ONLINE BLOOD BANK MANAGEMENT SYSTEM

A Project Report Submitted to the **BHARATHIDASAN UNIVERSITY**

In partial fulfillment of requirements for the award of the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

Submitted by

HAKKIM.S

(REG.NO.:182510)

Under the guidance of

Mr. R. MANIVASAGAN., MCA., M.Phil.,

Assistant Professor of Computer Science

PG AND RESEARCH DEPARTMENT OF COMPUTER SCIENCE

GOVERNMENT ARTS COLLEGE (AUTONOMOUS)

(Affiliated to Bharathidasan University and Accredited by NAAC with A Grade)

KARUR-639 005

AUGUST - 2020

CERTIFICATE

This is to certify that the project report entitled **"ONLINE BLOOD BANK MANAGEMENT SYSTEM"** is done by **HAKKIM.S (Reg. No.: 182510)** in partial fulfillment of the requirement for the award of degree of M.Sc., Computer Science for the academic year 2019-2020 is the original work done by the candidate under my guidance and submitted to **Bharathidasan University**, Trichy, through **Government Arts College (Autonomous)**, **Karur-5.**

Signature of the Guide **Signature of the HOD**

Submitted for Viva – Voce Examination held on _____

Signature of the Internal Examiner **Signature of the External Examiner**

DECLARATION

I hereby declare that this project entitled **"ONLINE BLOOD BANK MANAGEMENT SYSTEM" s**ubmitted to the **BHARATHIDASAN UNIVERSITY**, Trichy, in partial fulfillment of the requirement for the award of the degree of **MASTER OF SCIENCE IN COMPUTER SCIENCE** is a record of original work done by me and this is not submitted anywhere else for any other degree.

Signature of the Candidate

HAKKIM.S

(Reg. No : 182510)

Date:

Place:

ACKNOWLEDGEMENT

First & Foremost I acknowledge my gratitude to the almighty who sustained me with the powerful blessings through various persons at all stages of my project work.

I wish to express my sincere gratitude to the Principal **Dr. S. KOUSALYA DEVI., M.Sc., M.Phil. , PGDCA., Ph.D.,** Government Arts College (Autonomous), Karur-5 for providing necessary facilities to carry out this project.

I express my whole heartful thanks to **Dr. K. THANGADURAI., M.Sc., M.C.A., M.Phil., Ph.D.,** Head & Assistant Professor, Department of Computer Science, Government Arts College (Autonomous), Karur-5 for his Inspiration and assistance for completing this project.

I am grateful thanks to my project guide **Mr. R. MANIVASAGAN., M.C.A., M.Phil.,** Assistant Professor, Department of Computer Science, Government Arts College (Autonomous), Karur-5 for his guidance and valuable suggestions.

I express my heartfelt gratitude to all the staff members and programmer of Computer Science Department for their timely help.

Finally I also express my sincere thanks to my beloved parents, friends for their cooperation in finishing my project work successfully.

TABLE OF THE CONTENT

ABSTRACT

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.

CHAPTER 1

1.1 ABOUT THE PROJECT

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.

Blood bank is a place where blood bag that is collected from blood donation events is stored in one place. The term “blood bank” refers to a division of a hospital laboratory where the storage of blood product occurs and where proper testing is performed to reduce the risk. It processes blood that will be supplied to the patients in HSNZ according to their needs. Before the blood is supplied to the patients, the blood will undergo several tests to ensure that the blood receiver is not infected by serious diseases.

Blood Distribution Unit every month, HSNZ will organize blood donation event which is one of the way they can increase the blood stock. After the blood donation events, the blood bags that they obtained will undergo tests. All of the blood

received at the blood donation events must be managed thoroughly and systematically to avoid patient who need the blood infected by any viruses or diseases.

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

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
- Blood Recipient
- Blood collection
- Camp
- Stock details
- blood bank Reports
- Blood issued

CHAPTER 2

SYSTEM SPECIFICATION

2.1 HARDWARE REQUIREMENT:

Processor	:	Intel Pentium (Octa-Core)
Hard Disk	:	500 GB
RAM	:	2 GB
Processor Speed	:	2.10 GHz
Monitor	:	[1366 x 768 Display]
Mouse	:	Any Standard
Keyboard	:	Any Standard

2.2 SOFTWARE REQUIREMENT:

Operating System	:	Windows 7
IDE	:	Wamp Server
Server	:	Apache Server
Front-end	:	HTML ,PHP,JS and CSS
Back-end	:	My-SQL

2.3. ABOUT FRONT END:

The front end is an interface between the user and the back end. The front and back ends may be distributed amongst one or more systems.

In network computing, front end can refer to any hardware that optimizes or protects network traffic. It is called application front-end hardware 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, the front end translates a computer programming source code into an 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 section that deals with source code and the back end that generates code and optimizes.

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 , 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

Software Discription & Tools Used:

1. PHP:-

Introduction

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.

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.

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.

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.

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.

HTML FORMS

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.

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

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).

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.

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.

2.4 ABOUT 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.

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, 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.

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) 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.

e) 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 3

3.1 EXISTING SYSTEM

- Cannot Upload and Download the latest updates.
- No use of Web Services and Remoting.
- Risk of mismanagement and of data when the project is under development.
- Less Security.
- No proper coordination between different Applications and Users.
- Fewer Users – Friendly

Disadvantages

1. User friendliness is provided in the application with various controls.
2. The system makes the overall project management much easier and flexible.
3. Readily upload the latest updates, allows user to download the alerts by clicking the URL.
4. There is no risk of data mismanagement at any level while the project development is under process.
5. It provides high level of security with different level of authentication.

3.2 PROPOSSED SYSTEM

To debug the existing system, remove procedures those cause data redundancy, make navigational sequence proper. To provide information about audits on different level and also to reflect the current work status depending on organization/auditor or date. To build strong password mechanism.

Advantages:

- User friendliness I provided in the application with various controls.
- The system makes the overall project management much easier and flexible.
- Readily upload the latest updates , allows user to download the alerts by clicking the url .
- It provides high level of security with different level of authentication.

CHAPTER 4

4.1 MODULES:-

- **Donor registration**
- **Blood Request**
- **Search donors**
- **Contact Us**
- **Admin login**
- **Blood Donors list**
- **Blood Requesters list**
- **Purchase Blood**
- **Sale Blood**
- **Blood Stack**

4.2 MODULE DISCRIPTION:

BLOOD DONATION is a website based on PHP. The purpose of this project was to develop a blood management information system to assist in the management of blood donor records and ease or control the distribution of blood in various part of country basing on the hospitals demand. . This project includes mainly two modules i.e. login and main page.

- **Login:**

Admin

The page require user name and password to start the application. Login is a process by which individual access to a computer system is controlled by identifying and authenticating the user through the cardinalities presented by the user. Admin can add update or delete the user, city, state, camp etc.

Admin are of a two type:

Admin level user

General level user

Admin level user:

Admin level user are a admin user, if he/she login a admin panel they work on a all the pages of the admin site like add user, update user, delete user, add city, update city, delete city etc.

General level user:

General level user are a general user, if he/she login a admin panel they don not see all the pages of the admin site like add user, update user, delete user, add city, update city, delete city etc. They work on only a addition of the new user, city, state, camp etc.

Main Window:-

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.

Registration Page:

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.

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. After the donation, the donor is given a donor identification card with their name, blood

type and a barcode to be used as a reference for future donations . The barcode is used to retrieve the donor's record containing their personal information, medical history and donation information, including blood results .Only blood bank administrators have the authority to access the donor's records, since the system is only available for their use within the organization. This makes it difficult for donors to make changes to their personal information within the system. That is, for donors to update their personal information, such as their phone number, mailing address, or e-mail, they cannot update the information by themselves, but have to contact the blood bank center to update their information . At the back the card is a table that contains number of donations, date, location ,and the blood collector's signature. Existing donors can submit their donor ID cards to retrieve their personal information and donation records and start the blood donation process, and they will be given a new card after they have donated blood for a total of eight times. Having a donor ID card may be a tangible reminder to people that they are helping lives as a blood donor; however, possessing a physical card comes with drawbacks such as loss or damage. To ensure donors can still identify themselves with the system, other credentials, such as username and password, can be used as a safeguard if their donor ID card is lost or damaged. If the donated blood is disqualified, the donor will be notified through postal mail that their blood component is reactive to viruses , meaning that there is a positive result of the blood being infected, and the organization will also inform the donor to perform another blood test at the blood bank to confirm the result of blood. If the blood is qualified, the administrator then will deposit the blood into the inventory for future requests.

4.3 Input and Output Design:

BLOOD DONATION is a website based on PHP. The purpose of this project was to develop a blood management information system to assist in the management of blood donor records and ease or control the distribution of blood in various part of country basing on the hospitals demand. . This project includes mainly two modules i.e. login and main page.

- **Login:**

Admin

The page require user name and password to start the application. Login is a process by which individual access to a computer system is controlled by identifying and authenticating the user through the cardinalities presented by the user. Admin can add update or delete the user, city, state, camp etc.

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
- Blood Recipient
- Blood collection
- Camp
- Stock details
- Blood issued

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.

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.

Camp

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.

Search

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

Blood groups name

- A-
- A+
- B-

- B+
- AB+
- AB-
- O+
- O-

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.

Update Profile

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

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.

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.

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.

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.

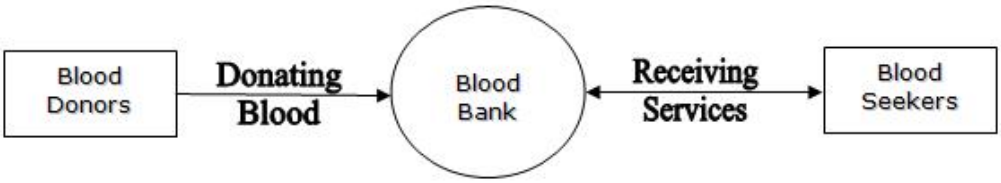
4.4 DATAFLOW DIAGRAM

INTRODUCTION OF DFD:-

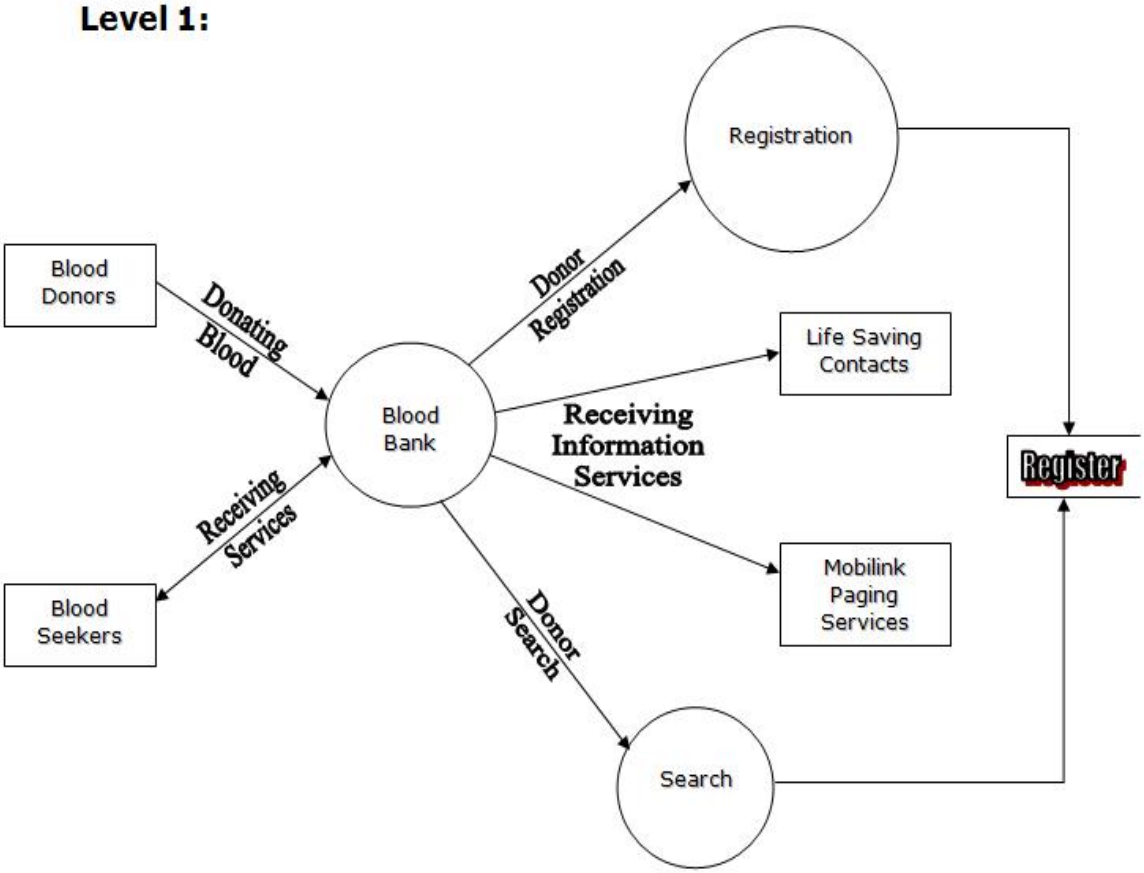
A DFD, in simple words, is a hierarchical graphical model of a system that shows the different processing activities or functions that the system performs and the data interchange among these functions. In the DFD terminology, it is useful to consider each function as a process that consumes some input data and produces some output data.

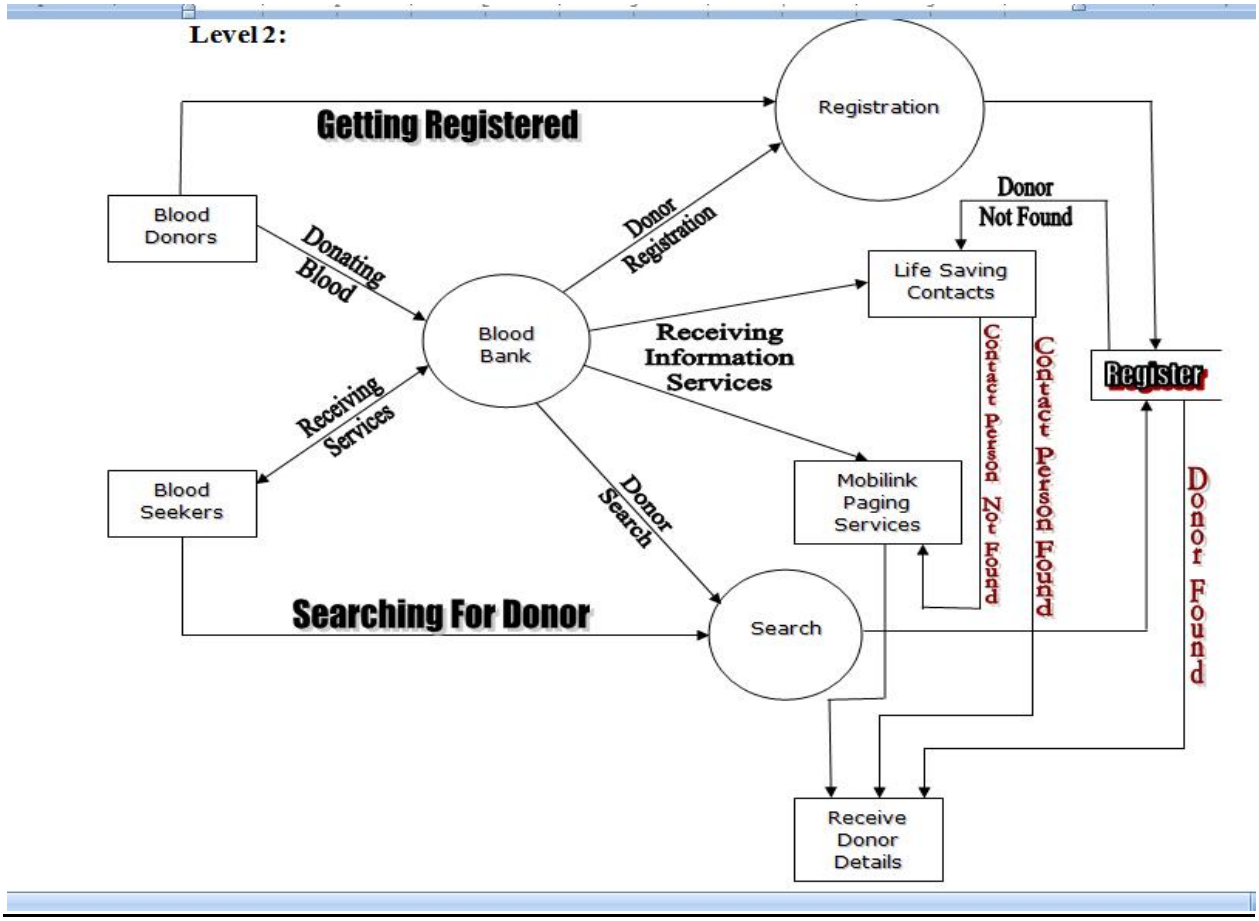
The DFD (also known as the bubble chart) is a simple graphical formalism that can be used to represent a system in terms of the input data to the system, various processing carried out on these data, and the output data generated by the system) The main reason why the DFD technique is so popular is probably because of the fact that DFD is a very simple formalism- it is simple to understand and use. A DFD model uses a very limited number of primitive symbols to represent the functions performed by a system and the data flow among these functions. Starting with a set of high-level functions that a system performs, a DFD model hierarchically represents various sub functions. In fact, any hierarchical model is simple to understand. Human mind is such that it can easily understand any hierarchical model of a system-because in a hierarchical model, starting with a very simple and abstract model of a system; different details of the system can be slowly introduced through different hierarchies.

Level 0:



Level 1:




Level 2:
Getting Registered
Registration
Blood Donors
Donating Blood
Donor Registration
Donor Not Found
Life Saving Contacts
Blood Bank
Receiving Information Services
Contact Person Not Found
Contact Person Found
Register
Receiving Services
Blood Seekers
Mobilink Paging Services
Donor Search
Donor Found
Searching For Donor
Search
Receive Donor Details

4.5 TABLE STRUCTURE:-

Donor Table:

Field name	Data type	Length
Donor_id	Int	10
Name	char	25
Father name	char	25
Gender	char	10
DOB	date	
Blood	varchar	15
Weight	int	10
Email	varchar	25
Address	varchar	100
Contact	int	10
Voluntary	text	
New_donor	varchar	15
Donor_pic	varchar	150
Status	int	10

Purchace Table:

Field name	Data type	Length
Blood_id	int	10
T_date	date	
T_qty	int	5
Branch	varchar	50
Type	varchar	2
T_id	int	10

Seeker Table:

Field name	Data type	Length
Id	**int**	**10**
Name	**char**	**25**
Gender	**char**	**10**
Blood	**varchar**	**10**
Bunit	**int**	**10**
Hospital Address	**varchar**	**200**
Rdate	**date**	
Cname	**char**	**25**
Caddress	**varchar**	**200**
Email	**varchar**	**20**
contact	**int**	**10**
Reason	**text**	
Pic	**varchar**	**100**
Status	**varchar**	**10**

Message Table:

Field name	**Data type**	**Length**
Id	**int**	**10**
Name	**char**	**25**
Contact	**text**	
Email	**varchar**	**20**
Message	**text**	
Status	**text**	

CHAPTER 5

5.1 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.

5.2 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.

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 may case; the selection of these test cases is based on the previous studies.

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.

INTEGRITY TESTING:

Integrity phases the entire module using the bottom-up approach and tested them. Integrity testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective was to take unit tested modules and build a program structure that has been dictated by design.

The testing strategy has two different approaches namely the top-down approach in which the integration is carried out from the top-level module to the bottom and the bottom-up approach in which the integration is carried out from the low-level module to the top.

The modules were tested using the bottom-up approach by introducing stubs for the top-level functions. This test was used to identify the errors in the interfaces, the errors in passing the parameters between the functions and to correct them

- **There are two types of testing available:**

Black Box Testing:-

In black box testing the structure of the program is not considered. Test cases are decided solely on the basis of the requirements or specifications of the program or module, and the consideration for selection of test classes. In this section, we will present some techniques for generating test cases for black-box testing. In black-box testing, the testing only knows the inputs that can be given to the system and what output the system can give. In other words the basis for deciding test case in functional testing is the requirement or specification of the system module. This form of testing is obvious functional or behavioral testing.The most obvious functional testing procedure is exhaustive testing, which as we have stated, is empirical. One criterion for generating test causes is to strategy has little chance of resulting in a set of test causes that is close to optimal(i.e. that detects the maximum errors with minimum test cases). Hence we need some other criterion or rule for selecting test cases. There are no formal rules for designing test cases for functional testing. In fact there are no precise criteria for selecting test cases however, there have been found to be very successful in detecting errors. Here we mention some of these techniques.

White Box Testing:-

In the previous section we discussed testing, which is concerned with the function that the tested program is proposed to program and does not deal with the internal structure of the program responsible for actually implementing that function. Those black-box testing is concerned with functionality rather than implementation of the program. On the other hand, white-box testing is concerned with testing

implementation of the program. The intent of this testing is not to exercise all the different input or output conditions(although that may be buy products) but to exercise the different programming structures and data structures used in the program. White-box testing is also called structure testing and we will use the two terms interchangeably to test the structures of a program, structural testing aims to achieve test cases that force the desired coverage of different structures various criteria have been proposed for this. Unlike the criteria for functional testing which are frequently imprecise, the criteria for structural testing are generally quite precise as they are based on program structures, which are formal and precise.

CHAPTER 6

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

- 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 7

BIBLIOGRAPHY

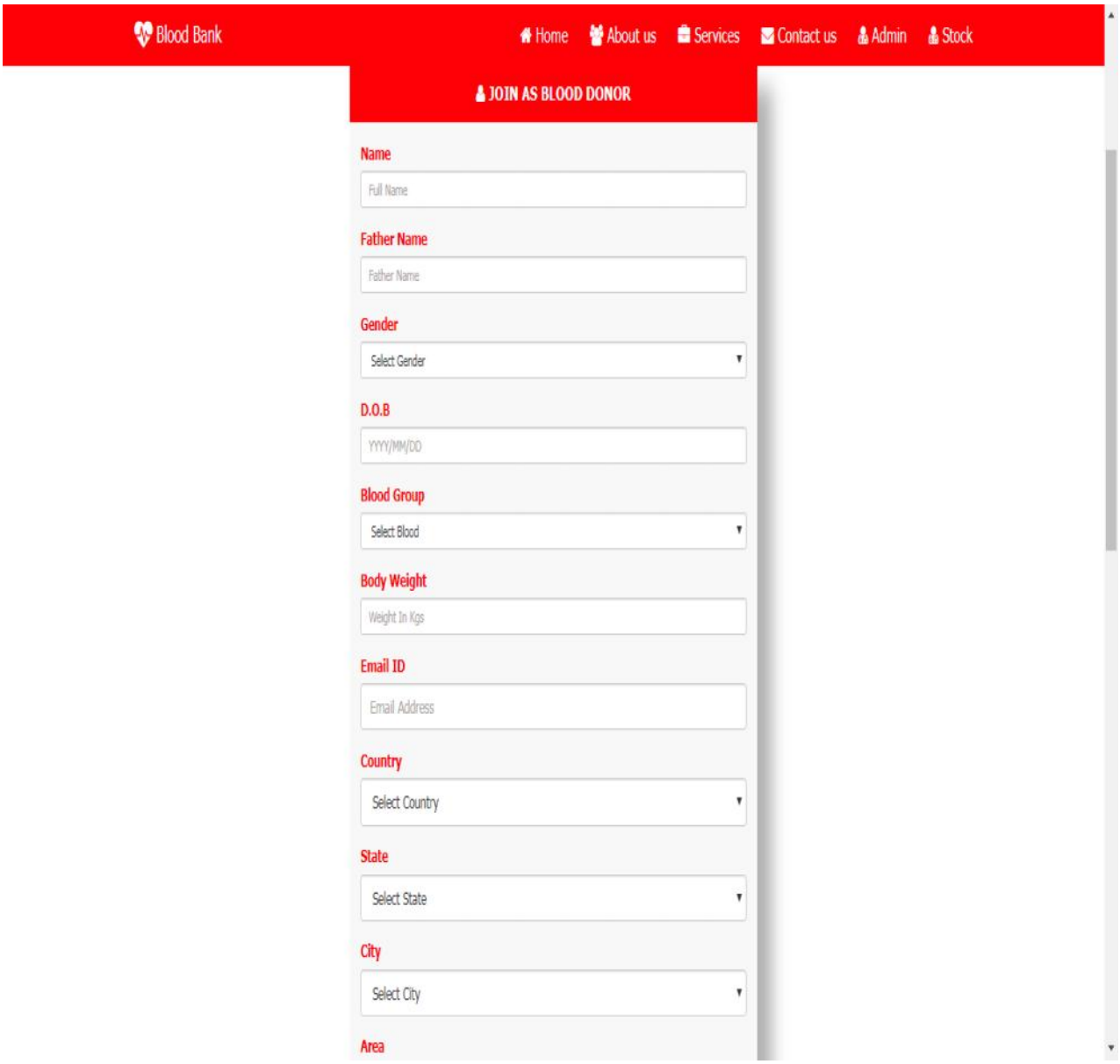
- www.php.net → PHP Manual
- http://www.google.com
- http://www.w3schools.com
- http://www.indianbloodgroup.com

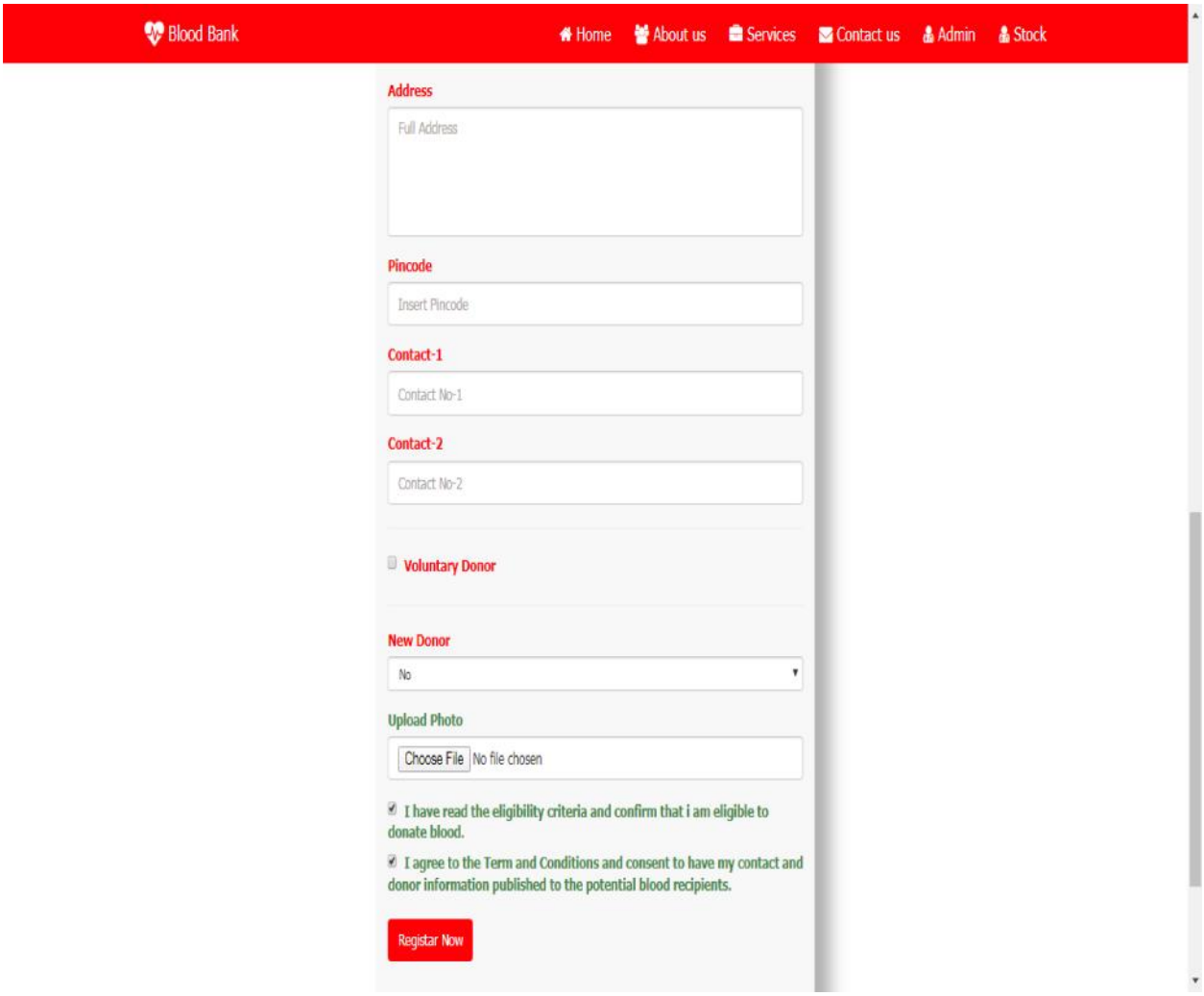
1. "PHP PROFESSIONAL PROJECTS"- Ashish Wilfred, meeta gupta and karti Bhatnagar with NIIT,PHI publication(Prentice – Hall India)
2. "PHP , MY SQL and Apache"-Julie c. meloni pearson education.
3. World Wide Web Designing , C . Xavier , Tata McGraw-Hill,2000.
4. Raghu RamaKrishnan ," Database Management System", Tata Graw-Hill,2003.
5. Robert D. Schneider, Jetty R. Garbus , "Optimizing SQL Server", Second Edition, Pearson Education Asia, ISBN: 981-4035-20-3
6. Matt Doyle," Beginning PHP 5.3", Wiley Publishing , Inc.

CHAPTER 8

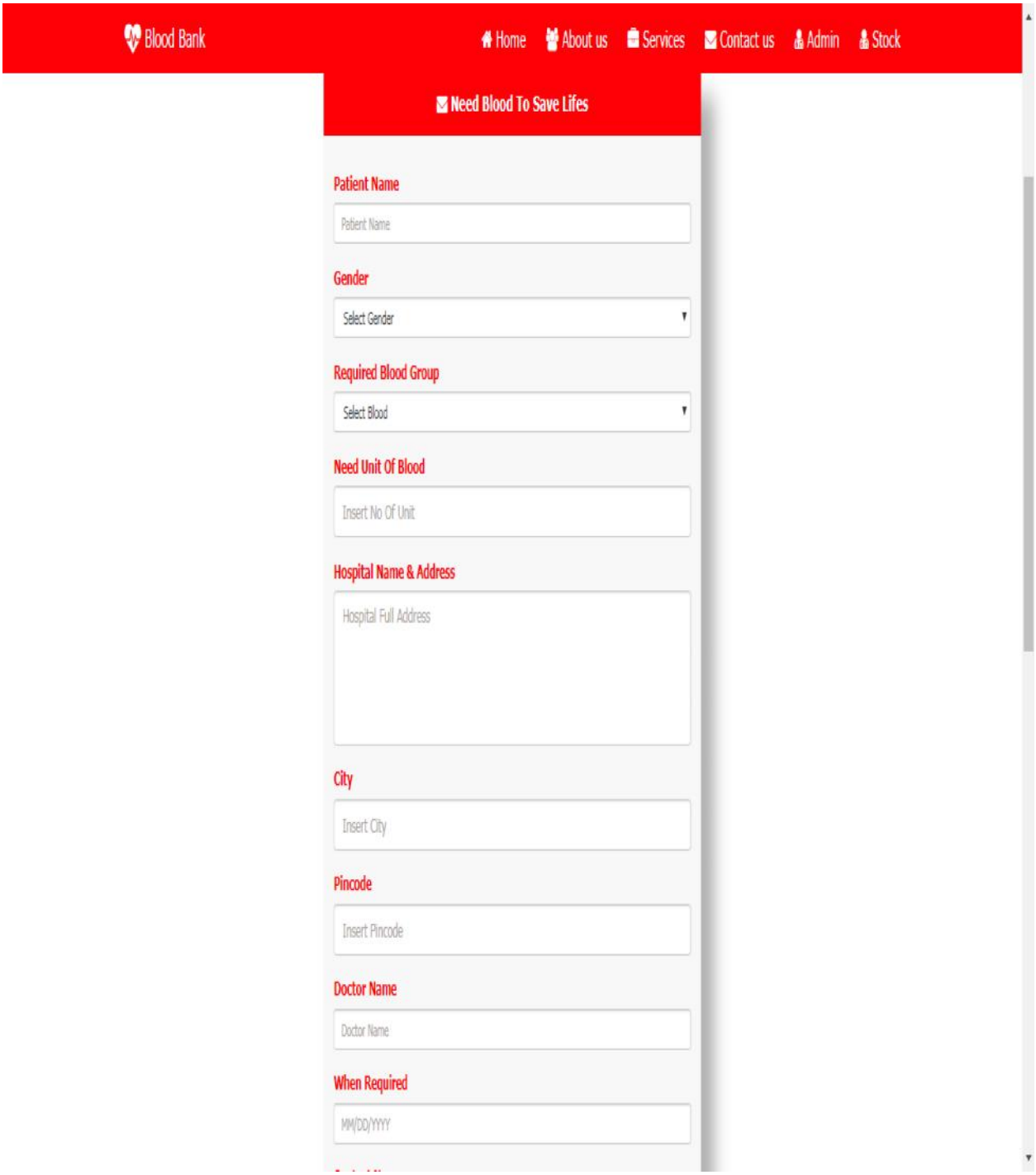
8.1 APPENDIX SCREEN SHOTS:

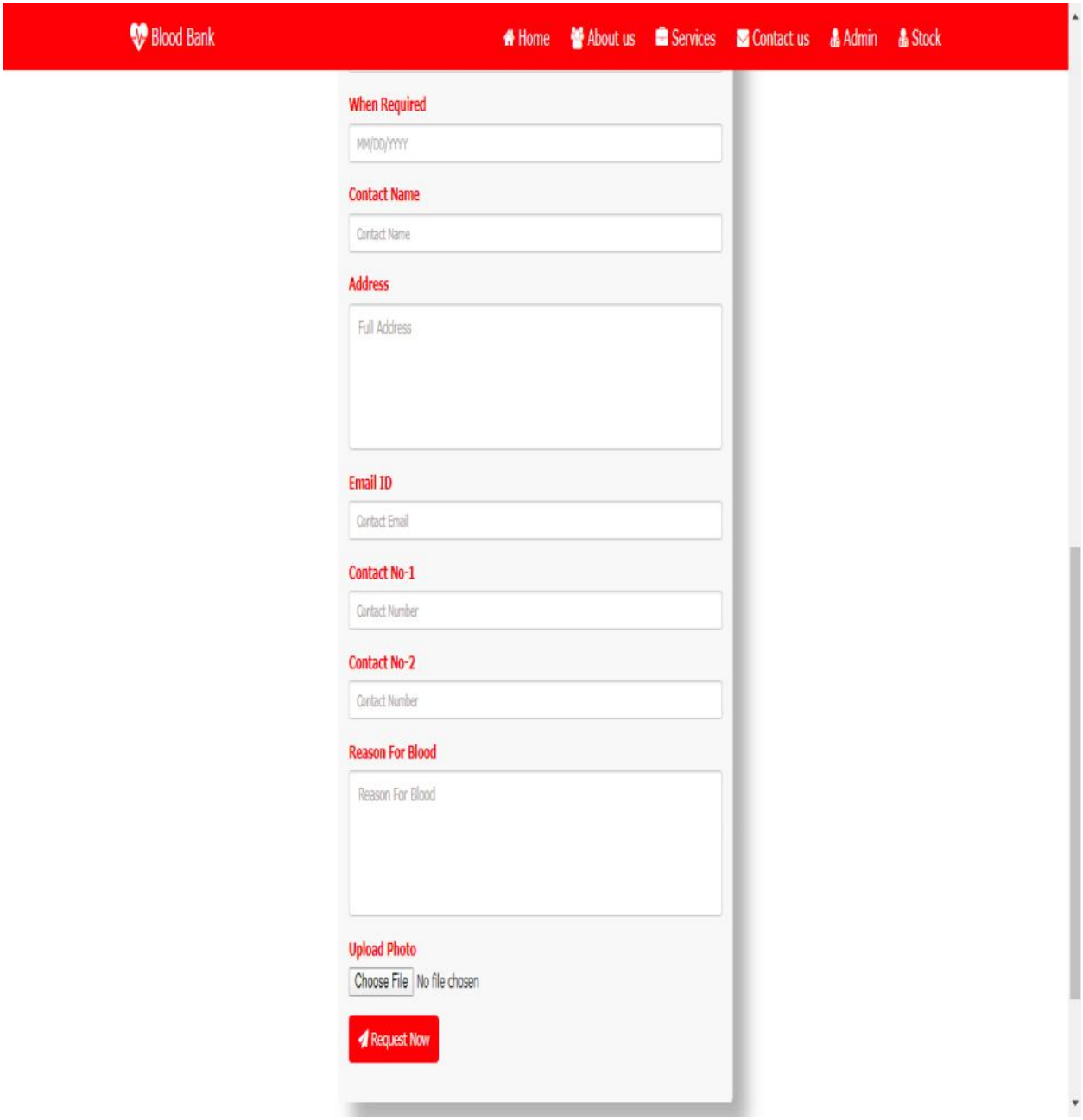
Donor registration:



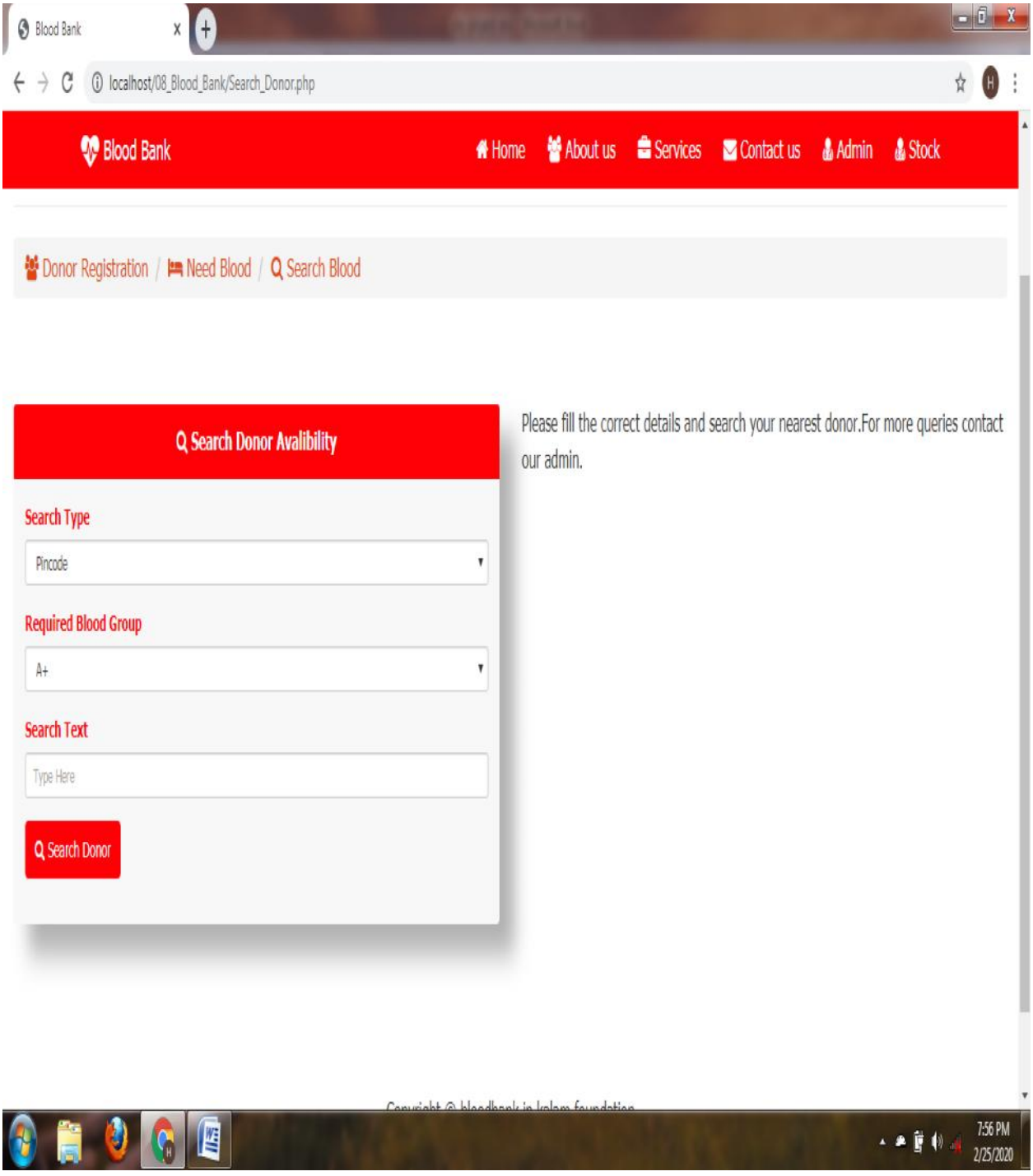

Blood Bank
Home
About us
Services
Contact us
Admin
Stock
Address
Full Address
Pincode
Insert Pincode
Contact-1
Contact No-1
Contact-2
Contact No-2
Voluntary Donor
New Donor
No
Upload Photo
Choose File
No file chosen
I have read the eligibility criteria and confirm that i am eligible to donate blood.
I agree to the Term and Conditions and consent to have my contact and donor information published to the potential blood recipients.
Registar Now

Seeker registration:

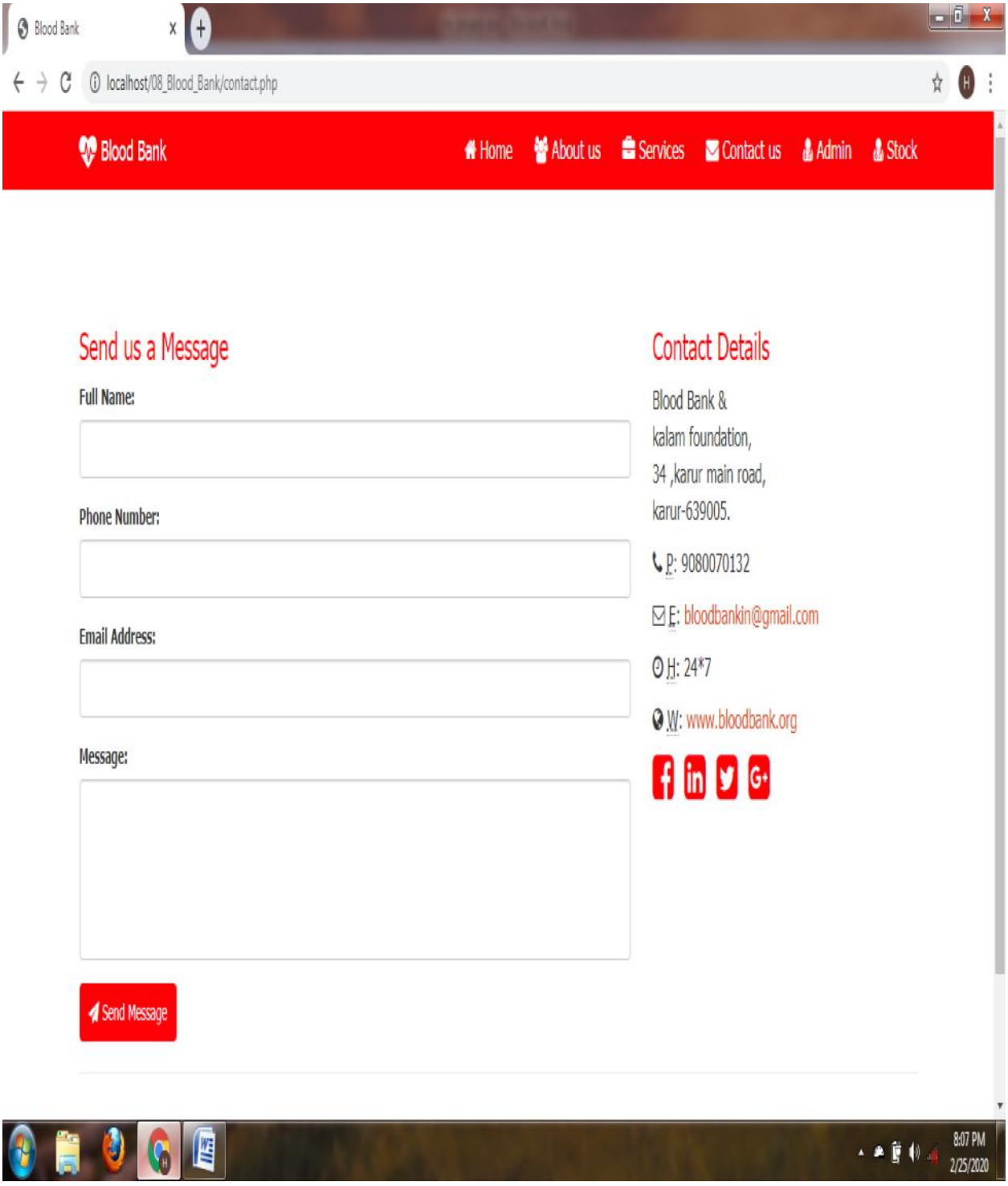



Blood Bank
Home
About us
Services
Contact us
Admin
Stock
When Required
MM/DD/YYYY
Contact Name
Contact Name
Address
Full Address
Email ID
Contact Email
Contact No-1
Contact Number
Contact No-2
Contact Number
Reason For Blood
Reason For Blood
Upload Photo
Choose File
No file chosen
Request Now

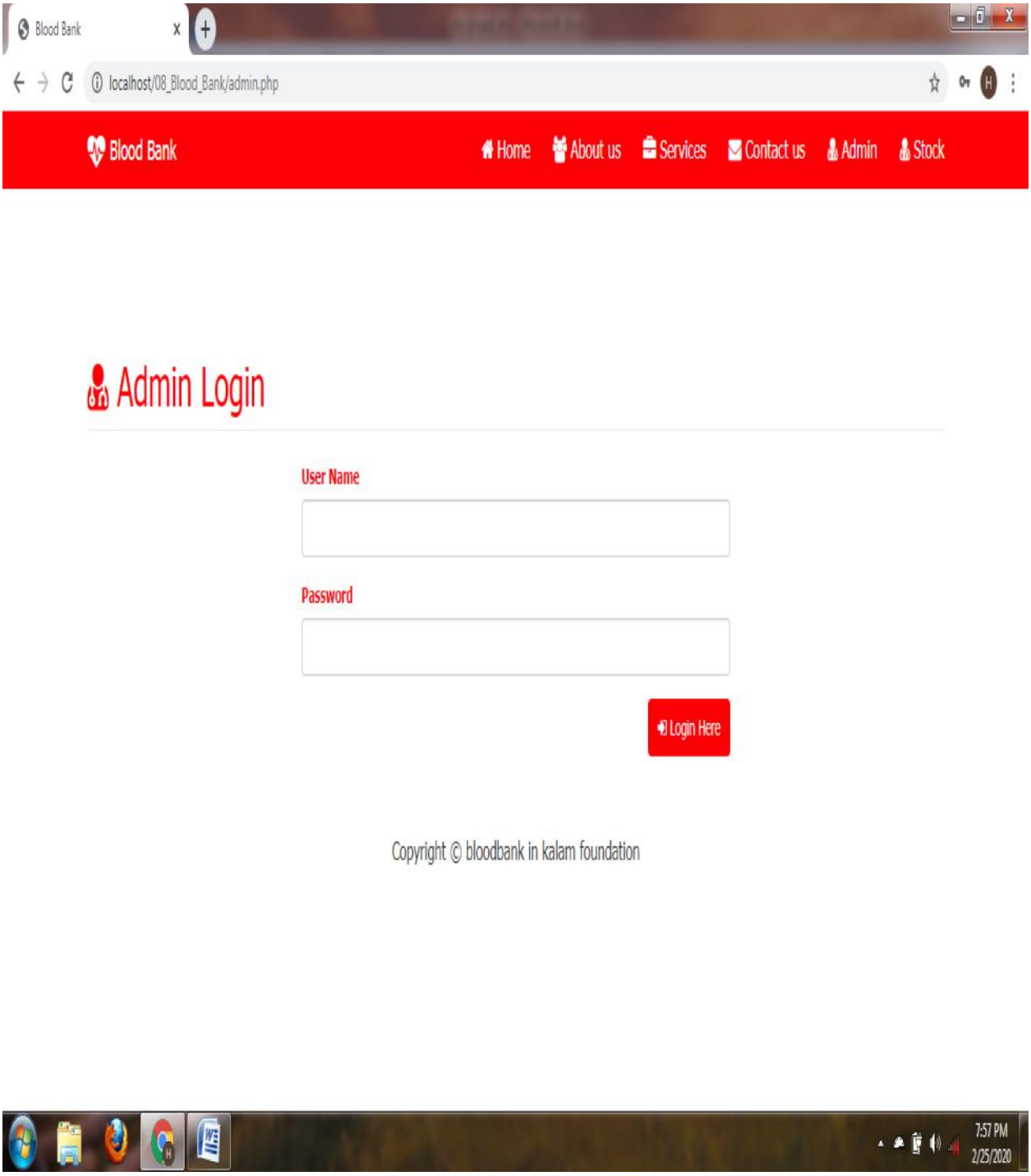
Search donor:



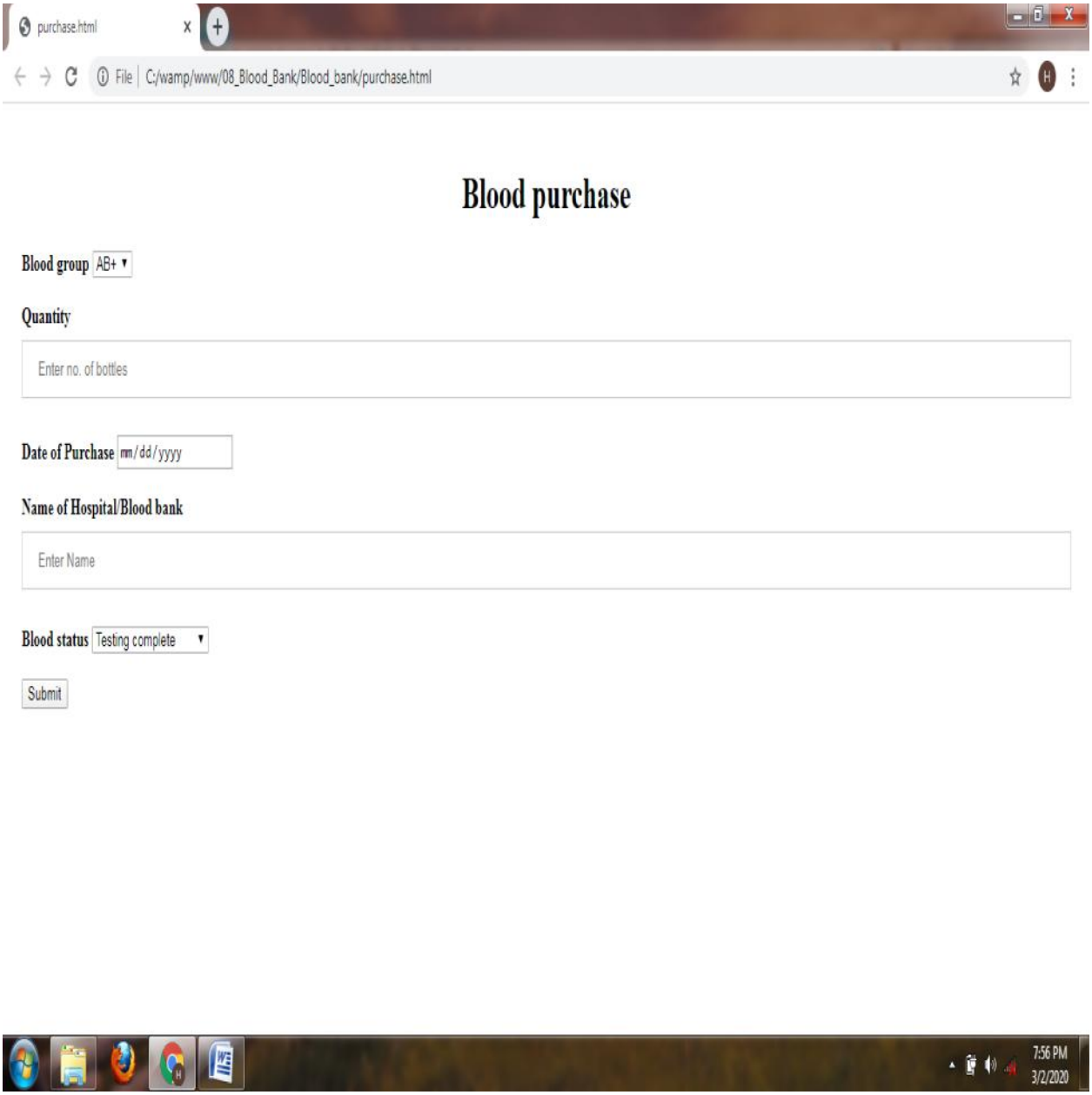
Message:


Blood Bank
localhost/08_Blood_Bank/contact.php
Blood Bank
Home
About us
Services
Contact us
Admin
Stock
Send us a Message
Full Name:
Phone Number:
Email Address:
Message:
Send Message
Contact Details
Blood Bank &
kalam foundation,
34 ,karur main road,
karur-639005.
P: 9080070132
E: bloodbankin@gmail.com
H: 24*7
W: www.bloodbank.org
8:07 PM
2/25/2020

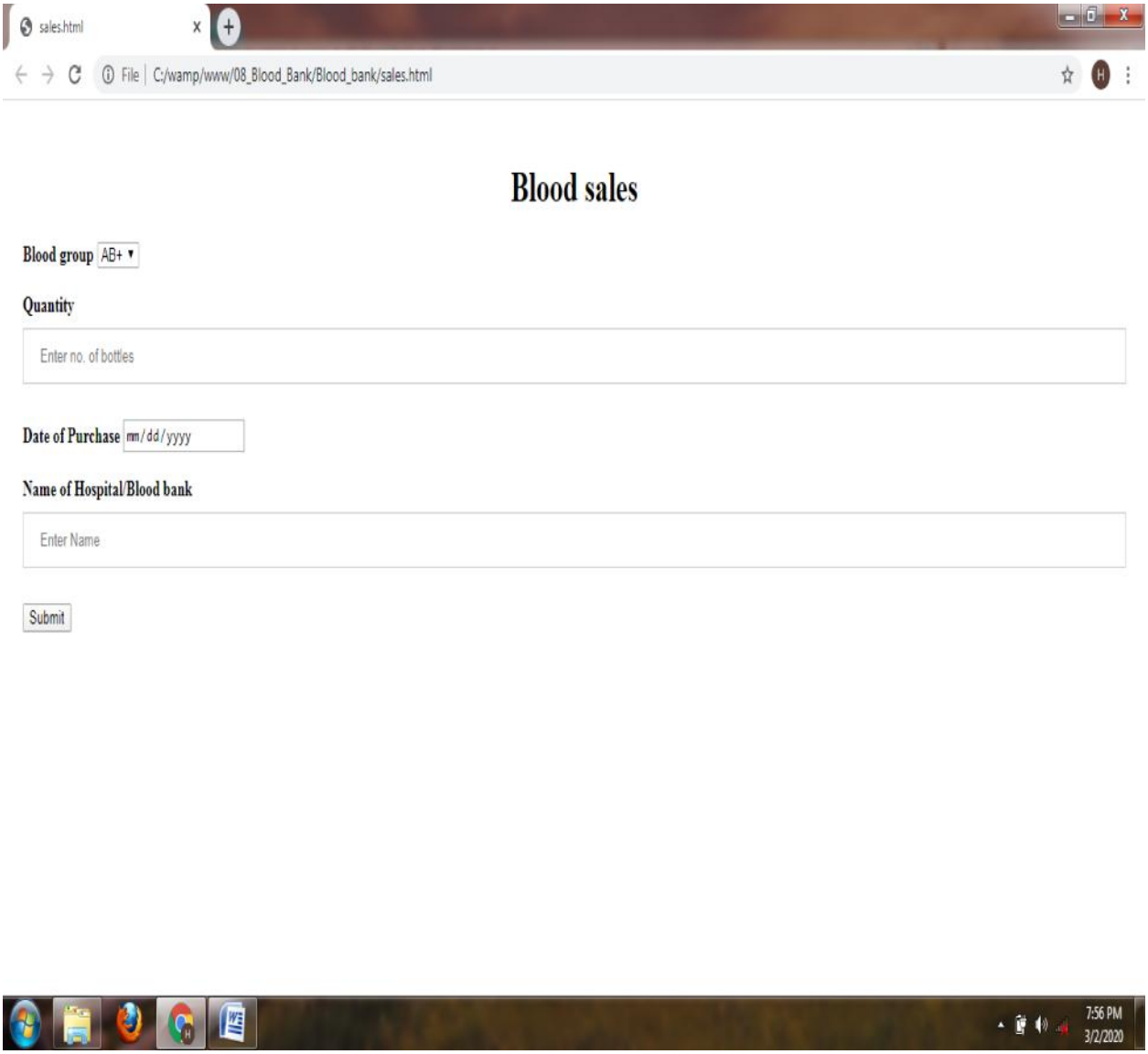
Admin login:


Blood Bank
localhost/08_Blood_Bank/admin.php
Blood Bank
Home
About us
Services
Contact us
Admin
Stock
Admin Login
User Name
Password
Login Here
Copyright © bloodbank in kalam foundation
7:57 PM
2/25/2020

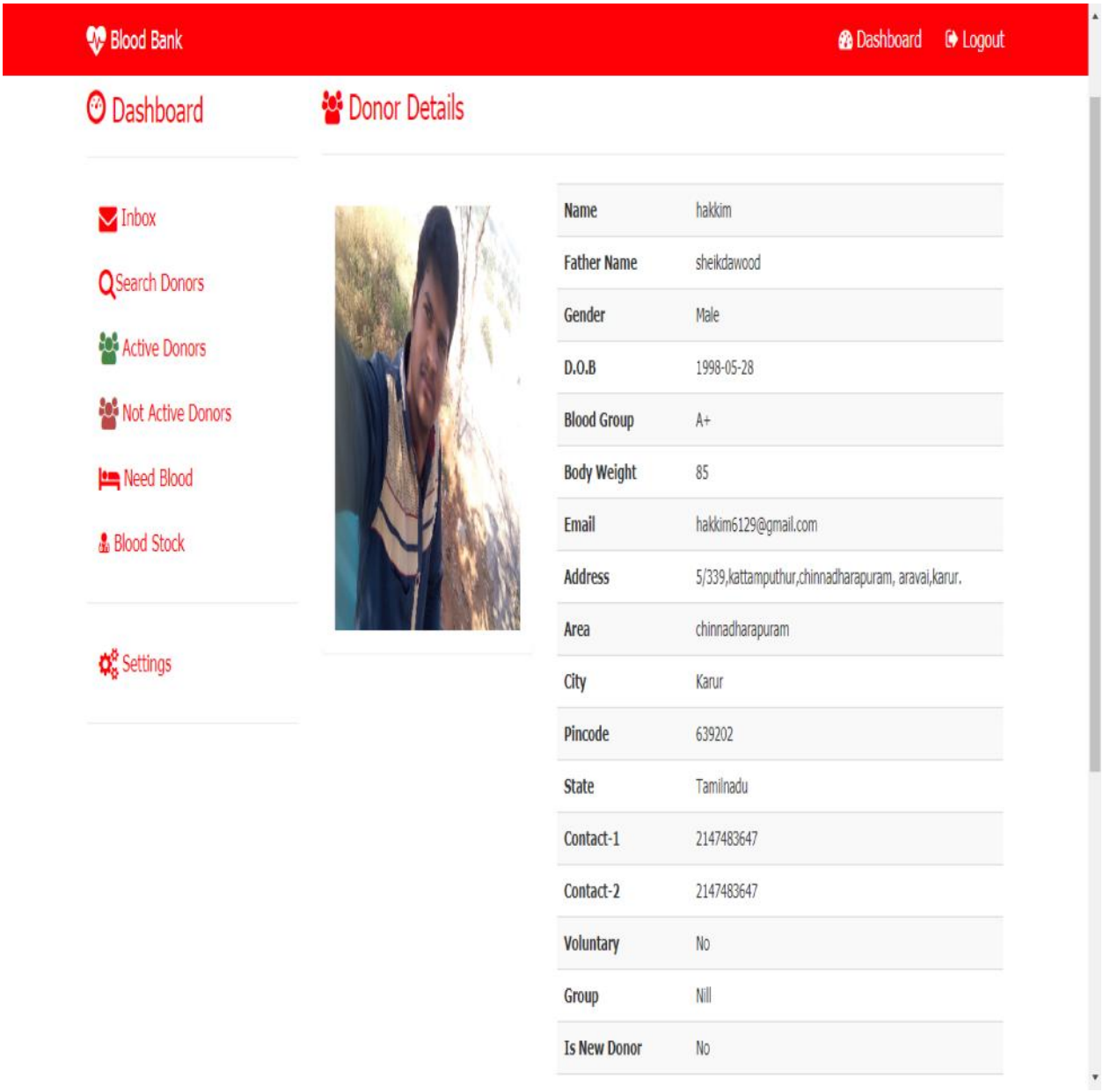
Purchase blood:

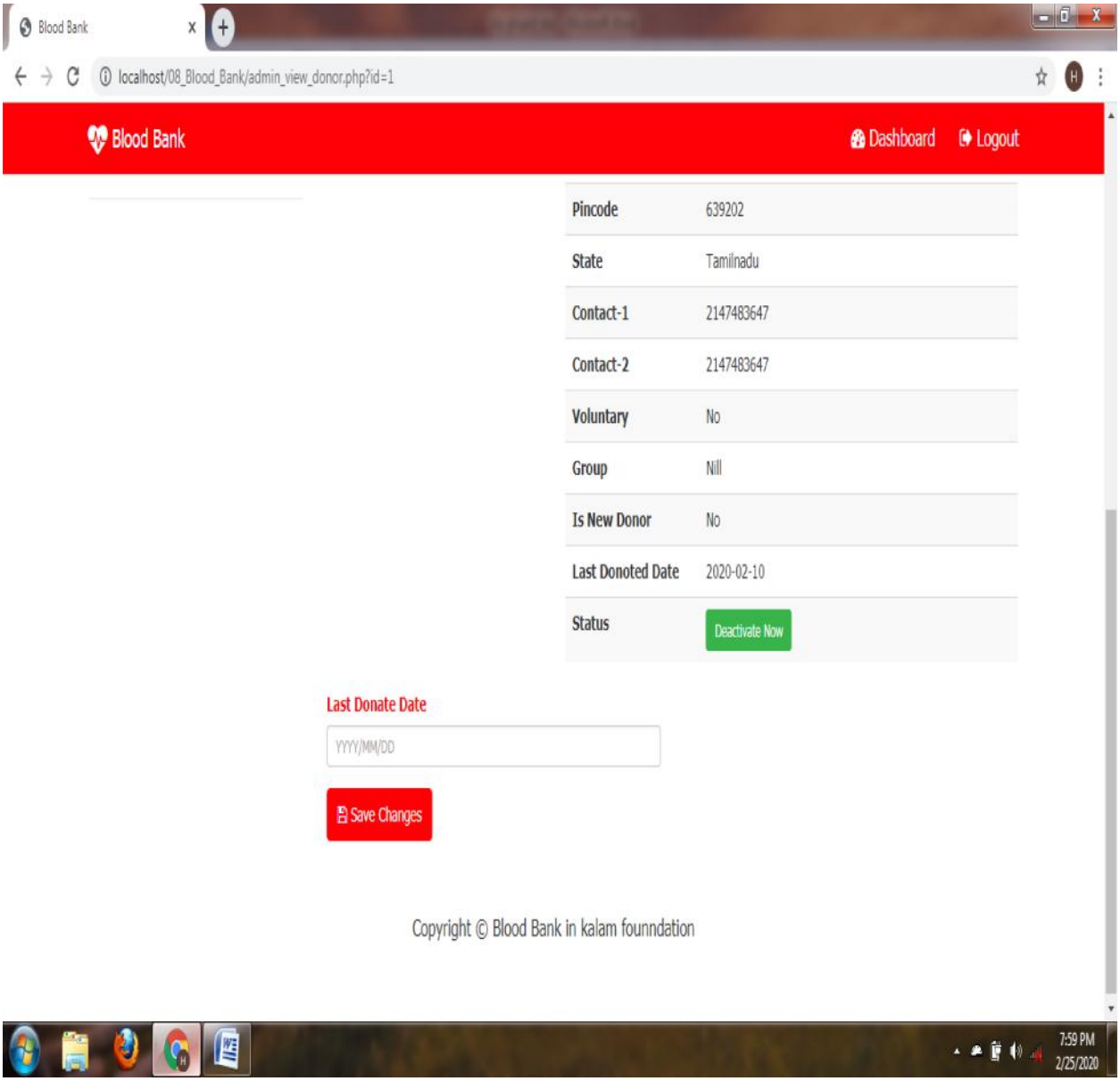


Sale blood:

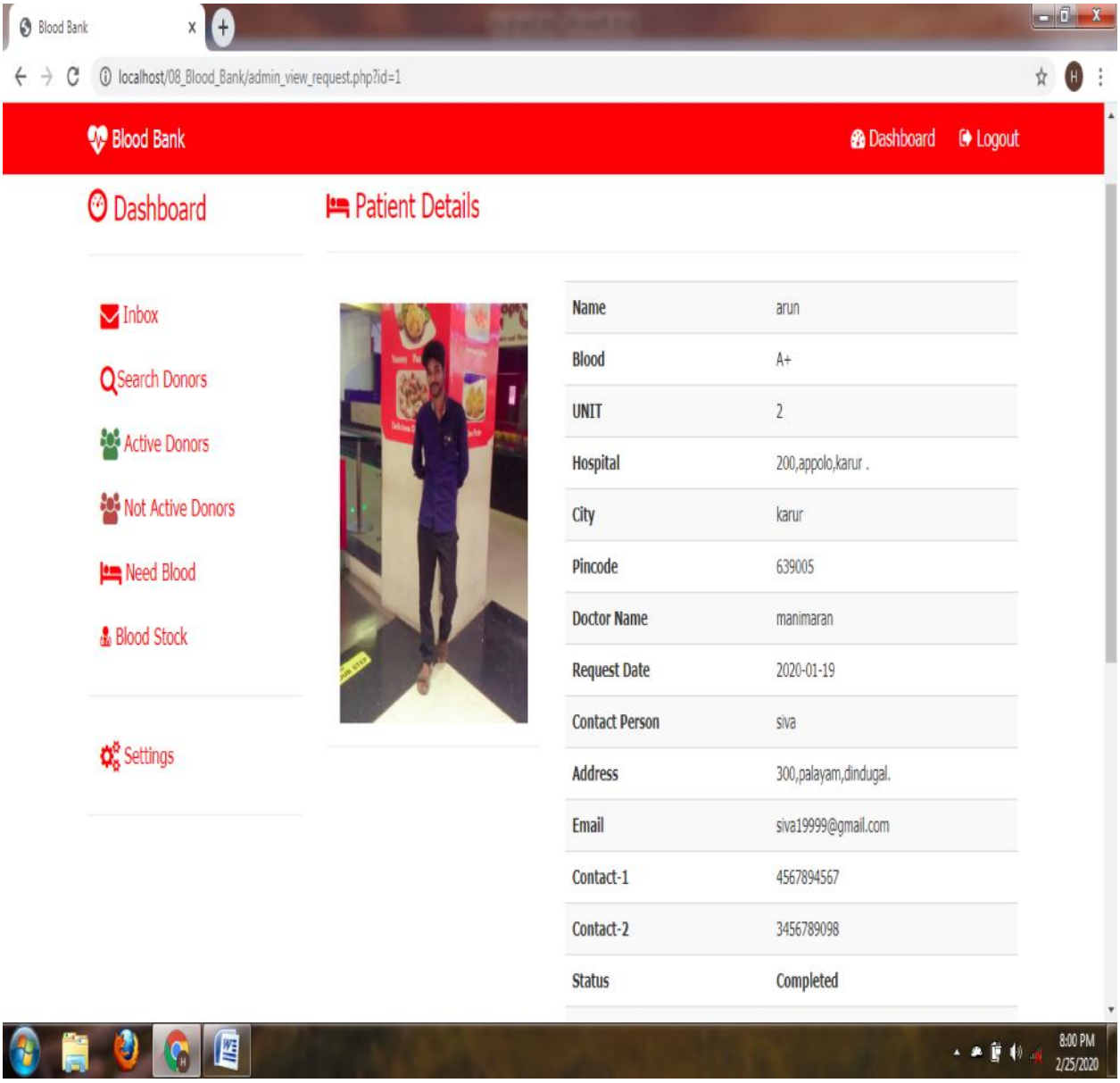

sales.html
File | C:/wamp/www/08_Blood_Bank/Blood_bank/sales.html
Blood sales
Blood group AB+
Quantity
Enter no. of bottles
Date of Purchase mm/dd/yyyy
Name of Hospital/Blood bank
Enter Name
Submit
7:56 PM
3/2/2020

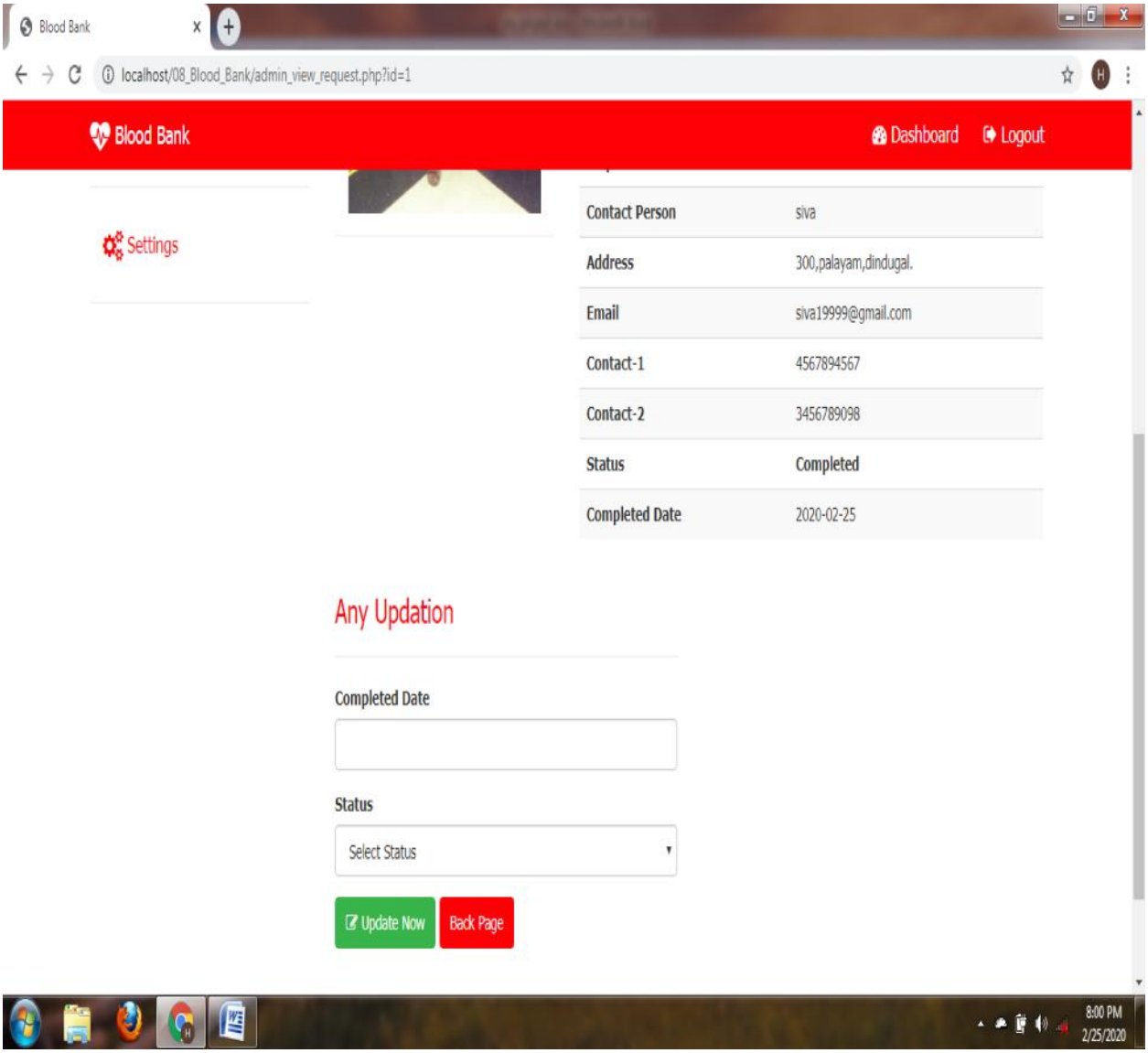
Donor details:



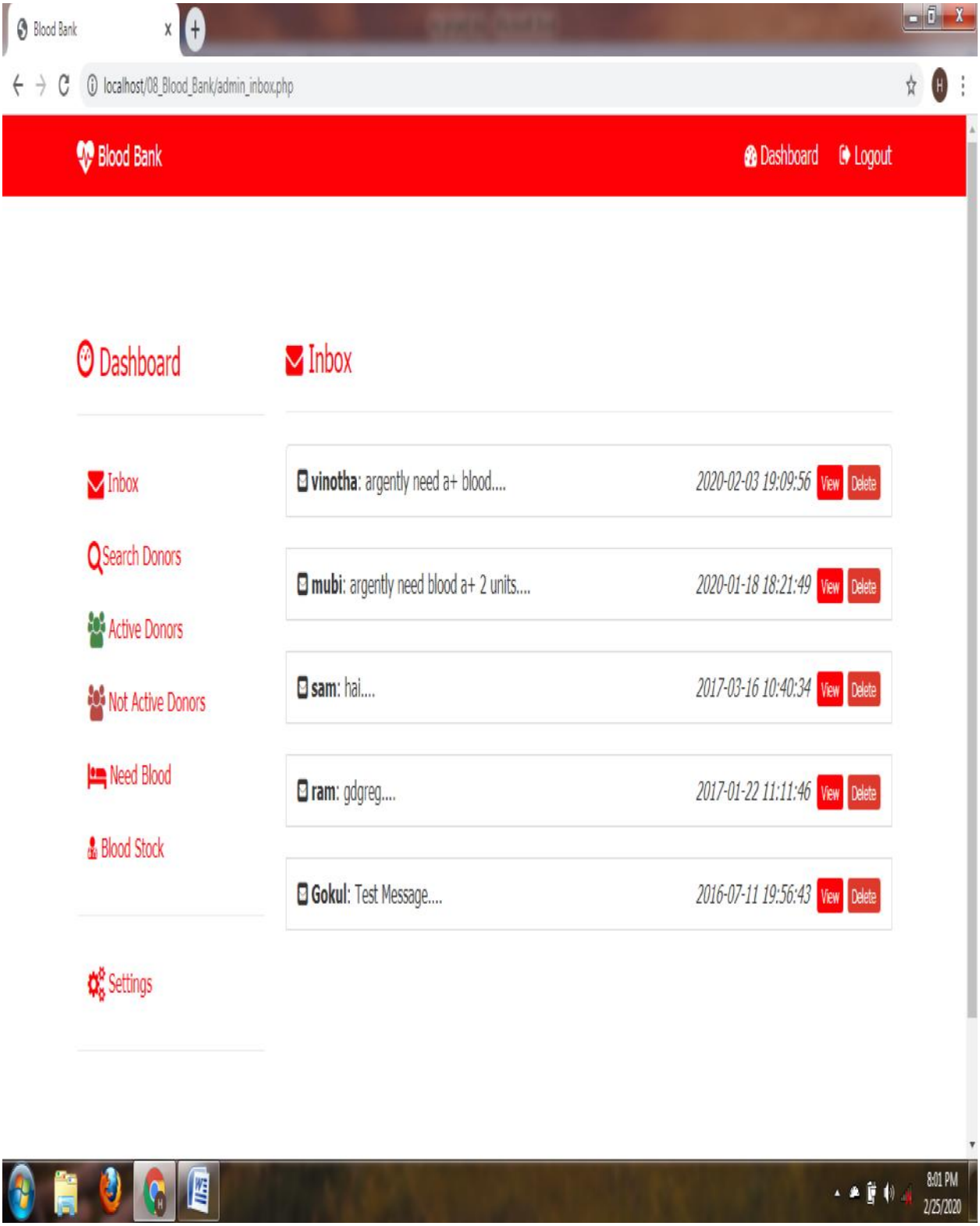

Blood Bank
localhost/08_Blood_Bank/admin_view_donor.php?id=1
Blood Bank
Dashboard
Logout
Pincode 639202
State Tamilnadu
Contact-1 2147483647
Contact-2 2147483647
Voluntary No
Group Nill
Is New Donor No
Last Donoted Date 2020-02-10
Status Deactivate Now
Last Donate Date
YYYY/MM/DD
Save Changes
Copyright © Blood Bank in kalam founndation
7:59 PM
2/25/2020

Seeker details:

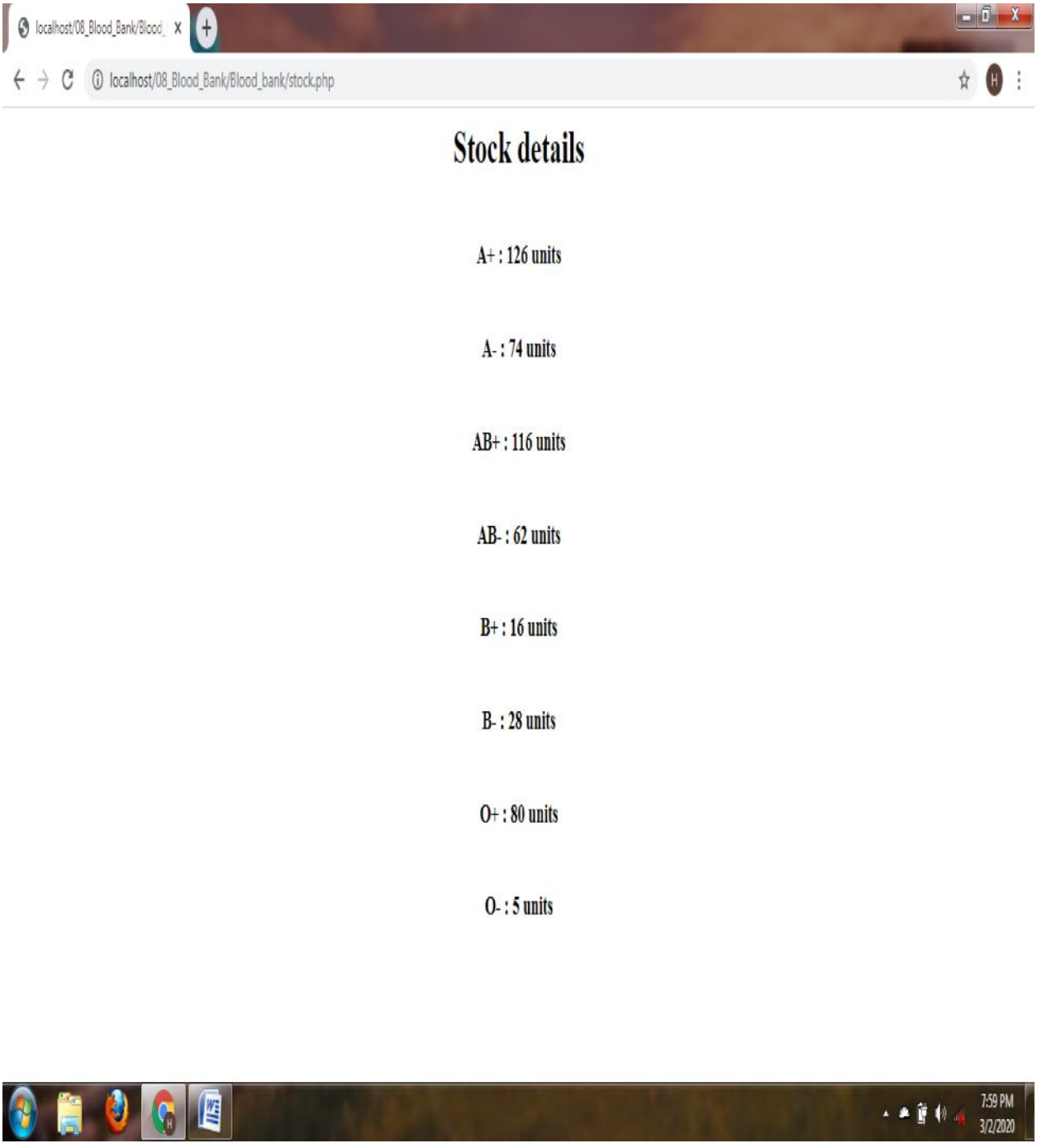



Blood Bank
localhost/08_Blood_Bank/admin_view_request.php?id=1
Blood Bank
Dashboard
Logout
Settings
Contact Person
siva
Address
300,palayam,dindugal.
Email
siva19999@gmail.com
Contact-1
4567894567
Contact-2
3456789098
Status
Completed
Completed Date
2020-02-25
Any Updation
Completed Date
Status
Select Status
Update Now
Back Page
8:00 PM
2/25/2020

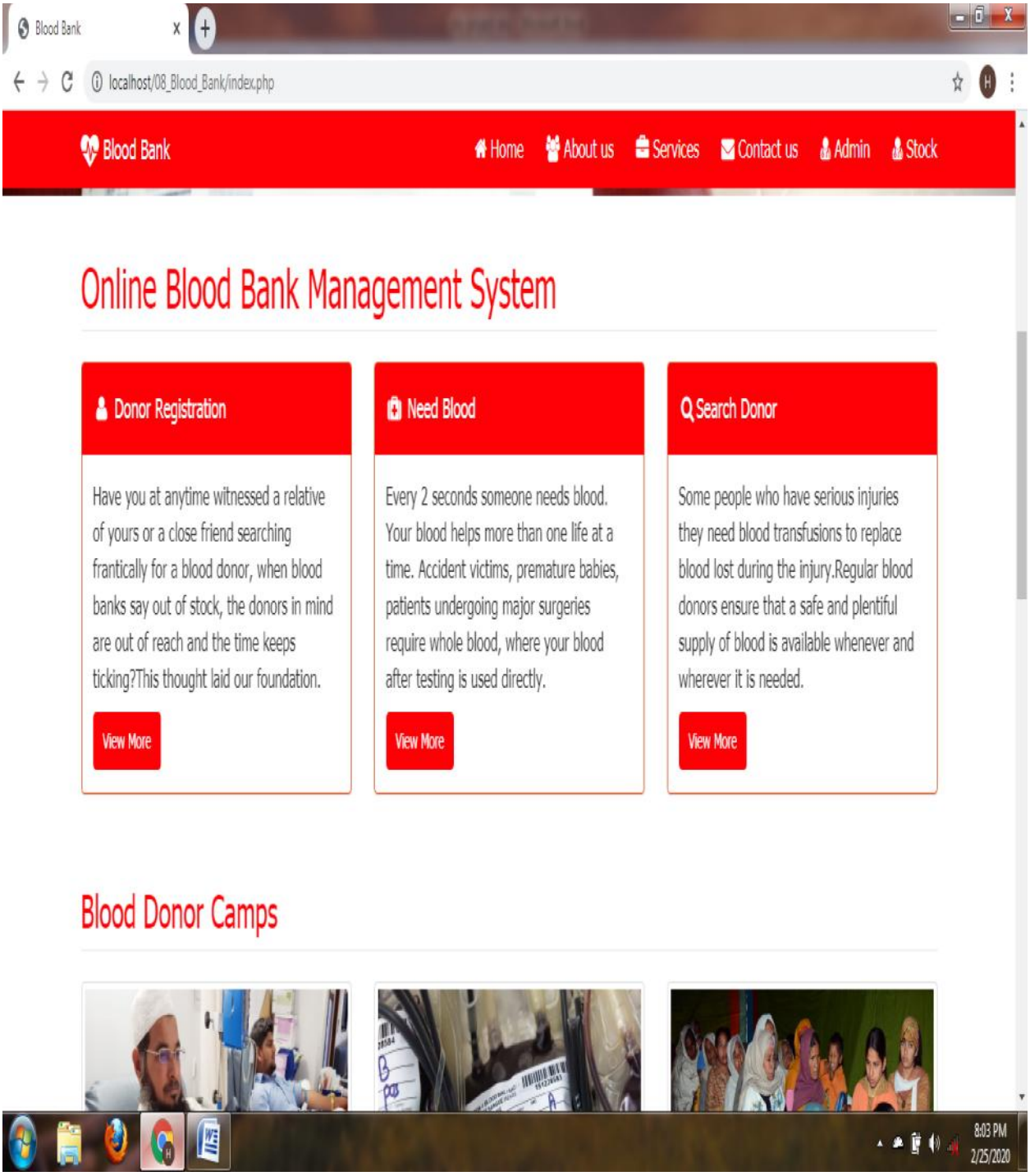
Message:


Blood Bank
localhost/08_Blood_Bank/admin_inbox.php
Blood Bank
Dashboard
Logout
Dashboard
Inbox
Inbox
Search Donors
Active Donors
Not Active Donors
Need Blood
Blood Stock
Settings
vinotha: argently need a+ blood.... 2020-02-03 19:09:56 View Delete
mubi: argently need blood a+ 2 units.... 2020-01-18 18:21:49 View Delete
sam: hai.... 2017-03-16 10:40:34 View Delete
ram: gdgreg.... 2017-01-22 11:11:46 View Delete
Gokul: Test Message.... 2016-07-11 19:56:43 View Delete
8:01 PM
2/25/2020

Stock details:



User home page:



8.2 APPENDIX

Admin.php:-

```
<?php
session_start();
include("config.php");
?>
<html lang="en">
<head>
<?php include("head.php");?>
</head>
<body>
<?php include("top_nav.php"); ?>
<!-- Navigation -->
<!-- Page Content -->
<div class="container" style="margin-top:70px;">
<!-- Page Heading/Breadcrumbs -->
<div class="row">
<div class="col-lg-12">
<h1 class="page-header text-primary"><i class='fa fa-user-md'></i> Admin Login

</h1>
</div>
<div class="row">
<div class="col-md-3"></div>
<div class="col-md-6">
<?php
if(isset($_POST["submit"]))
{                       if($_POST["user"]=="admin"&&$_POST["pass"]=="admin")
{                                                       $_SESSION['usertype']
='admin';
$_SESSION['username']='admin';
header("location:admin_inbox.php");
}
else
{
echo "<div class='alert alert-danger'><b>Error</b> User Name and Password
Incorrect.</div>";
}
}
?>
```

```
<form role="form" action="admin.php" method="post">
<div class="form-group">
<label for="user_name" class="text-primary">User Name</label>
<input class="form-control" name="user"  id="user" type="text" required>
</div>
<div class="form-group">
<label for="pass" class="text-primary">Password</label>
<input class="form-control" id="pass" name="pass" type="password" value=""
required>
</div>
<button class="btn btn-primary pull-right" name="submit" type="submit"><i
class="fa fa-sign-in"></i> Login Here</button>
</form>
</div>
<div class="col-md-3"></div>
</div>
</div>
<!-- /.row -->
<!-- Footer -->
<?php include"footer.php";?>
</div>
</body>
</html>
```

Donor registration.php:-

```
<?php
include("config.php");
//include("functions.php");
error_reporting(0);?>
<!DOCTYPE html>
<html lang="en">
<head>
<?php include("head.php");?>
</head>
<body>

<?php
include("top_nav.php");
?>
<div class="container" style='margin-top:70px;'>
<div class="row">
```

```
<div class="col-md-12">
<h3 class=" text-primary">
<i class='fa fa-users'></i> New Donor Registration
</h3><hr>
<?php  include("blood_bread.php"); ?>
</div>
</div>
<div class="row centered-form ">
<div class="col-xs-12 col-sm-8 col-md-6 col-sm-offset-2 col-md-offset-3">
<?php
if(isset($_POST["submit"]))
{
$target_dir = "donor_image/";
$img="donor_image/noimage.jpg";
$target_file = $target_dir.rand(100,999).
basename($_FILES["fileToUpload"]["name"]);
$uploadOk = 1;
$imageFileType = pathinfo($target_file,PATHINFO_EXTENSION);
// Check if image file is a actual image or fake image
$check = getimagesize($_FILES["fileToUpload"]["tmp_name"]);
if($check !== false) {
echo "";
$uploadOk = 1;
} else {
//  echo "File is not an image.";
$uploadOk = 0;
}

// Check if file already exists
if (file_exists($target_file)) {
// echo "Sorry, file already exists.";
$uploadOk = 0;
}
// Check file size
if ($_FILES["fileToUpload"]["size"] > 5000000000) {
// echo "Sorry, your file is too large.";
$uploadOk = 0;
}
// Allow certain file formats
```

```
if($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg"
&& $imageFileType != "gif" ) {
// echo "Sorry, only JPG, JPEG, PNG & GIF files are allowed.";
$uploadOk = 0;
}
// Check if $uploadOk is set to 0 by an error
if ($uploadOk == 0) {
// echo "Sorry, your file was not uploaded.";
// if everything is ok, try to upload file
} else {
if (move_uploaded_file($_FILES["fileToUpload"]["tmp_name"], $target_file)) {
$img=$target_file;
} else {
//   echo "Sorry, there was an error uploading your file.";
}
}
$country="";
$state="";

$qry="SELECT COUNTRY_NAME FROM country WHERE
COUNTRY_ID={$_POST["COUNTRY"]}";
$res=$con->query($qry);
if($res->num_rows>0)
{
if($row=$res->fetch_assoc())
{
$country=$row["COUNTRY_NAME"];
}
}

$qry="SELECT STATE_NAME FROM state WHERE
STATE_ID={$_POST["STATE"]}";
$res=$con->query($qry);
if($res->num_rows>0)
{
if($row=$res->fetch_assoc())
{
$state=$row["STATE_NAME"];
}
```

```
}
$cityname=$_POST["CITY"];
$sql="
INSERT INTO blood_donor
(NAME, FATHER_NAME, GENDER, DOB, BLOOD, BODY_WEIGHT, EMAIL,
ADDRESS, AREA, CITY, PINCODE, STATE, CONTACT_1, CONTACT_2,
VOLUNTARY, VOLUNTARY_GROUP,NEW_DONOR, LAST_D_DATE,
DONOR_PIC,COUNTRY)
VALUES
('{$_POST["NAME"]}', '{$_POST["FATHER_NAME"]}', '{$_POST["GENDER"]}',
'{$_POST["DOB"]}', '{$_POST["BLOOD"]}', '{$_POST["BODY_WEIGHT"]}',
'{$_POST["EMAIL"]}', '{$_POST["ADDRESS"]}', '{$_POST["AREA"]}',
'$cityname', '{$_POST["PINCODE"]}', '{$state}', '{$_POST["CONTACT_1"]}',
'{$_POST["CONTACT_2"]}', '{$_POST["VOLUNTARY"]}',
'{$_POST["VOLUNTARY_GROUP"]}',
'{$_POST["NEW_DONOR"]}','{$_POST["LAST_D_DATE"]}',
'{$img}','{$country}');";
if($con->query($sql))
{
echo '
<div class="alert alert-success">
<a href="#" class="close" data-dismiss="alert" aria-label="close">&times;</a>
<strong>Success!</strong> Thank you for adding you as donor.
</div>
';
}
}
?>
<div class="panel panel-primary">
<div class="panel-heading">
<h3 class="panel-title text-center" style="padding:5px;font-size:16px;font-
weight:bold"><span class="fa fa-user "> </span> JOIN AS BLOOD DONOR</h3>
</div>
<div class="panel-body">
<form method="post" action="Donor_reg.php" autocomplete="off" role="form"
enctype="multipart/form-data">
<div class="form-group">
<label class="control-label text-primary" for="NAME" >Name</label>
<input type="text" placeholder="Full Name" id="NAME" name="NAME"  required
class="form-control input-sm">
```

```
</div>
<div class="form-group">
<label class="control-label text-primary" for="FATHER_NAME">Father
Name</label>
<input type="text" placeholder="Father Name" id="FATHER_NAME"
name="FATHER_NAME" required class="form-control input-sm">
</div>

<div class="form-group">
<label class="control-label text-primary"  for="GENDER">Gender</label>
<select id="gen" name="GENDER" required class="form-control input-sm">
<option value="">Select Gender</option>
<option value="Male">Male</option>
<option value="Female">Female</option>
<option value="others">others</option>
</select>
</div>
<div class="form-group">
<label class="control-label text-primary" for="DOB">D.O.B</label>
<input type="text"  placeholder="YYYY/MM/DD" required id="DOB" name="DOB"
class="form-control input-sm DATES">
</div>
<div class="form-group">
<label class="control-label text-primary" for="BLOOD" >Blood Group</label>
<select id="blood" name="BLOOD" required class="form-control input-sm">
<option value="">Select Blood</option>
<option value="A+">A+</option>
<option value="A-">A-</option>
<option value="B+">B+</option>
<option value="B-">B-</option>
<option value="O+">O+</option>
<option value="O-">O-</option>
<option value="AB+">AB+</option>
<option value="AB-">AB-</option>
</select>
</div>
<div class="form-group">
<label class="control-label text-primary" for="BODY_WEIGHT" >Body
Weight</label>
```

```
<input type="text" required placeholder="Weight In Kgs"  name="BODY_WEIGHT"
id="BODY_WEIGHT" class="form-control input-sm">
</div>
<div class="form-group">
<label class="control-label text-primary" for="EMAIL" >Email ID</label>
<input type="email"  required name="EMAIL" id="EMAIL" class="form-control"
placeholder="Email Address">
</div>

<div class="form-group">
<label class="control-label text-primary" for="COUNTRY">Country</label>
<select name="COUNTRY" id="COUNTRY" required class="form-control">
<option value="">Select Country</option>
<?php
$sql="SELECT COUNTRY_ID,COUNTRY_NAME FROM country ORDER BY
COUNTRY_NAME ASC";
$result=$con->query($sql);
if($result->num_rows>0)
{
while($row=$result->fetch_assoc())
{
echo "<option value='{$row['COUNTRY_ID']}'>{$row['COUNTRY_NAME']}
                        </option>";
}
}
?>
</select>
</div>
<div class="form-group">
<label class="control-label text-primary" for="STATE">State</label>
<select name="STATE" id="STATE" required class="form-control">
<option value="">Select State</option>
<?php
$sql="SELECT STATE_ID,STATE_NAME FROM state ORDER BY
STATE_NAME ASC";
$result=$con->query($sql);
if($result->num_rows>0)
{
while($row=$result->fetch_assoc())
{
```

```
echo "<option value='{$row['STATE_ID']}'>{$row['STATE_NAME']}
                    </option>";
}
}

?>
</select>
</div>
<div class="form-group">
<label class="control-label text-primary" for="CITY" >City</label>
<select name="CITY" id="CITY" required class="form-control">
<option value="">Select City</option>
<?php
$sql="SELECT CITY_NAME,CITY_ID FROM city ORDER BY CITY_NAME";
$result=$con->query($sql);
if($result->num_rows>0)
{
while($row=$result->fetch_assoc())
{
echo "<option value='{$row['CITY_ID']}'>{$row['CITY_NAME']}  </option>";
}
}

?>
</select>

</div>

<div class="form-group">
<label class="control-label text-primary" for="AREA" >Area</label>
<input type="text" required name="AREA" id="AREA" class="form-control"
placeholder="Insert Area">
</div>

<div class="form-group">
<label class="control-label text-primary" for="ADDRESS">Address</label>
<textarea required name="ADDRESS" id="ADDRESS" rows="5"
style="resize:none;"class="form-control" placeholder="Full Address"></textarea>
</div>
```

```
<div class="form-group">
<label class="control-label text-primary" for="PINCODE">Pincode</label>
<input type="text" required name="PINCODE" id="PINCODE" class="form-control"
placeholder="Insert Pincode">
</div>
<div class="form-group">
<label class="control-label text-primary" for="CONTACT_1" >Contact-1</label>
<input type="text" required name="CONTACT_1" id="CONTACT_1" class="form-
control" placeholder="Contact No-1">
</div>
<div class="form-group">
<label class="control-label text-primary" for="CONTACT_2" >Contact-2</label>
<input type="text" required name="CONTACT_2" id="CONTACT_2" class="form-
control" placeholder="Contact No-2">
</div>
<hr>
<div class="form-group">
<label class="control-label text-primary"><input type="checkbox" id="c1" > 
Voluntary Donor</label>
</div>
<div id="volu">
<div class="form-group">

<select name="VOLUNTARY"  id="VOLUNTARY"   class="form-control input-
sm">
<option value="">Select</option>
<option value="Yes">Yes</option>
<option selected value="No">No</option>
</select>
</div>
<div class="form-group">
<input type="text"  name="VOLUNTARY_GROUP" id="VOLUNTARY_GROUP"
class="form-control" placeholder="Voluntary Group Name" value="Nill">
</div>
<div class="form-group">
<label class="control-label text-primary"  for="LAST_D_DATE">Last Blood
Donoted Date</label>
<input type="text"  name="LAST_D_DATE" value="0000/00/00"
id="LAST_D_DATE" placeholder="YYYY/MM/DD" class="form-control input-sm
DATES">
```

```
</div>
</div>
<hr>
<div class="form-group" id="new">
<label class="control-label text-primary"  for="NEW_DONOR">New Donor</label>
<select name="NEW_DONOR"  id="NEW_DONOR"  class="form-control input-sm">
<option value="">Select</option>
<option value="Yes" >Yes</option>
<option value="No" selected>No</option>
</select>
</div>
<div class="form-group">
<label class="control-label text-success" for="fileToUpload" >Upload Photo</label>
<input type="file" class="form-control"  name="fileToUpload">
</div>
<div class="form-group">
<label class="control-label text-success"><input type="checkbox" checked id="c2">  I have read the eligibility criteria and confirm that i am eligible to donate blood.</label>
<label class="control-label text-success"><input type="checkbox" checked id="c3" >  I agree to the Term and Conditions and consent to have my contact and donor information published to the potential blood recipients.</label>
</div>
<div class="form-group">
<button class="btn btn-primary" type="submit" name="submit" >Registar Now</button>
</div>
</form>
</div>
</div>
</div>
</div>
</div>

<?php include("footer.php"); ?>
<script>
$(document).ready(
function(){
$("#volu").hide();
```

```
$("#c1").click(function(){
if($("#c1").is(':checked'))
{
$("#volu").show(1000);
$("#new").hide(100);
}
else
{
$("#volu").hide(1000);
$("#new").show(100);
}
});

/*
$("#CITY").change(function(){
var city=$("#CITY").val();
//alert(city);
$.post('functions.php',{G_CITY_ID:city},function(data){
//                      alert(data);
$("#STATE").html(data);
});

});*/
$("#COUNTRY").change(function(){
var countr=$("#COUNTRY").val();
//alert(city);
$.post('get_state.php',{G_STATE_ID:countr},function(data){
//                      alert(data);
$("#STATE").html(data);
});

});

$("#STATE").change(function(){
var stid=$("#STATE").val();
//alert(city);
$.post('get_city.php',{G_STATE_ID:stid},function(data){
//                      alert(data);
$("#CITY").html(data);
});
```

```
})

});
$(function() {
var availableTags = [
<?php
$sql="SELECT AREA_NAME FROM area";
$result=$con->query($sql);

if($result->num_rows>0)
{
$i=0;
$n=$result->num_rows;
while($row=$result->fetch_assoc())
{
$i++;
if($n!=$i)
{
echo '"'.$row['AREA_NAME'].'",';
}
else
{
echo '"'.$row['AREA_NAME'].'"';
}}
}
?>
];
$( "#AREA" ).autocomplete({
source: availableTags
});
});
</script>

</body>
</html>
```