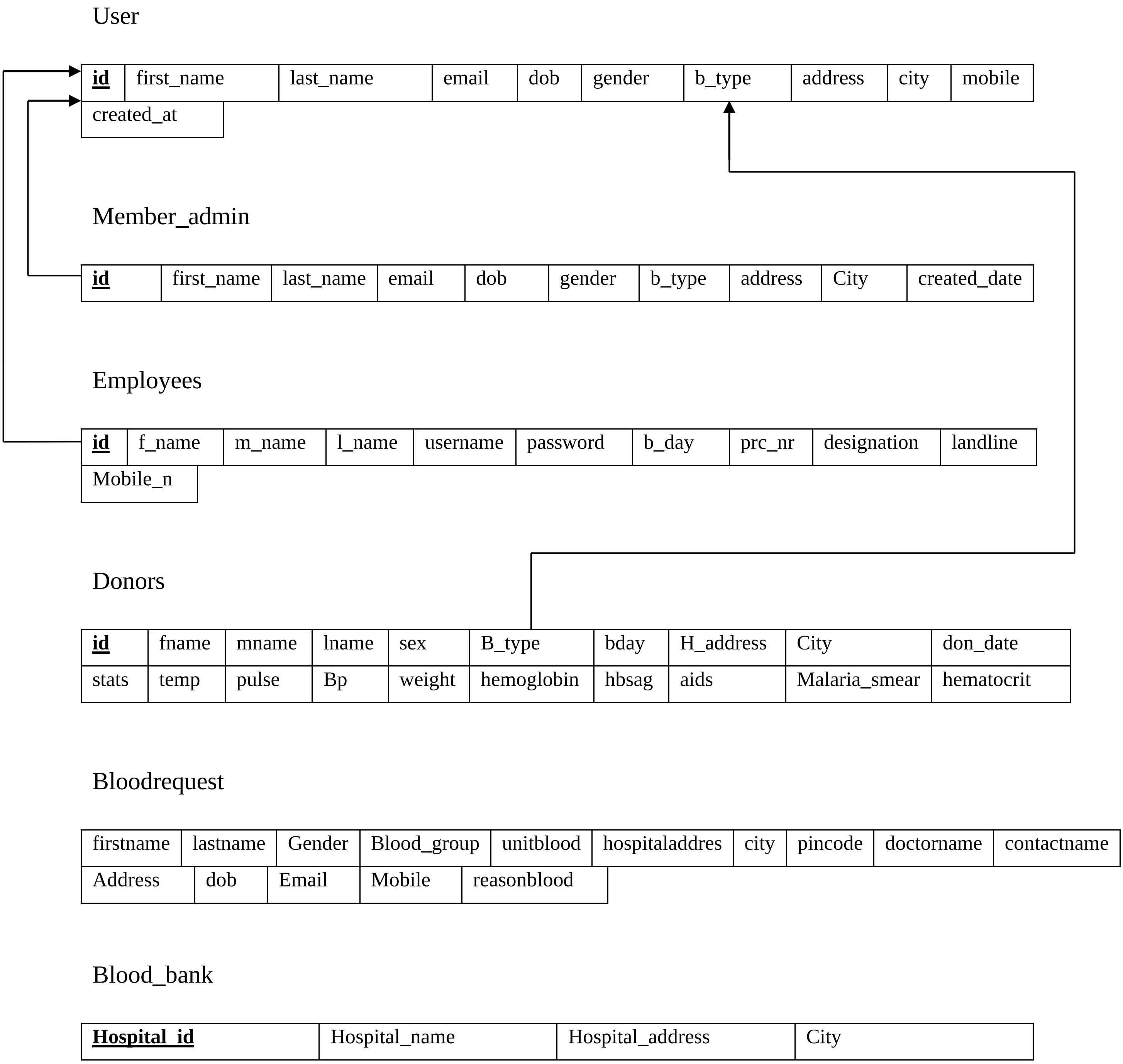
3.1E-R DIAGRAM



(b)

3.2 SCHEMA DIAGRAM

A database schema is the skeleton structure that represents the logical view of the entire database. A database schema defines its entities and the relationship among them.



User

id	first_name	last_name	email	dob	gender	b_type	address	city	mobile
created_at									

Member_admin

id	first_name	last_name	email	dob	gender	b_type	address	City	created_date

Employees

id	f_name	m_name	l_name	username	password	b_day	prc_nr	designation	landline
Mobile_n									

Donors

id	fname	mname	lname	sex	B_type	bday	H_address	City	don_date
stats	temp	pulse	Bp	weight	hemoglobin	hbsag	aids	Malaria_smear	hematocrit

Bloodrequest

firstname	lastname	Gender	Blood_group	unitblood	hospitaladdres	city	pincode	doctorname	contactname
Address	dob	Email	Mobile	reasonblood					

Blood_bank

Hospital_id	Hospital_name	Hospital_address	City

4. IMPLEMENTATION

4.1 INTRODUCTION

PHP, which stands for “PHP: Hypertext Pre-processor” is a widely-used Open Source general purpose scripting language that is especially suited for Web development and can be embedded into HTML. Its syntax draws upon C, Java, and Perl, and is very to lean. The main goal of language is to allow web developers to write dynamically generated web pages quickly, but you can do much more with PHP. PHP is a powerful language and the interpreter, whether included in a web server as a module or executed as a separate CGI binary, is able to access files, execute commands and open network connections on the server. These properties make anything run on a web server insecure by default.

PHP is designed specifically to be a more secure language for writing CGI programs than Perl or C, and with correct selection of compile-time and runtime configuration options, and proper coding practices, it can give you exactly the combination of freedom and security you need. As there are many different ways of utilizing PHP, there are many configuration options controlling its behaviour.

A large selection of options guarantees you can use PHP for a lot of purpose, but it also means there are combinations of these options and server configurations that result in an insecure setup. The configuration flexibility of PHP is equally rivalled by the code flexibility. PHP can be used to build complete server applications, with all the power of a shell user, or it can be used for simple server/side includes with little risk in a tightly controlled environment.

BACK END (MySQL)

DATABASE

A Database Management System (DBMS) is computer software designed for the purpose of managing databases, a large set of structured data, and run operations on the data requested by numerous users. Typical examples of DBMSs include Oracle, DB2, Microsoft Access, Microsoft SQL Server, Firebird, PostgreSQL, MySQL, SQLite, FileMaker and Sybase Adaptive Server Enterprise. DBMSs are typically used by Database administrators in the creation of Database systems.

Typical examples of DBMS use include accounting, human resources and customer support systems. Originally found only in large companies with the computer hardware needed to support large data sets, DBMSs have more recently emerged as a fairly standard part of any company back office.

A DBMS is a complex set of software programs that controls the organization, storage, management, and retrieval of data in a database.

MYSQL is a open source relational database management system (RDBMS) based on Structured Query Language (SQL)

SQL

Structured Query Language (SQL) is the language used to manipulate relational databases. SQL is tied very closely with the relational model.

- In the relational model, data is stored in structures called relations or tables.

SQL statements are issued for the purpose of:

- Data definition: Defining tables and structures in the database (DDL used to create, alter and drop schema objects such as tables and indexes).

SERVER

- **WampServer** is a Web development platform on Windows that allows you to create dynamic Web applications with Apache2, PHP, MySQL and MariaDB. **WampServer** automatically installs everything you need to intuitively develope Web applications. You will be able to tune your server without even touching its setting files.

The full form of WAMP stands for Windows, Apache, Mysql, and PHP.

4.2 SOURCE CODE

4.2.1 FRONT END

HTML (Hyper Text Markup Language) is the standard markup language for creating web pages and web applications.

LOGIN CODE:

```
<div class="container">
<div class="col-md-4"></div>
<div class="col-md-4">
<?php if(isset($invalid)): ?>
<div class="alert-danger" id="invalid"><?= $invalid; ?></div>
<?php endif; ?>
<div class="panel panel-primary">
<div class="panel-heading">
<div class="col-md-4">
<img src="assets/security-icon.png" class="img img-responsive img-thumbnail">
</div>
<h3>Admin Login</h3>
</div>
<div class="panel-body">
<form class="form-vertical" role="form" method="post" action="index.php">
<div class="form-group">
<input type="text" class="form-control" required="true" name="username"
placeholder="Username">
</div>
<div class="form-group">
<input type="password" required="true" class="form-control" name="password"
placeholder="Password">
</div>
<div class="form-group loginBtn">
<button type="submit" name="loginBtn" class="btn btn-primary btn-sm">Login</button>
</div>
</form>
</div>
```

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

4.2.2 BACKEND CODE

PHP is the server side scripting language designed primarily for web development but also used as a general purpose programming language.

```
<?php
require_once 'php/DBConnect.php';
$db = new DBConnect();
$db->checkAuth();

$invalid = NULL;
if(isset($_POST['loginBtn'])){
$username = $_POST['username'];
$password = $_POST['password'];

if($username == "root"){
if($password == "srikar"){
session_start();
$_SESSION['username'] = $username;
header("Location: http://localhost/online-blood-bank-management-system-in-php-master/admin/home.php");
} else {
$invalid = "Invalid Password!";
}
}else{
$invalid = "Invalid username or password!";
}
}

$title="Admin Login";
include 'layout/_header.php';

?>
```

4.3 TRIGGER

A trigger is a special kind of stored procedure that automatically executes when an event occurs in database server.

The trigger is used in our project is as follows:

TRIGGERNAME	: empbackup
TABLE	: employees
TIME	: AFTER
EVENT	: DELETE

DEFINITION

INSERT INTO employeebackup VALUES (OLD.id,OLD.f_name,OLD.m_name,OLD.l_name,OLD.username,OLD.password,OLD.b_day,OLD.prc_nr,OLD.designation,OLD.landline,OLD.mobile_nr)

DEFINER

root@localhost

4.4 STORED PROCEDURE

A stored procedure is a set of Structured Query Language (SQL) statements with an assigned name, which are stored in a relational database management system such that it can be reused and shared by multiple programs.

The stored procedure used for this project is as follows:

ROUTINE NAME	: getBlood
TYPE	: PROCEDURE
DIRECTION	: IN
PARAMETER	: b_type

DEFINITION

SELECT * FROM users WHERE users b.b_type = b_type

DEFINER

‘root’@’localhost’

SECURITY TYPE

DEFINER

5. TESTING

5.1 SOFTWARE TESTING

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques includes the process of executing a program or applications with the intent of finding software bugs(errors or other defects).

It involves the execution of a software component or system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test:

- Meets the requirements that guided its design and development.
- Responds correctly to all kinds of inputs.
- Performs its functions within an acceptable time.
- Is sufficiently usable.
- Can be installed and run in its intended environments.
- Achieves the general results its stakeholders desire.

As the number of possible tests for even simple software components is practically infinite, all software testing uses some strategy to select tests that are feasible for the available time and resources. As a results, software testing typically(but not exclusively)attempts to execute a program or application with the intent of finding software bugs(errors or other defects).

Software testing can provide objective, independent information about the quality of software and risk of its failure to users and/or sponsors.

Software testing can be conducted as soon as executable software (even if partially complete) exists. The overall approach to software development often determines when and how testing is conducted. For example, in a phased process, most testing occurs after system requirements have been defined and then implemented in testable programs. In contrast, under an Agile approach, requirements, programming, and testing are often done concurrently.

5.2 TESTING PROCESS

5.2.1 LEVELS OF TESTING

- UNIT TESTING: Unit testing, also known as component testing, refers to tests that verify the functionality of a specific section of code,usually at the function level.
- INTEGRATION TESTING: Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design.
- COMPONENT INTERFACE TESTING: The practice of component interface testing can be used to check the handling of data passed between various units, or subsystem components, beyond full testing between those units.
- TESTING SYSTEM: System testing, or end-to-end testing, tests a complete;y integrated system to verify that it meets its requirements.

5.2.2 TESTING METHODS

5.2.2.1 UNITS TESTING

Units testing, also known as component testing, refers to tests that verify the functionality of a specific section of code, usually at the function level. In an object-oriented environment, this is usually at the class level, and the minimal unit tests include the constructors and destructors.

These types of tests are usually written by developers as they work on code (white-box style), to ensure that the specific functions is working as expected. One function might have multiple tests, to catch corner cases or other branches in the code. Unit testing alone cannot verify the functionality of a piece of software, but rather is used to ensure that the building blocks of the software work independently from each other.

Unit testing is a software development process that involves synchronized application of a broad spectrum of defect prevention and detection strategies in order to reduce software development risks, time, and costs.

5.2.2.2 INTEGRATION TESTING

Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Software components may be integrated in an iterative way or all together (“big bang”). Normally the former is considered a better practice since it allows interface issues to be located more quickly and fixed.

Integration testing works to expose defects in the interfaces and interaction between integrated components (modules). Progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a system.

5.3 TEST ENVIRONMENT

BLACK BOX TEXTING

Black box testing treats the software as a “black box”, examining functionality without any knowledge of internal implementation. The testers are only aware of what the software is supposed to do, not how it does it. Black-box testing methods includes equivalence partitioning, boundary value analysis, all-pairs testing , state transition tables, decision table testing, fuzz testing, model-based testing, use case testing, exporatory testing and specification-based testing.

WHITE BOX TESTING

White-box testing tests internal structures or workings of program, as opposed to the functionality exposed to the end-user. In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriats outputs. While white-box testing can be applied at the unit, integration and system levels of the software testing process, it is usually done at the unit level. It can test paths within a unit, paths between units during integration, and between subsystem during a system-level test. Through this method of test design can uncover many errors or problems; it might not detect unimplemented parts of the specification or missing requirements.

6. USER INTERFACE

Fig 6.1 Home page:

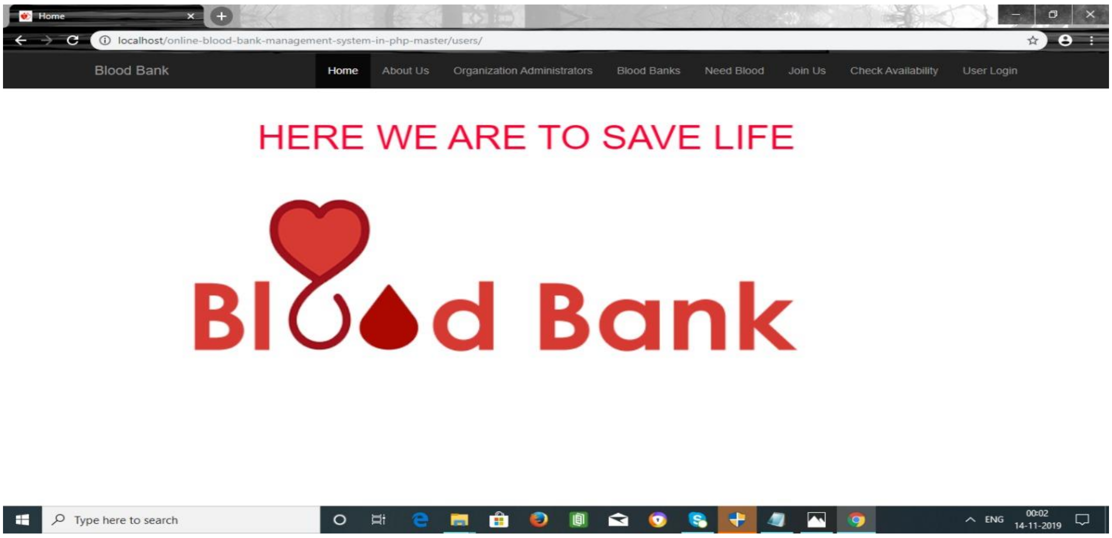


Fig 6.2 User Login page:

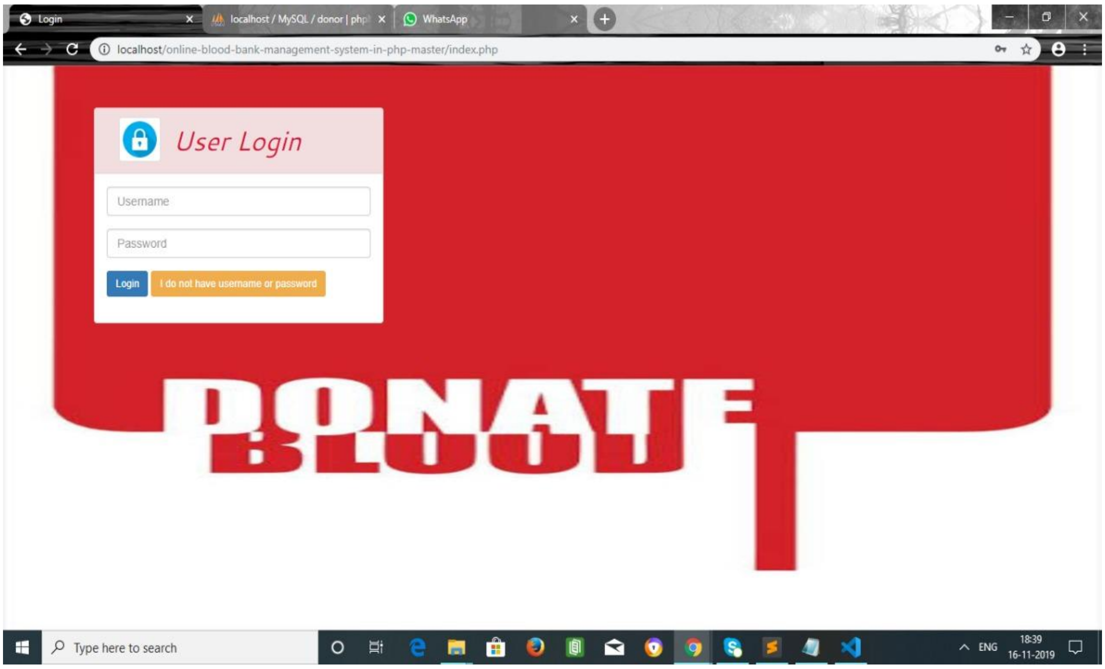
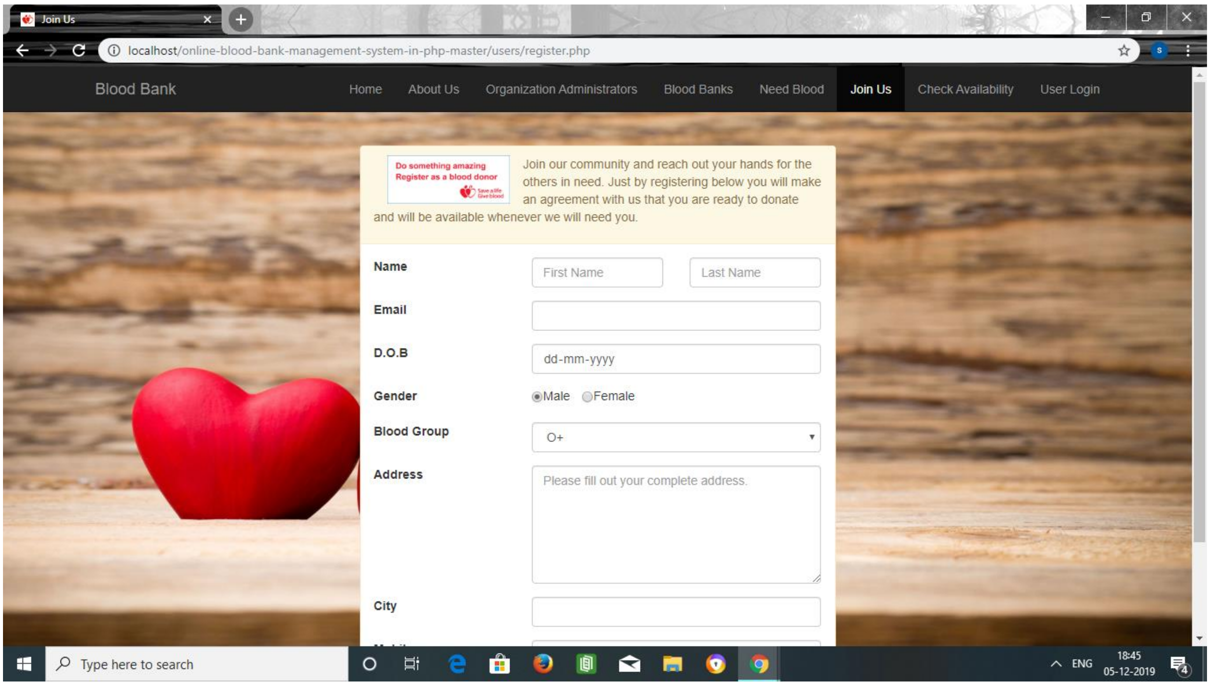

Login
localhost / MySQL / donor | php
WhatsApp
localhost/online-blood-bank-management-system-in-php-master/index.php
User Login
Username
Password
Login
I do not have username or password
DONATE
BLOOD
Type here to search
ENG
18:39
16-11-2019

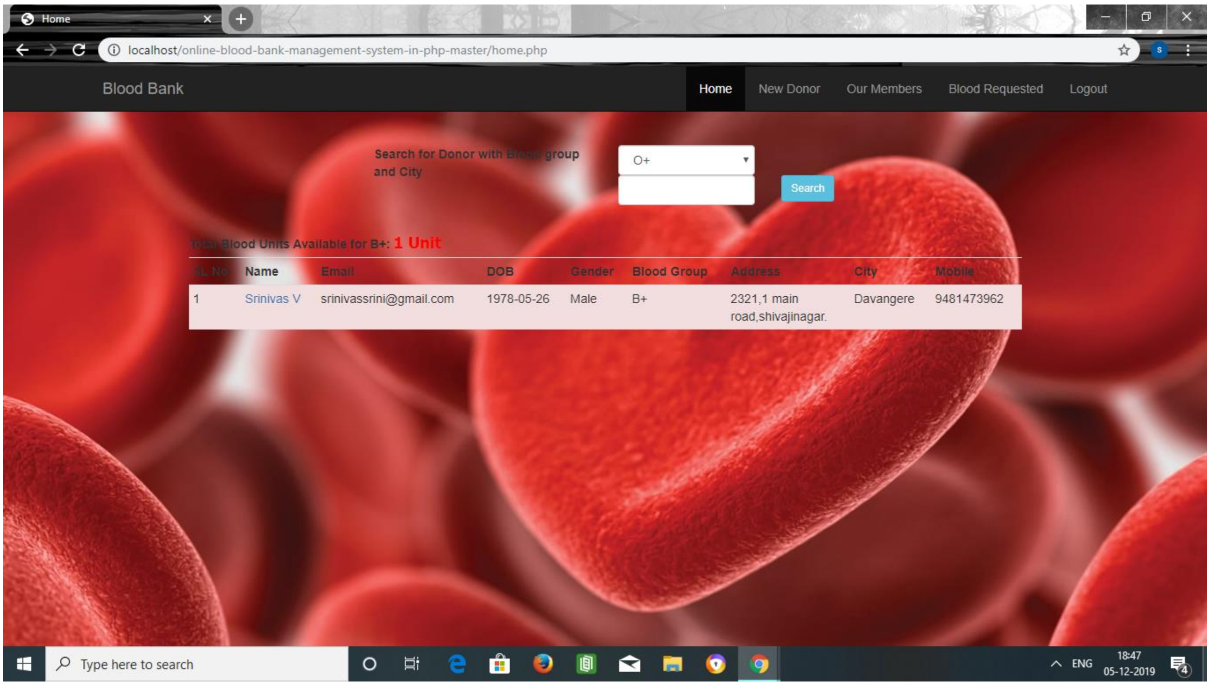
Fig 6.3 Join Us page:



Join our community and reach out your hands for the others in need. Just by registering below you will make an agreement with us that you are ready to donate and will be available whenever we will need you.

Name	First Name	Last Name
Email		
D.O.B	dd-mm-yyyy	
Gender	● Male ○ Female	
Blood Group	O+	
Address	Please fill out your complete address.	
City		

Fig 6.4 Check availability page(user):



Search for Donor with Blood group and City: O+

Total Blood Units Available for B+: 1 Unit

Sl. No	Name	Email	DOB	Gender	Blood Group	Address	City	Mobile
1	Srinivas V	srinivassrini@gmail.com	1978-05-26	Male	B+	2321,1 main road,shivajinagar.	Davangere	9481473962

Fig 6.5 Admin login page:

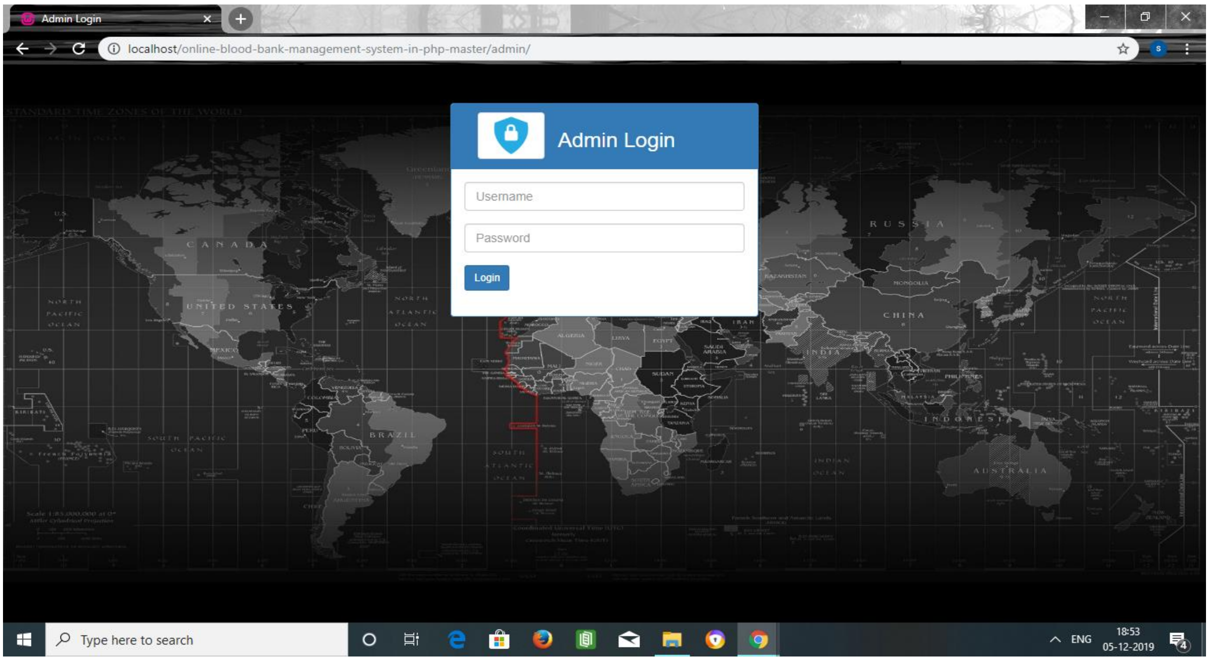


Fig 6.6 Add Employee page:

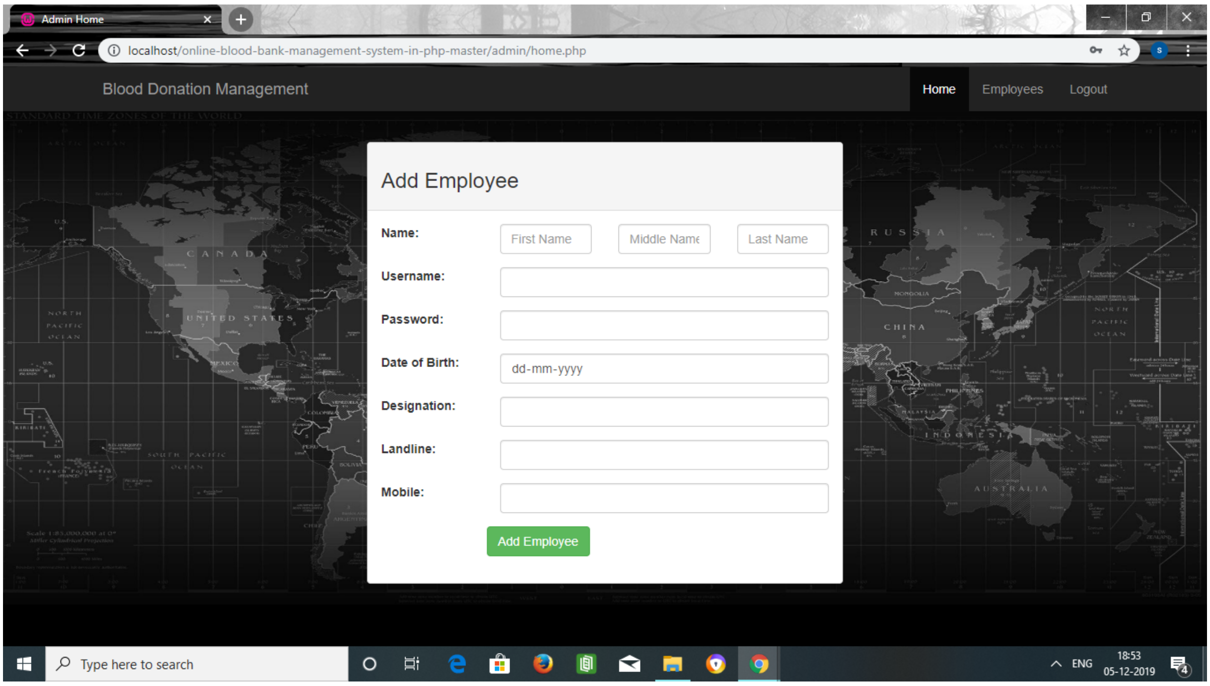
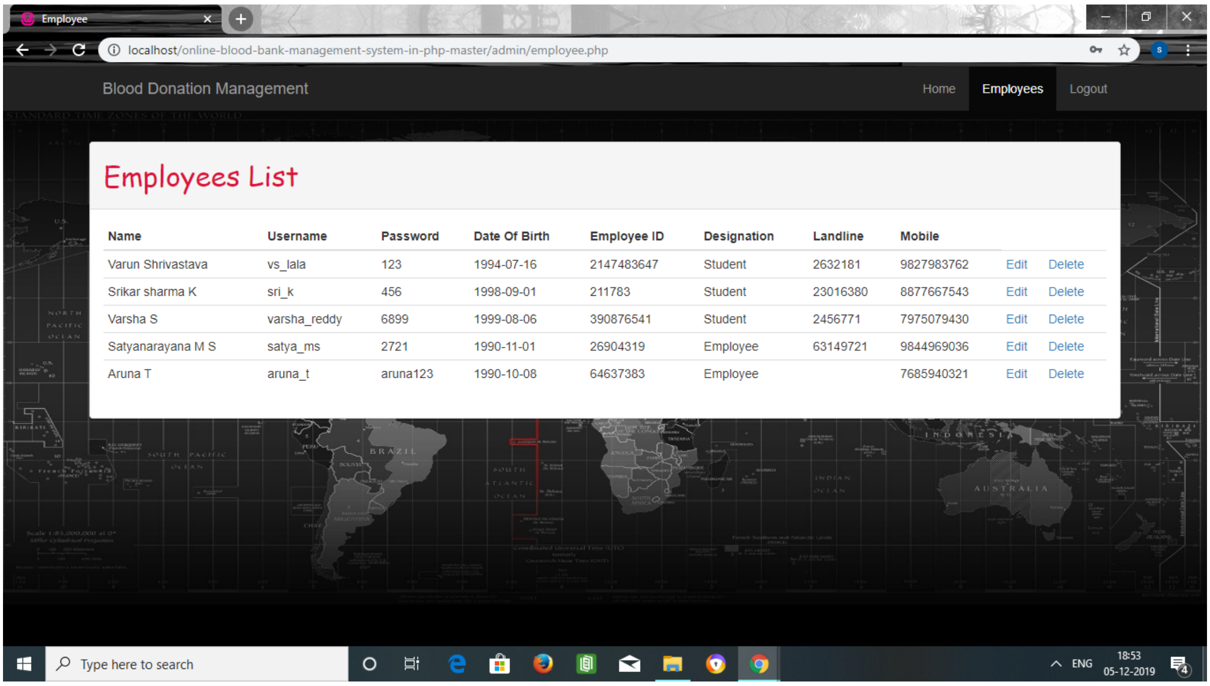


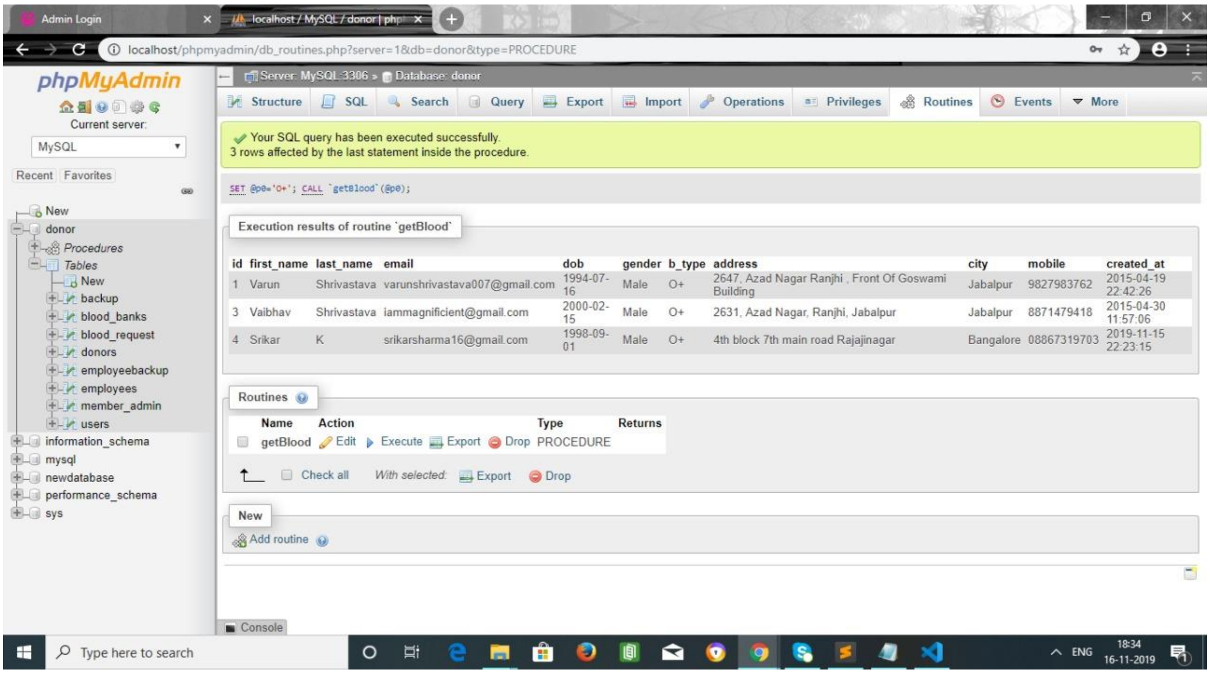
Fig 6.7 Triggers page:


Employee
localhost/online-blood-bank-management-system-in-php-master/admin/employee.php
Blood Donation Management
Home
Employees
Logout
Type here to search
ENG
18:53
05-12-2019

Employees List

Name	Username	Password	Date Of Birth	Employee ID	Designation	Landline	Mobile		
Varun Shrivastava	vs_lala	123	1994-07-16	2147483647	Student	2632181	9827983762	Edit	Delete
Srikar sharma K	sri_k	456	1998-09-01	211783	Student	23016380	8877667543	Edit	Delete
Varsha S	varsha_reddy	6899	1999-08-06	390876541	Student	2456771	7975079430	Edit	Delete
Satyanarayana M S	satya_ms	2721	1990-11-01	26904319	Employee	63149721	9844969036	Edit	Delete
Aruna T	aruna_t	aruna123	1990-10-08	64637383	Employee		7685940321	Edit	Delete

Fig 6.8 Stored procedure page:



Your SQL query has been executed successfully.
3 rows affected by the last statement inside the procedure.

```
SET @p0='O+'; CALL `getBlood`(@p0);
```

Execution results of routine `getBlood`

id	first_name	last_name	email	dob	gender	b_type	address	city	mobile	created_at
1	Varun	Shrivastava	varunshrivastava007@gmail.com	1994-07-16	Male	O+	2647, Azad Nagar Ranjhi , Front Of Goswami Building	Jabalpur	9827983762	2015-04-19 22:42:26
3	Vaibhav	Shrivastava	iammagnificient@gmail.com	2000-02-15	Male	O+	2631, Azad Nagar, Ranjhi, Jabalpur	Jabalpur	8871479418	2015-04-30 11:57:06
4	Srikar	K	srikarsharma16@gmail.com	1998-09-01	Male	O+	4th block 7th main road Rajajinagar	Bangalore	08867319703	2019-11-15 22:23:15

Routines

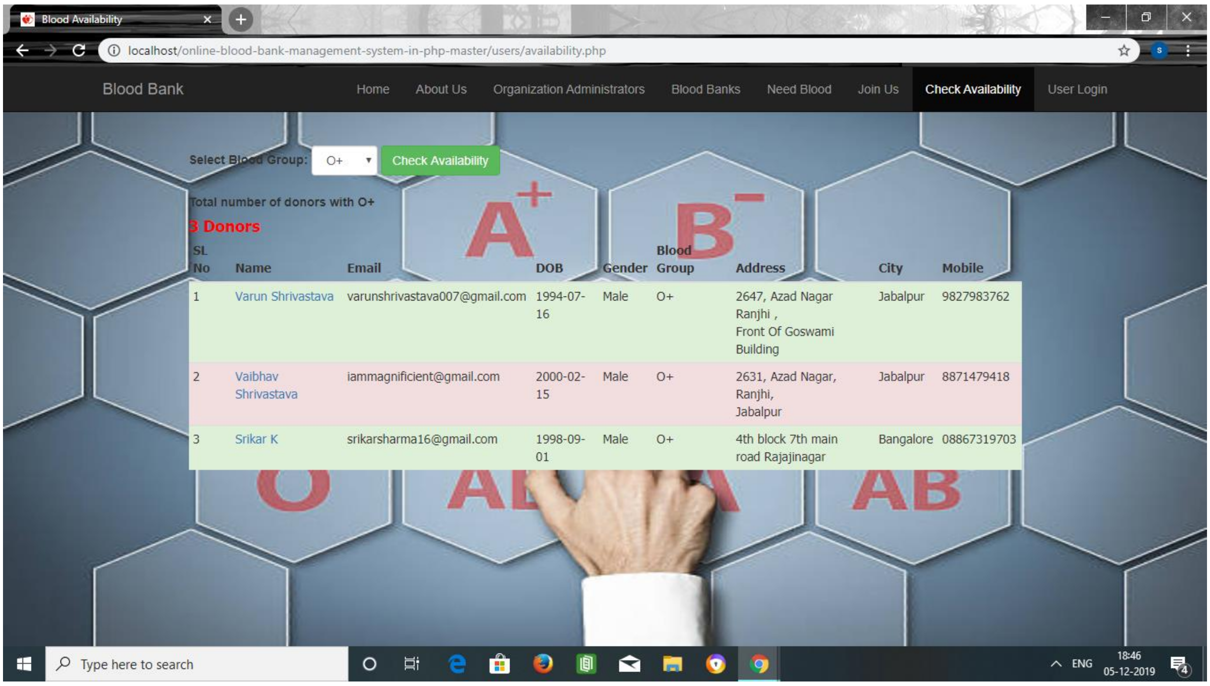
	Name	Action	Type	Returns
□	getBlood	Edit Execute Export Drop	PROCEDURE	

□ Check all With selected: Export Drop

New

Add routine

Fig 6.9 Check availability page(home):



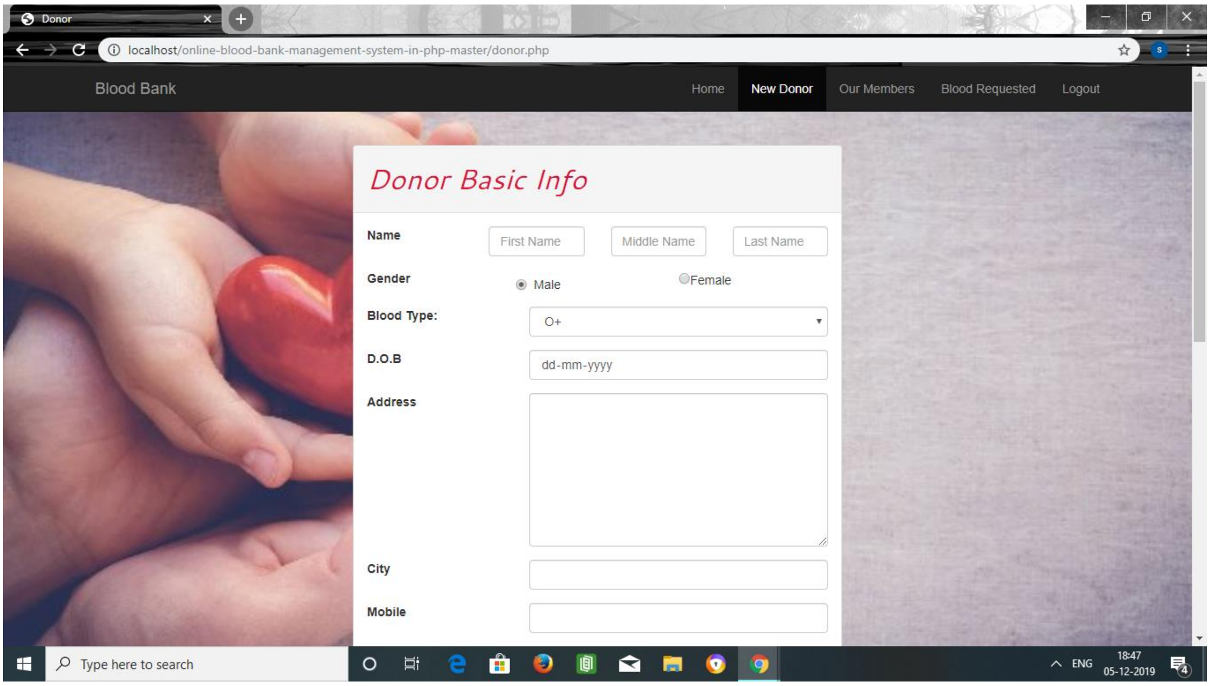
Select Blood Group: O+ Check Availability

Total number of donors with O+

3 Donors

SL No	Name	Email	DOB	Gender	Blood Group	Address	City	Mobile
1	Varun Shrivastava	varunshrivastava007@gmail.com	1994-07-16	Male	O+	2647, Azad Nagar Ranjhi , Front Of Goswami Building	Jabalpur	9827983762
2	Vaibhav Shrivastava	iammagnificient@gmail.com	2000-02-15	Male	O+	2631, Azad Nagar, Ranjhi, Jabalpur	Jabalpur	8871479418
3	Srikar K	srikarsharma16@gmail.com	1998-09-01	Male	O+	4th block 7th main road Rajajinagar	Bangalore	08867319703

Fig 6.10 Donor information page:



Donor Basic Info

Name: First Name | Middle Name | Last Name

Gender: ● Male ○Female

Blood Type: O+

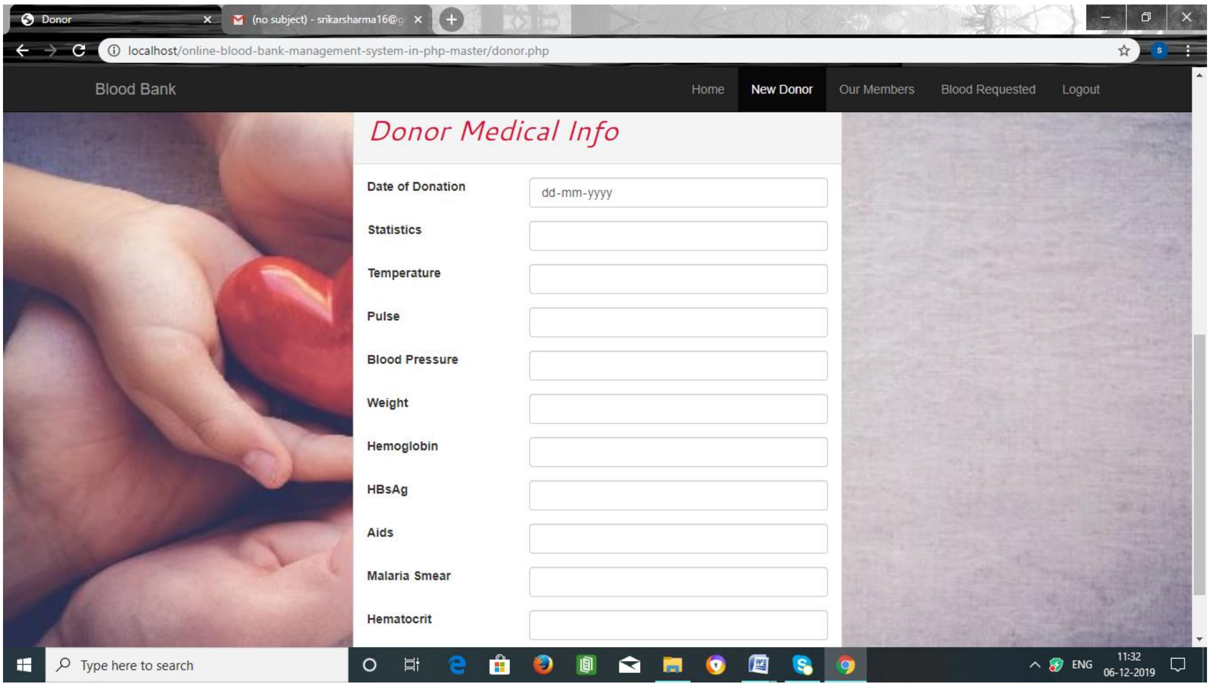
D.O.B: dd-mm-yyyy

Address

City

Mobile

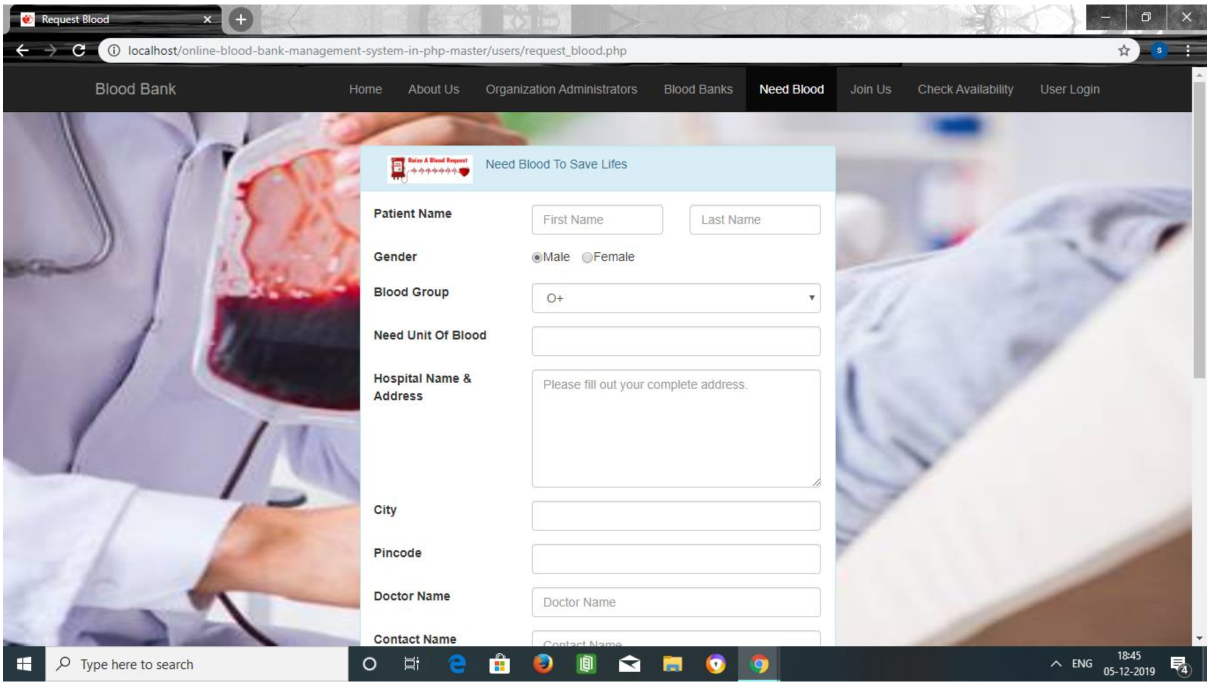
Fig 6.11 Medical information page:


Donor
(no subject) - srikarsharma16@g
localhost/online-blood-bank-management-system-in-php-master/donor.php
Blood Bank
Home
New Donor
Our Members
Blood Requested
Logout
Type here to search
ENG
11:32
06-12-2019

Donor Medical Info

Date of Donation	dd-mm-yyyy
Statistics	
Temperature	
Pulse	
Blood Pressure	
Weight	
Hemoglobin	
HBsAg	
Aids	
Malaria Smear	
Hematocrit	

Fig 6.12 Request page:



Need Blood To Save Lifes

Patient Name	First Name Last Name
Gender	● Male ○ Female
Blood Group	O+
Need Unit Of Blood	
Hospital Name & Address	Please fill out your complete address.
City	
Pincode	
Doctor Name	Doctor Name
Contact Name	Contact Name

7. CONCLUSION

An online based blood bank management system for a blood bank will allow donors to register more easily for donation of blood, hence making the emergency situations to be tackled in an efficient way.

Also, it is easier to find the number of people donating a particular blood group, or the number of blood units available in one blood bank, or the number of donors participating in an blood donation.

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" were 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 project from a personal point of view also helped us in understanding the following aspects of project development:

.

8. FUTURE ENHANCEMENTS

1. Adaara card linking with the website.

2. Mapped with Insurance Companies for claim processing.

3. Billing of patients.

4. Blood Bank Information Management.

5. Video Conferencing facility for remote areas for donation.

6. Hangout for different doctors and patients at different locations.

REFERENCES

1. Fundamentals of database systems 7th Edition, Ramez Elmasri and Shamkant B. Navathe .

2. Database Management System, Ramakrishnan, and Gehrke, 3rd Edition, 2014, McGraw Hill.

3. Silberschatz Korth and Sudharshan, Database System Concepts, 6th Edition, McGrawHill, 2013.

4. Coronel, Morris, and Rob, Database Principles Fundamentals of Design, Implementations and Management, Cengage Learning 2012.

5. PHP: The Complete Reference 1st Edition (English, Paperback, Steven Holzner).