**CHAPTER-1**

**INTRODUCTION**

**INTRODUCTION**

**Objective**

The main objective of this specification is to support the automated tracking of blood products from the initial ordering of a blood transfusion for a patient, through to the taking of a blood sample for cross matching, to administration of a blood transfusion and subsequent updates to care records.

To allow the probable recipients to make search and match the volunteer donors, and make request for the blood.

Blood Bank Management Software is designed & suitable for several Blood Bank either operating as individual organization or part of Hospital. It covers all Blood banking process from Donor recruitment, donor management, mobile sessions, component preparation, screening covering all tests, blood stock inventory maintenance, patient registration, cross matching, patient issues etc.

**Introduction**

This project is developed to manage the blood stock in the ‘‘BLOOD BANK‘’ and the blood prices are maintained in the database.

Blood sales and blood purchase are entered and maintained in this project. Blood stock reports, sales reports and blood purchase reports are managed in this project. It will help us to find the blood group with its most efficient time to take care of the blood and it is more easy to hand over the blood to the hospitals to help people to get blood on time.

This all thing is been stored and been seen in this Blood Bank Management System. To help more people trying best to do so.

The following is an attempt to put together a complete, yet reasonably flexible template for the specification of software designs. Wherever possible, I have tried to provide guidelines (instead of prescribing requirements) for the contents of various sections and subsections of the document. Some may prefer to require more detailed subsections of a particular section, choosing one or more of the subsection topics from the list of guidelines provided. In this sense, this document is really a template for a template.

It is my desire that a completed software design specification meet the following criteria:

* It should be able to adequately serve as training material for new project members, imparting to them enough information and understanding about the project implementation, so that they are able to understand what is being said in design meetings, and won't feel as if they are drowning when they are first asked to create or modify source code.
* It should serve as "objective evidence" that the designers and/or implementers are following through on their commitment to implement the functionality described in the requirements specification.
* It needs to be as detailed as possible, while at the same time not imposing too much of a burden on the designers and/or implementers that it becomes overly difficult to create or maintain.

**1.1 Intended User**

Anybody can use this BBMS to Donor as well as who need blood e.g., Public, Hospitals, Blood Banks, etc. This project main aim is to remove the old way of maintaining records on paper with the technology and providing the patients a better way to connect with the blood bank.

**1.2 Project Scope**

Primarily, the scope pertains to the Management product features for making **ONLINE**

**BLOOD** **MANAGEMENT** project.

Online blood bank aims for serving human welfare. We all have the information, you will ever need. Many people are here for you, to help you, willing to donate blood for you any time. You can help us by registering our online blood bank if you are willing to donate your blood when needed.

* The scope of the specification includes the following scenarios:
* Emergency issue of blood,
* Management of returned and unused blood units.

**1.3 Limitations**

* End User’s will not be able to get the information about the availability of the blood in the bank of which he/she donated.
* Only the Admin has all right to edit the things in the End User’s Profile.
* Reports takes time to produce.

**CHAPTER- 2**

**SYSTEM ANALYSIS**

**2.1 Overall Description**

**Product Prospective**

* To provide a means for the blood bank to publicize and advertise blood donation programs.
* To provide an efficient donor and blood stock management functions to the blood bank by recording the donor and blood details.
* To improve the efficiency of blood stock management by alerting the blood bank staffs when the blood quantity is below it par level or when the blood stock has expired.
* To provide synchronized and centralized donor and blood stock database.
* To provide immediate storage and retrieval of data and information.

**2.2 Login Interface**

User should enter the valid username and password to get access to its profile.

**2.3 Donor Profile**

User will be able to see its Account Number. The receipts of the blood donated to the bank, Donation to the Bank, Need of the Blood to the Bank and Request for Blood.

**2.4 Blood Stock Management**

It will show the Blood Detailed of the specific bottle with its Full Donor Detail or Account No. if he/she is registered to the Bank.

**2.5 Operating Environment**

**Operating system: Linux**

**Intel P4 1.5GHz or above.**

**512MB ram.**

**80GB HDD Minimum.**

**User Classes**

**System Owner: The Blood Bank**

**System Users: Administrators: has full privilege on the system's functions**

Public: can view the blood donation events and donate or can make requests for donation (Donor and Recipients fall under this category).

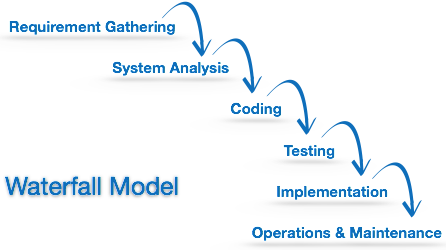
Design Constraints.

Software Development Paradigm

The software development paradigm helps developer to select a strategy to develop the software. A software development paradigm has its own set of tools, methods and procedures, which are expressed clearly and defines software development life cycle. A few of software development paradigms or process models are defined as follows:

Waterfall Model

Waterfall model is the simplest model of software development paradigm. It says the all the phases of SDLC will function one after another in linear manner. That is, when the first phase is finished then only the second phase will start and so on.



This model assumes that everything is carried out and taken place perfectly as planned in the previous stage and there is no need to think about the past issues that may arise in the next phase. This model does not work smoothly if there are some issues left at the previous step. The sequential nature of model does not allow us go back and undo or redo our actions.

This model is best suited when developers already have designed and developed similar software in the past and are aware of all its domains.

Then one standard SDLC model is used to build the software. In the fourth phase of the plan of next iteration is prepared.

# **SDLC - Waterfall Model**

The Waterfall Model was the first Process Model to be introduced. It is also referred to as a **linear-sequential life cycle model**. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

The Waterfall model is the earliest SDLC approach that was used for software development.

The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap.

Waterfall Model - Design

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

The following illustration is a representation of the different phases of the Waterfall Model.



The sequential phases in Waterfall model are −

* **Requirement Gathering and analysis** − All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
* **System Design** − The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
* **Implementation** − With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
* **Integration and Testing** − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
* **Deployment of system** − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
* **Maintenance** − There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model, phases do not overlap.

Waterfall Model - Application

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are −

* Requirements are very well documented, clear and fixed.
* Product definition is stable.
* Technology is understood and is not dynamic.
* There are no ambiguous requirements.
* Ample resources with required expertise are available to support the product.
* The project is short.

Waterfall Model - Advantages

The advantages of waterfall development are that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one.

Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order.

Some of the major advantages of the Waterfall Model are as follows −

* Simple and easy to understand and use
* Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
* Phases are processed and completed one at a time.
* Works well for smaller projects where requirements are very well understood.
* Clearly defined stages.
* Well understood milestones.
* Easy to arrange tasks.
* Process and results are well documented.

Waterfall Model - Disadvantages

The disadvantage of waterfall development is that it does not allow much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage.

The major disadvantages of the Waterfall Model are as follows −

* No working software is produced until late during the life cycle.
* High amounts of risk and uncertainty.
* Not a good model for complex and object-oriented projects.
* Poor model for long and ongoing projects.
* Not suitable for the projects where requirements are at a moderate to high risk of changing. So, risk and uncertainty is high with this process model.
* It is difficult to measure progress within stages.
* Cannot accommodate changing requirements.
* Adjusting scope during the life cycle can end a project.
* Integration is done as a "big-bang. at the very end, which doesn't allow identifying any technological or business bottleneck or challenges early.

**CHAPTER-3**

**FEASIBILITY STUDY**

**3 GENERAL INFORMATION**

## 3.1 Purpose

The purpose of this study 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 parts of the country basing on the hospital demands. Without quick and timely access to donor records, creating market strategies for blood donation and sensitizing of blood donors becomes very difficult.

The blood management information system offers functionalities to quick access to donor records collected from various parts of our country. It enables monitoring of the results and performance of the blood donation activity such that relevant and measurable objectives of the organization can be checked.

It provides management timely, confidential and secure medical reports that facilities planning and decision making and hence improved medical service delivery. The reports generated by the system give answer to most of the challenge management faces as far as blood donor records are concerned.

## 3.2 Scope

* The scope of this project is to maintain the blood bank information system.
* All information of donors/users details, information who wants to Dontae blood.

## 

## 3.3 System Overview

* A blood bank is a center where [blood](https://en.wikipedia.org/wiki/Blood) gathered as a result of [blood donation](https://en.wikipedia.org/wiki/Blood_donation) is stored and preserved for later use in [blood transfusion](https://en.wikipedia.org/wiki/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 (to reduce the risk of transfusion related adverse events). However, it sometimes refers to a collection center, and indeed some hospitals also perform collection.
* The Blood Bank Management System is great project. This project is designed for successful completion of project on blood bank management system.
* (BBMS) is a Web based application that is designed to store, process, retrieve and analyse information concerned with the administrative and inventory management within a blood bank.
* This blood bank management system is an online website so it is easily available to everyone.
* When a person want to donate blood he have to register to the system. Donor registration is very easy, to get register to the system he have to fill up registration form. After submitting the registration form he can create username and password. Donor have to give information like blood group, contact details etc. donor can also change his account information when he wants using his username and password.
* Using this blood bank system people can search blood group available which they are needed.
* They check it online using our blood bank management website. If in case blood group is not available in blood bank they can also get contact numbers of the persons who has the same blood group he is need. And he can request the person to donate the blood for saving someone life.
* Our system also allow user to search online the person who have the same blood group he needs and if he find the If he find a donor in his city then we give him all details of the donor, if he doesn’t find any donor then he is given the contact numbers and addresses of the Life Saving Contact Persons for big cities.

## 3.5 Points of Contact

### 3.5.1 Information

After thoroughly analyzing the existing system the following objectives were set:-

The application of blood bank management system are as follows:-

* User can check whether the blood is available or not.
* This system is used to check available amount of blood required.
* User can access whenever needed.
* The features of this project is searching facilities based on various factors. Such as blood

donors, different blood groups details.

* The project deals with monitoring the information blood type, storage area and data of

storage.

Editing, adding and updating of records is improved which results in proper resource

management of Blood Management System data.

### 3.5.2 Coordination

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No. | Name | E.no | Designation | Email ID |
| 1 | JIBIN JOSEPH |  |  |  |
| 2 | DISHANT DHINGRA |  |  |  |
| 3 | MEHUL KAKKAR |  |  |  |
| 4 | ANIKET KUMAR JHA |  |  |  |

##### Note:-As this is the training project created at college level as per our syllabus. So there is no interaction needed with any other organization.

## 

## 

## 3.6 Environment

### 3.6.1 Organizations Involved

Responsible organization: Surjan Singh Risal Singh Memorial Education Society

Sponsor: Sirifort College of Computer Technology And Management.

Developer: MEHUL KAKKAR, ANIKET KUMAR JHA, DISHANT DHINGRA, JIBIN JOSEPH

User: Admin, Data entry operator(DEO)

This project will be a website application to be developed in Java using JDK.

* NetBeans IDE 8.0.2

**3.6.2 Input/ Output**

### The system requires following input and output:-

### INPUT REQUIRMENTS

* User ID
* Email Address
* Password

OUTPUT REQUIRMENTS:-

* Mails
* Message
* Notices

### 3.6.3 Processing

HARDWARE REQUIRMENTS:-

Development Time

1. Memory minimum 32 GB (Approx.)
2. Other standard cabinet assembly

Deployment Time

1. Memory minimum 10 GB (Approx.)
2. Other standard cabinet assembly

DOCUMENTATION AND PRESENTATION TOOLS:-

1. Microsoft Word (2003)

SOFTWARE REQUIRMENTS

Development Time

1. Operating System (XP)
2. Browser
3. NetBeans IDE 8.0.2
4. Microsoft Office (2003)

Deployment Time

1. Operating System (XP)

2. Browser

QWERTY

### 3.6.4 Security

This program uses object oriented mechanisms to protect its data passed using methods. Also at the entry level login ID and password is required to enter into the system. After that a session is maintained at each level of module.

### 3.6.5 System Interaction

This is a college ERP website and we are creating it in parts. We are concerned with Inventory module and the other two modules will be created by the other group and our database is common and the Timetable and Inventory module will interact with the other modules i.e. , Login Form, sell and purchase module etc.

### 3.7 Physical Environment

Since we will be creating this module at the college level so we will be focusing on college modules and situations and then create the website and not focus on the organization.

## 3.7.1 Current Functional Procedures

* The current functional procedures of the system are purely manual as the system cannot perform any task of its own. User has to give the command to perform any specific task then only system will produce output.
* The cost incurred in operating the current system is considerable as it is a project of a high-level language.
* The people involved in the project should be skilled up to the level at which they are familiar with the JAVA Programming, DFD, ER Diagrams, NetBeans.
* The current system requires only few embers of staff, as the maintenance is not so much required.
* Excellent attention to details.
* Proven ability to work as a team members.

## 3.7.2 Functional Objectives

Analyze the anticipated functions of the system, considering such areas as new services, increased capacity, legislative and policy requirements, privacy and security requirements, audit controls, and target completion date.

The following objective must be taken into account while developing an ERP for a college.

Speed - Speed is defined as the time taken to implement the ERP system in a college. While designing ERP, the time taken by the organization to implement ERP must be taken into account.

Scope – ERP system should consider and include all the functional and technical characteristics required by the college.

Resources – Resources are everything that is needed to support the project. This includes people, hardware, system, software, etc.

Risk - Risk is defined as the factor that resists the overall success of ERP implementation. Therefore, all the risk that will arise must be taken into account while developing ERP.

Complexity – Complexity is the degree of difficulty anticipated during implementation, operation and maintaining the ERP systems.

Benefits – To get the maximum benefit from ERP, care must be taken to design the ERP by following the procedures followed by the organization.

## 3.8 Performance Objectives

All the software and the products used to develop the project are used efficiently and the project that is being developed by us is managed properly.

Major Performance Objectives include:-

Staff: as task will become easier and automated, so less staff members are required to maintain all the information.

Processing speed: as manual task is reduced, processing speed of workwill be increased.

Services:-management information services will be improved due to ease to get information from database.

Decision making system: - control over decision making system will be improved due to simple click and get method. This click and get method is required to reduce the effort of getting the information from database.

## 3.9 Assumptions and Constraints

Different assumptions and constraints are taken in consideration while preparing this system such as:

Operation life:-This is the training project made by BCA students at college level. It is operational till the end of the training.

Period of time for comparison of system alternatives:-Before deciding the different modules of the project, around two months devoted in the research of the modules provided by the different other alternative system.

Operation environment:-It is assumed that it will perfectly run ion any operating environment.

Availability of information:-It is assumed that all the information that are required for this project is available while developing the logics of the project.

## 3.10 Methodology

We studied on the internet regarding the project content to make it effective and use by a simple person. We have also various projects to make our project worth of doing all the tasks, which are not done by other projects.

WATERFALL MODEL

Requirement Specification

Analysis & Design

Implementation & unit testing

Integration & System Testing

Maintenance

## 3.11 Evaluation Criteria

A responsibility of this application is to provide access to view the demand and availability of the inventories so that the work is efficient and there will be no wastage of time and memory.

Priority: Priority is given due to following reasons

Easy Access to Database.

Reduce man power resources.

User friendly approach.

Reduced paper work.

Fast and easy to use

Security

## 3.12 Recommendation

We take into account the suggestion from the participation of the survey and try to implement all the features that we can put into the application.

##### 3.13 PROPOSED SYSTEM

## 3.14 Description of Proposed System

This system will be used in two user modules which are ADMIN,DATA ENTRY OPERATOR(DEO).

As all of these have different requirements, the modules are designed to meet their need and avoid any type of confusion. The uses of all two user modules are described below:-

1. ADMIN

* + Login
  + View item details
  + View sale receipts
  + View purchase receipts
  + View damaged item details
  + View static information
  + Querying

## 2. DATA ENTRY OPERATOR (DEO)

* + Login
  + Add item details
  + View item details
  + Add sale receipts
  + Add purchase receipts
  + Add damaged item details
  + View static information

## 3.15 Improvements

Improvements are the modification that is regularly needed in any system at the maintenance time to improve its action.

The improvements of the system in terms of the objectives are as follows:-

* The project will be more feasible.
* An application that tells the timing and have records of student login for protection of their account.

## 3.16 Time and Resource Costs

As this is the project made at college level so there is no much funding require in monetary terms but need to utilize more time in collection of data through the means of survey conducted at the different organizations and societies through distributing on paper forms and online Google forms.

Also more time is utilize in collecting information and resources required in developing the project.

This is the project made by a group at college level so there is no staff needed to be recruited for different purposes.

We have invested 5 MONTHS to collect the information needed for development of the project.

### 3.18 Equipment Impacts

Updated and currently launched hardware are required for perfectly running this project.

We are using NetBeans IDE 8.0.2 and JDK (database); no other equipment requirement is there

### 3.19 Software Impacts

To run this system accurately, it is preferred to use browser like Google Chrome, Mozilla Firefox, etc.

### NetBeans IDE 8.0.2 for the development of system.

### 3.20 Organizational Impacts

In order to maintain coordination between the different modules of the project (Inventory Module) the staff needs to have some technical knowledge about the organization or the basic system knowledge.

### 3.21 Operational Impacts

Effects of different operations on the system, as:

\* User operating procedures: User needs to have some knowledge about the procedure to communicate with system. It will be improved due to user friendly interface.

\* Operating center procedures: Different operating centers (staff) have different facilities to operate with the system. The project will benefit the management and developers, as manual work and efforts will be reduced.

\* Operating center/user relationships: The operating center\user relationships will improve if we are provided some information about the project from the college campus.

\* Source data processing: This will depends on the needs in project completion.

\* Data entry procedures: Forms having buttons, text-boxes, checkbox, data grid view etc. All the details of various users, students, companies, books etc needs to be entered into the database.

\* Data retention requirements: These are provided itself in the database, information storage and retrieval procedures are provided in the database. Database will be non-redundant.

\* Output reporting procedures, media, and schedules: Media involved are the machine systems. Schedules are user defined.

\* System failure contingencies and recovery procedures: Backup of the database is needed in case of emergency or system failure.

### 3.22 Developmental Impacts

Describe the developmental impacts, such as:

* Specific activities to be performed by the user in support of development of the system
* Resources required to develop databases
* Computer processing resources required to develop and test the new system
* Privacy and security implications

### 3.23 Site or Facility Impacts

All the project work is done in the college campus.

### 3.24 Security and Privacy Impacts

Information Security:-The security of information is very important. Information can be of items, prices or any one must be secure.

Physical Security:-Destruction/damage of any asset of item may cause fine.

Personnel Security:-Disclosure, modification, interruption, disruption, removal of personal information by disgruntled.

Information Protection (IP):-The information of USER and PRODUCT must be secure so no one can misuse the information.

We have installed Password on the system to protect the loss or theft of data.

## 3.25 Rationale for Recommendations

There are many alternative systems that are already in work. After judging those systems, we have made many improvements in our system such as admin update the detail of the inventory .

3.26 Alternative System

## 3.27 Description of [Manual Work]

There is an alternative system where admin will login to his/her account and check that the student had paid fine or not.

They also get to know that how much fine a particular student has to pay.

There is also a system where admin will login his/her account and check that the queries of the student and respond to them. Admin will generate the fine receipt. Admin can update student’s account.

There is a system where student will register his/her account and ask queries. Student can pay fine and update his/her account.

**CHAPTER-4**

**SYSTEM DESIGN**

About System Architecture Interaction system was helpful to separate work and interaction. It promoted employees to make the lesson planning and work into college and send their information to the admin. And also it helps the admin to get information about the employees in detail such as: update, view, manage their salaries, provide incentives, bonus etc.

**DATA FLOW DIAGRAM**

A DFD also known as 'bubble chart', has the purpose of clarifying system requirements and identifying major transformations. It shows the flow of data through a system. It is a graphical tool because it represent increasing information flow and functional detail. Four simple notations are used to complete a DFD. These notations are given below:-

**DATA FLOW**

The data flow is used to describe the movement of information from one part of the system to another part. Flows represent data in motion. It is a pipe line through which information flows. Data flow is represented by an arrow.

**PROCESS**

A circle or bubble represents a process that transforms incoming data to outgoing data. Process shows a part of the system that transforms inputs to outputs.

PROCESS

**EXTERNAL ENTITY**

A square defines a source or destination of system data. External entities represent any entity that supplies or receives information from the system but is not a part of the system.

EXTERNAL ENTITY

ENTITY

**DATA STORE**

The data store represents a logical file. A logical file can represent either a data store symbol which can represent either a data structure or physical file on disk. The data store is used to collect data at rest or temporary repository of data. It is represented by open rectangle.

**OUTPUT**

The output symbols used when a hard copy is produced and the user of the copies cannot be clearly specified or there are several users of the output.

11.4.1 Subsystem Architecture

If a particular component is one which merits a more detailed discussion than what was presented in the System Architecture section, provide that more detailed discussion in a subsection of the System Architecture section (or it may even be more appropriate to describe

the component in its own design document). If necessary, describe how the component was further divided into subcomponents, and the relationships and interactions between the

subcomponents (similar to what was done for top-level components in the System Architecture section). If any subcomponents are also deemed to merit further discussion, then describe them in a separate subsection of this section (and in a similar fashion). Proceed to go into as many levels/subsections of discussion as needed in order for the reader to gain a high-level understanding of the entire system or subsystem (but remember to leave the gory details for the Detailed System Design section).

If this component is very large and/or complex, you may want to consider documenting its design in a separate document and simply including a reference to it in this section. If this is the option you choose, the design document for this component should have an organizational format that is very similar (if not identical to) this document.

**Policies and Tactics**

Administrators are assigned with their default password and they can change it afterwards. • Password can be changed only after Logging in. • For coding of different forms we have used separate files and query strings. A simple HTML form calls a file for insertion in database. • For easy understanding of file operation simple names are used for naming the files. • Every administrator need first login to go in the corresponding section

**Detailed System Design**

The application will have a user friendly and menu based interface. It will have a login screen for entering user-name, password will be provided. Access to different screens will be based upon the user. There is a screen for displaying information regarding filling of student details. **Classification**

Web browser was used as client asp.net as the business logic tier to achieve its function, and database as the data layer: 1. Client: Client was web browser which implemented the systems display logic. The function was to send request to web server through the web browser by user (buyer and dealer). While web server returns the requested html pages or html pages dynamically generated to the client, which were showing the web browser.

**Business logic tier:** It was achieved mainly by asp.net. It responded to the client request and achieved the business logic with web server.

**Data tier:** It was realized with database system; use to store the business data such as property information and control data such as user data SQL server was used to achieve the data tier.

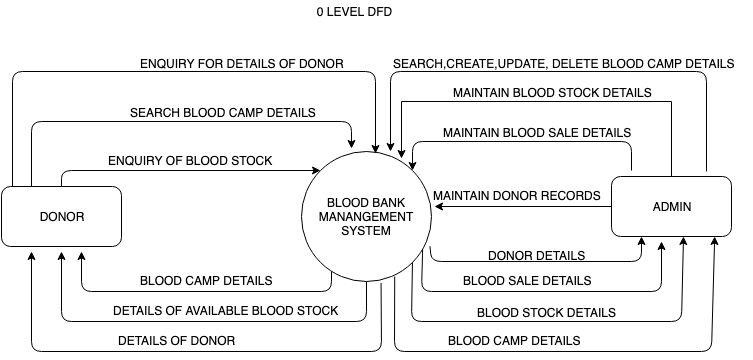
**Definition** The specific purpose and semantic meaning of the component, this may need to refer back to the requirements specification.

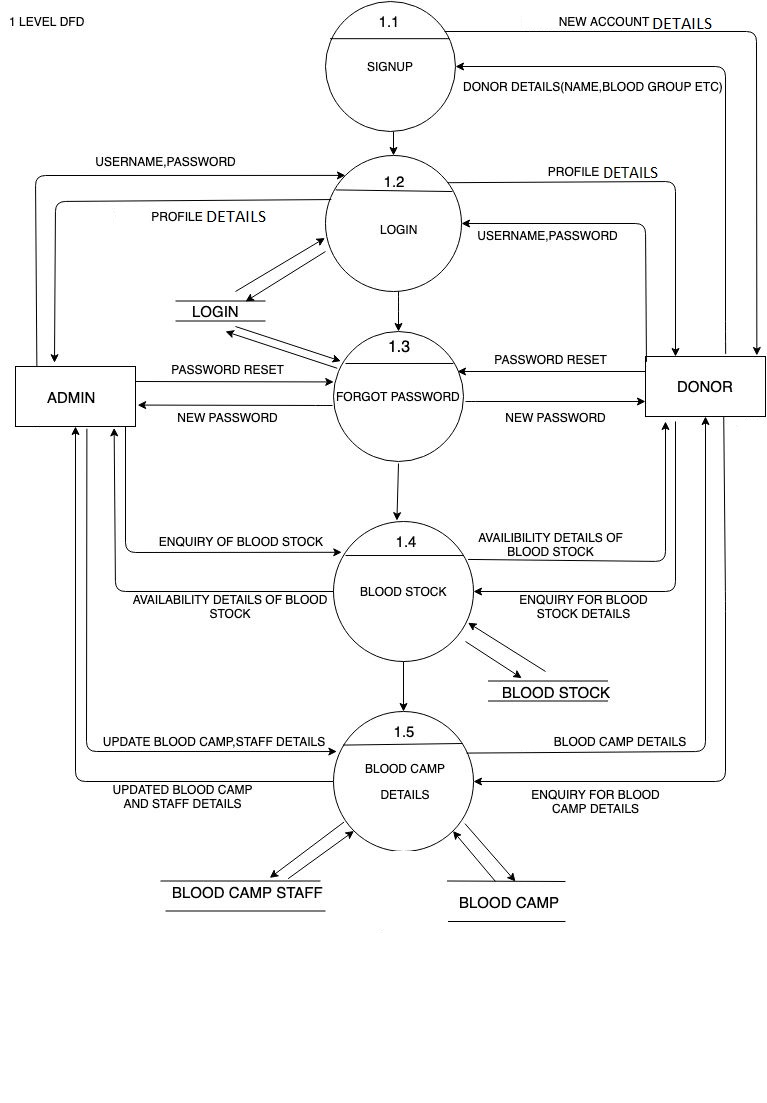
**Responsibilities** The primary responsibilities and/or behavior of this component are: What does this component accomplish? What roles does it play? What kinds of services does it provide to its clients? For some components, this may need to refer back to the requirements specification

**Constraints-** Any relevant assumption, limitations, or constraints for this component. This should include constraints on timing, storage, or component state, and might rules for interacting with this component.

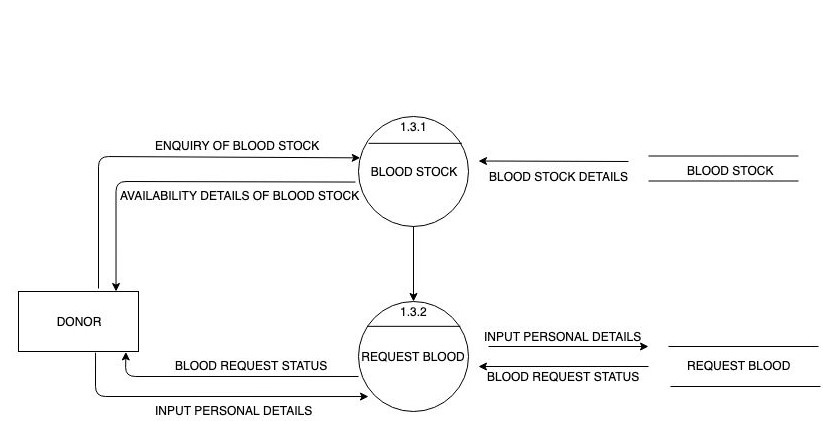
**Composition** A description of the use and meaning of the subcomponents that are a part of this component.

**Database Design-** The database design specifies how the date of the software is going to be stored. 11.6.7 Table schemas The complete (compliable) set of CREATE TABLE statements (and other SQL statements) that declare the database schema, including integrity constraints, domain specifications, assertions, and access privileges -- documented in a template with the intended use of each table and column.

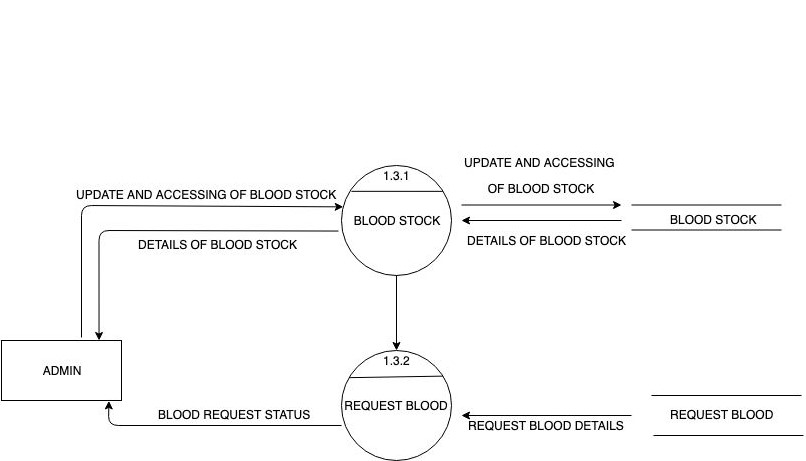


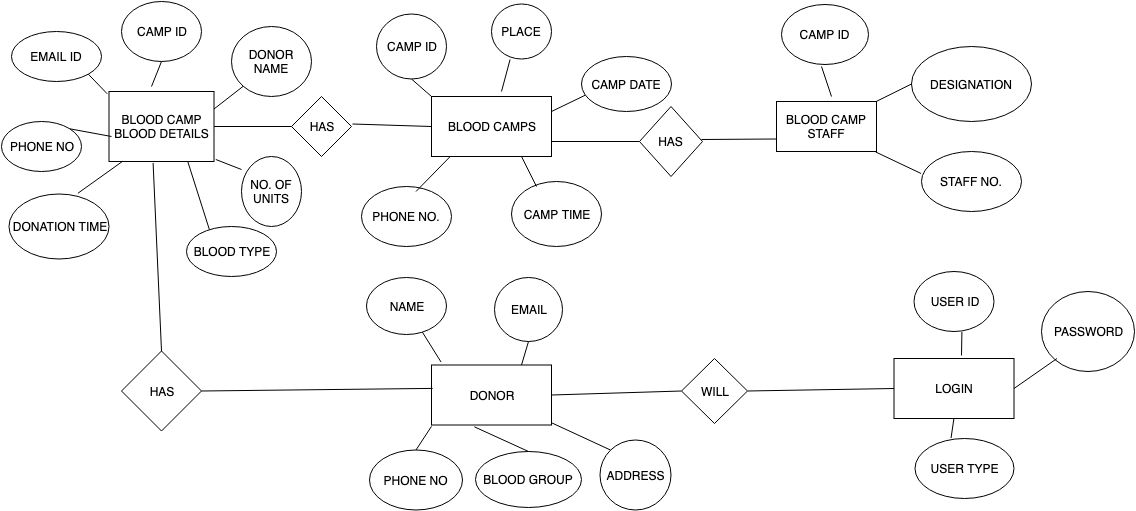


**LEVEL 2 DFD BLOOD STOCK DONOR SIDE**



**LEVEL 2 DFD BLOOD STOCK ADMIN SIDE**





**Hardware Interfaces**

The System must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for e.g. Modem, WAN – LAN, Ethernet Cross Cable and WIFI. The System requires Database also for the store of any usage of the system like Derby etc. System also requires DNS for the naming on the internet. At the last user need web browser for interacting with the system.

**Software Interfaces**

The system is on server so it requires any scripting language like JavaScript etc. The system requires Database also for storing any transaction of the system like Derby etc. System also require server on the internet like Glass fish server. System also requires DNS for the naming on the internet. At the last, user needs web browser for interacting with the system

**Communications Interfaces**

The management system shall use the HTTPS protocol for communication over the Internet and for the Intranet communication will be through TCP/IP protocol suite. The user must have SSL certificate licensing registered web browser.

|  |  |  |  |
| --- | --- | --- | --- |
| **MODULES** | | |  |
| S.NO. | MODULES | DISCRIPTION |  |
| 1 | BLOOD STOCK | It gives the discription regarding the availability of blood of a particular type. |  |
| 2 | BLOOD CAMPS | It Gives the description regarding the blood camps to held in future and held in the past |  |
| 3 | LOGIN | It includes login of the admin as well as the donor on the basis of usertype , user id and password |  |
| 4 | ABOUT US | It gives the description regarding the organisation and its history |  |
| 5 | GALLERY | It gives photos of events, camps for the viewers. |  |
| 6 | CONTACT US | This gives the contact info. Of organization. |  |
| 7 | TERMS AND CONDITION | This gives information regarding terms and condition regarding donations. |  |
|  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **BLOOD CAMP STAFF** | | | |  |
| DISCRIPTION | THIS TABLE SHOWS DETAILS REGARDING STAFF AT BLOOD CAMPS | | |  |
| ATTRIBUTE | DISCRIPTION | TYPE | EXAMPLE |  |
| CAMP ID | IT SPECIFIES THE CAMP ID AT WHICH STAFF IS ENGAGED | varchar(25) | JAN0025 |  |
| STAFF ID | SPECIFIES THE STAFF ID | integer | 2879 |  |
| STAFF NAME | IT SPECIFIES THE NAME OF STAFF | varchar(20) | MR. ARUNESH KUMAR |  |
| DESIGNATION | SPECIFIES THE DESIGNATION OF STAFF | varchar(20) | [NURSE](mailto:mkakkar47@gmail.com) |  |
| PRIMARY KEY | STAFF ID | | |  |
| FOREIGN KEY | CAMP ID | | |  |
| SQL CODE | Create table blood camp Staff (staff id integer, camp id varchar (25),staff name varchar(20), designation varchar(20); |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DONOR** | | | |  |
| DISCRIPTION | IT SHOWS THE INFO OR DETAILS OF DONOR | | |  |
| ATTRIBUTE | DISCRIPTION | TYPE | EXAMPLE |  |
| EMAIL ID | SPECIFIES EMAIL ID OF DONOR | varchar(20) | [mkakkar47@gmail.com](mailto:mkakkar47@gmail.com) |  |
| NAME | SPECIFIED NAME OF DONOR | varchar(30) | MR.Dishant Dhingra |  |
| PHONE NO. | IT SPECIFIES THE NO. OF UNITS OF BLOOD DONATED | Integer | 10 |  |
| ADDRESS | SPECIFIED ADDRESS OF DONOR | varchar(50) | [Sector-4,Rohini,DELHI](mailto:mkakkar47@gmail.com) |  |
| BLOOD GROUP | SPECIFIES BLOOD GROUP OF DONOR | varchar(3) | B+ |  |
| PRIMARY KEY | NAME | | |  |
| FOREIGN KEY | EMAIL | | |  |
| SQL CODE | Create table donor(email varchar(20),name varchar(30),phone no. integer,address varchar(50),blood group varchar(3); |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **BLOOD CAMPS** | | | |  |
| DISCRIPTION | THIS TABLE SHOWS DETAILS REGARDING BLOOD CAMPS | | |  |
| ATTRIBUTE | DISCRIPTION | TYPE | EXAMPLE |  |
| CAMP ID | IT SPECIFIES THE CAMP ID AT WHICH THE BLOOD IS DONATED | varchar(25) | JAN0025 |  |
| PLACE | SPECIFIED THE PLACE WHERE CAMP SETUP | varchar(50) | RK PURAM DELHI 110055 |  |
| CAMP DATE | IT SPECIFIES THE DATE OF CAMP | varchar(10) | 20-11-2019 |  |
| CAMP TIME | SPECIFIED THE TIME OF CAMP | varchar(10) | [10:00 AM](mailto:mkakkar47@gmail.com) |  |
| CONTACT | SPECIFIES THE CONTACT OF THE CAMP MANAGEMENT | Integer | 1182858977 |  |
| PRIMARY KEY | CAMP ID | | |  |
| FOREIGN KEY | CAMP ID | | |  |
| SQL CODE | Create table blood camp (camp id varchar(25), place varchar(50), camp date varchar(10), conatact integer; |  |  |  |
|  | | | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **BLOOD STOCK** | | | |
| DISCRIPTION | THIS TABLE SHOWS THE STOCK OF BLOOD OF PARTICULAR TYPE | | |
| ATTRIBUTE | DISCRIPTION | TYPE | EXAMPLE |
| BLOOD TYPE | IT SPECIFIES THE BLOOD TYPE | varchar(30) | MEHUL123 |
| NO. OF UNITS | SPECIFIES THE AMOUNT OF BLOOD AVAILABLE | varchar(20) | [BLOOD@123](mailto:BLOOD@123) |
| PRIMARY KEY | USER ID | | |
| FOREIGN KEY |  | | |
| SQL CODE | Create table blood stock (bloodtype varchar(4), no. of units integer; |  |  |
|  | | | |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **BLOOD CAMP BLOOD** | | | |  |
| DISCRIPTION | THIS TABLE SHOWS THE BLOOD DETAILS AT BLOOD CAMPS | | |  |
| ATTRIBUTE | DISCRIPTION | TYPE | EXAMPLE |  |
| CAMP ID | IT SPECIFIES THE CAMP ID AT WHICH THE BLOOD IS DONATED | varchar(25) | JAN0025 |  |
| DONOR NAME | SPECIFIED NAME OF DONOR | varchar(30) | MR.Dishant Dhingra |  |
| NO. OF UNITS | IT SPECIFIES THE NO. OF UNITS OF BLOOD DONATED | Integer | 10 |  |
| EMAIL ID | SPECIFIED ADDRESS OF DONOR | varchar(50) | [mkakkar47@gmail.com](mailto:mkakkar47@gmail.com) |  |
| BLOOD TYPE | SPECIFIES BLOOD GROUP OF DONOR | varchar(4) | B+ |  |
| PHONE NO. | SPECIFIED CONTACT NO. OF DONOR | Integer | 9871499715 |  |
| PRIMARY KEY | EMAIL | | |  |
| FOREIGN KEY | CAMP ID | | |  |
| SQL CODE | Create table blood camp blood(camp id varchar(25)),donor name varchar(30),No. of units Integer , blood type varchar(4); |  |  |  |
|  | | | |  |

# **CHAPTER-5**

# **Software Testing Overview**

Software Testing is evaluation of the software against requirements gathered from users and system specifications. Testing is conducted at the phase level in software development life cycle or at module level in program code. Software testing comprises of Validation and Verification.

Software Validation

Validation is process of examining whether or not the software satisfies the user requirements. It is carried out at the end of the SDLC. If the software matches requirements for which it was made, it is validated.

* Validation ensures the product under development is as per the user requirements.
* Validation answers the question – "Are we developing the product which attempts all that user needs from this software ?".
* Validation emphasizes on user requirements.

Software Verification

Verification is the process of confirming if the software is meeting the business requirements, and is developed adhering to the proper specifications and methodologies.

* Verification ensures the product being developed is according to design specifications.
* Verification answers the question– "Are we developing this product by firmly following all design specifications ?"
* Verifications concentrates on the design and system specifications.

Target of the test are -

* **Errors** - These are actual coding mistakes made by developers. In addition, there is a difference in output of software and desired output, is considered as an error.
* **Fault** - When error exists fault occurs. A fault, also known as a bug, is a result of an error which can cause system to fail.
* **Failure**- failure is said to be the inability of the system to perform the desired task. Failure occurs when fault exists in the system.

Manual Vs Automated Testing

Testing can either be done manually or using an automated testing tool:

* **Manual** - This testing is performed without taking help of automated testing tools. The software tester prepares test cases for different sections and levels of the code, executes the tests and reports the result to the manager.

Manual testing is time and resource consuming. The tester needs to confirm whether or not right test cases are used. Major portion of testing involves manual testing.

* **Automated** This testing is a testing procedure done with aid of automated testing tools. The limitations with manual testing can be overcome using automated test tools.

A test needs to check if a webpage can be opened in Internet Explorer. This can be easily done with manual testing. But to check if the web-server can take the load of 1 million users, it is quite impossible to test manually.

There are software and hardware tools which helps tester in conducting load testing, stress testing, regression testing.

Testing Approaches

Tests can be conducted based on two approaches –

* Functionality testing
* Implementation testing

When functionality is being tested without taking the actual implementation in concern it is known as black-box testing. The other side is known as white-box testing where not only functionality is tested but the way it is implemented is also analyzed.

Exhaustive tests are the best-desired method for a perfect testing. Every single possible value in the range of the input and output values is tested. It is not possible to test each and every value in real world scenario if the range of values is large.

Black-box testing

It is carried out to test functionality of the program. It is also called ‘Behavioral’ testing. The tester in this case, has a set of input values and respective desired results. On providing input, if the output matches with the desired results, the program is tested ‘ok’, and problematic otherwise.



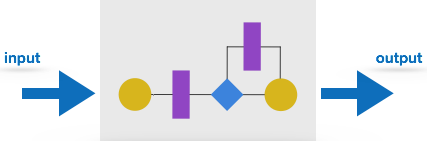
In this testing method, the design and structure of the code are not known to the tester, and testing engineers and end users conduct this test on the software.

Black-box testing techniques:

* **Equivalence class** - The input is divided into similar classes. If one element of a class passes the test, it is assumed that all the class is passed.
* **Boundary values** - The input is divided into higher and lower end values. If these values pass the test, it is assumed that all values in between may pass too.
* **Cause-effect graphing** - In both previous methods, only one input value at a time is tested. Cause (input) – Effect (output) is a testing technique where combinations of input values are tested in a systematic way.
* **Pair-wise Testing** - The behavior of software depends on multiple parameters. In pairwise testing, the multiple parameters are tested pair-wise for their different values.
* **State-based testing** - The system changes state on provision of input. These systems are tested based on their states and input.

White-box testing

It is conducted to test program and its implementation, in order to improve code efficiency or structure. It is also known as ‘Structural’ testing.



In this testing method, the design and structure of the code are known to the tester. Programmers of the code conduct this test on the code.

The below are some White-box testing techniques:

* **Control-flow testing** - The purpose of the control-flow testing to set up test cases which covers all statements and branch conditions. The branch conditions are tested for both being true and false, so that all statements can be covered.
* **Data-flow testing** - This testing technique emphasis to cover all the data variables included in the program. It tests where the variables were declared and defined and where they were used or changed.

Testing Levels

Testing itself may be defined at various levels of SDLC. The testing process runs parallel to software development. Before jumping on the next stage, a stage is tested, validated and verified.

Testing separately is done just to make sure that there are no hidden bugs or issues left in the software. Software is tested on various levels -

Unit Testing

While coding, the programmer performs some tests on that unit of program to know if it is error free. Testing is performed under white-box testing approach. Unit testing helps developers decide that individual units of the program are working as per requirement and are error free.

Integration Testing

Even if the units of software are working fine individually, there is a need to find out if the units if integrated together would also work without errors. For example, argument passing and data updation etc.

System Testing

The software is compiled as product and then it is tested as a whole. This can be accomplished using one or more of the following tests:

* **Functionality testing** - Tests all functionalities of the software against the requirement.
* **Performance testing** - This test proves how efficient the software is. It tests the effectiveness and average time taken by the software to do desired task. Performance testing is done by means of load testing and stress testing where the software is put under high user and data load under various environment conditions.
* **Security & Portability** - These tests are done when the software is meant to work on various platforms and accessed by number of persons.

Acceptance Testing

When the software is ready to hand over to the customer it has to go through last phase of testing where it is tested for user-interaction and response. This is important because even if the software matches all user requirements and if user does not like the way it appears or works, it may be rejected.

* **Alpha testing** - The team of developer themselves perform alpha testing by using the system as if it is being used in work environment. They try to find out how user would react to some action in software and how the system should respond to inputs.
* **Beta testing** - After the software is tested internally, it is handed over to the users to use it under their production environment only for testing purpose. This is not as yet the delivered product. Developers expect that users at this stage will bring minute problems, which were skipped to attend.

Regression Testing

Whenever a software product is updated with new code, feature or functionality, it is tested thoroughly to detect if there is any negative impact of the added code. This is known as regression testing.

Testing Documentation

Testing documents are prepared at different stages -

Before Testing

Testing starts with test cases generation. Following documents are needed for reference –

* **SRS document** - Functional Requirements document
* **Test Policy document** - This describes how far testing should take place before releasing the product.
* **Test Strategy document** - This mentions detail aspects of test team, responsibility matrix and rights/responsibility of test manager and test engineer.
* **Traceability Matrix document** - This is SDLC document, which is related to requirement gathering process. As new requirements come, they are added to this matrix. These matrices help testers know the source of requirement. They can be traced forward and backward.

While Being Tested

The following documents may be required while testing is started and is being done:

* **Test Case document** - This document contains list of tests required to be conducted. It includes Unit test plan, Integration test plan, System test plan and Acceptance test plan.
* **Test description** - This document is a detailed description of all test cases and procedures to execute them.
* **Test case report** - This document contains test case report as a result of the test.
* **Test logs** - This document contains test logs for every test case report.

After Testing

The following documents may be generated after testing :

* **Test summary** - This test summary is collective analysis of all test reports and logs. It summarizes and concludes if the software is ready to be launched. The software is released under version control system if it is ready to launch.

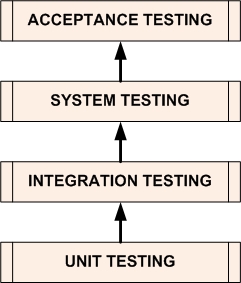
Testing vs. Quality Control, Quality Assurance and Audit

We need to understand that software testing is different from software quality assurance, software quality control and software auditing.

* **Software quality assurance** - These are software development process monitoring means, by which it is assured that all the measures are taken as per the standards of organization. This monitoring is done to make sure that proper software development methods were followed.
* **Software quality control** - This is a system to maintain the quality of software product. It may include functional and non-functional aspects of software product, which enhance the goodwill of the organization. This system makes sure that the customer is receiving quality product for their requirement and the product certified as ‘fit for use’.
* **Software audit** - This is a review of procedure used by the organization to develop the software. A team of auditors, independent of development team examines the software process, procedure, requirements and other aspects of SDLC. The purpose of software audit is to check that software and its development process, both conform standards, rules and regulations.

# **Software Testing Levels**

**SOFTWARE TESTING LEVELS** are the different stages of the software development lifecycle where testing is conducted. There are four levels of software testing: Unit >> Integration >> System >> Acceptance.



## **Levels**

|  |  |
| --- | --- |
| **Level** | **Summary** |
| [Unit Testing](http://softwaretestingfundamentals.com/unit-testing/) | A level of the software testing process where individual units of a software are tested. The purpose is to validate that each unit of the software performs as designed. |
| [Integration Testing](http://softwaretestingfundamentals.com/integration-testing/) | A level of the software testing process where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. |
| [System Testing](http://softwaretestingfundamentals.com/system-testing/) | A level of the software testing process where a complete, integrated system is tested. The purpose of this test is to evaluate the system’s compliance with the specified requirements. |
| [Acceptance Testing](http://softwaretestingfundamentals.com/acceptance-testing/) | A level of the software testing process where a system is tested for acceptability. The purpose of this test is to evaluate the system’s compliance with the business requirements and assess whether it is acceptable for delivery. |

## **CHAPTER-6**

## **SOFTWARE MAINTENANCE**

## **Importance of software maintenance**

The key software maintenance issues are both managerial and technical. Key management issues are: alignment with customer priorities, staffing, which organization does maintenance, estimating costs. Key technical issues are: limited understanding, [impact analysis](https://en.wikipedia.org/wiki/Change_impact_analysis), testing, maintainability measurement.

Software maintenance is a very broad activity that includes error correction, enhancements of capabilities, deletion of obsolete capabilities, and optimization. Because change is inevitable, mechanisms must be developed for evaluation, controlling and making modifications.

So any work done to change the software after it is in operation is considered to be maintenance work. The purpose is to preserve the value of software over the time. The value can be enhanced by expanding the customer base, meeting additional requirements, becoming easier to use, more efficient and employing newer technology. Maintenance may span for 20 years,[ whereas development may be 1–2 years.[

## **Software maintenance planning**

An integral part of software is the maintenance one, which requires an accurate maintenance plan to be prepared during the software development. It should specify how users will request modifications or report problems. The budget should include resource and cost estimates. A new decision should be addressed for the developing of every new system feature and its quality objectives. The software maintenance, which can last for 5–6 years (or even decades) after the development process, calls for an effective plan which can address the scope of software maintenance, the tailoring of the post delivery/deployment process, the designation of who will provide maintenance, and an estimate of the life-cycle costs. The selection of proper enforcement of standards is the challenging task right from early stage of software engineering which has not got definite importance by the concerned stakeholders.

## **Software maintenance processes**

This section describes the six software maintenance processes as:

1. The implementation process contains software preparation and transition activities, such as the conception and creation of the maintenance plan; the preparation for handling problems identified during development; and the follow-up on product configuration management.
2. The problem and modification analysis process, which is executed once the application has become the responsibility of the maintenance group. The maintenance programmer must analyze each request, confirm it (by reproducing the situation) and check its validity, investigate it and propose a solution, document the request and the solution proposal, and finally, obtain all the required authorizations to apply the modifications.
3. The process considering the implementation of the modification itself.
4. The process acceptance of the modification, by confirming the modified work with the individual who submitted the request in order to make sure the modification provided a solution.
5. The migration process ([platform migration](https://en.wikipedia.org/wiki/Software_migration), for example) is exceptional, and is not part of daily maintenance tasks. If the software must be ported to another platform without any
6. change in functionality, this process will be used and a maintenance project team is likely to be assigned to this task.
7. Finally, the last maintenance process, also an event which does not occur on a daily basis, is the retirement of a piece of software.

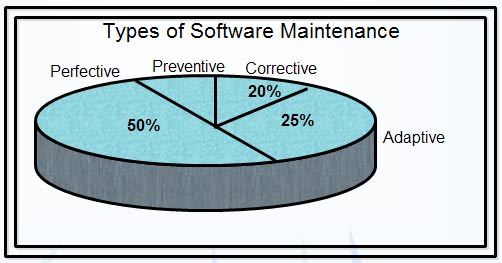
There are a number of processes, activities and practices that are unique to maintainers, for example:

* Transition: a controlled and coordinated sequence of activities during which a system is transferred progressively from the developer to the maintainer;
* [Service Level Agreements](https://en.wikipedia.org/wiki/Service_Level_Agreement) (SLAs) and specialized (domain-specific) maintenance contracts negotiated by maintainers;
* Modification Request and Problem Report Help Desk: a problem-handling process used by maintainers to prioritize, documents and route the requests they receive;

# [Types of Software Maintenance](http://ecomputernotes.com/software-engineering/types-of-software-maintenance)

There are four types of maintenance, namely, corrective, adaptive, perfective, and preventive.

Corrective maintenance deals with the repair of faults or defects found in day-today system functions. A defect can result due to errors in software design, logic and coding. Design errors occur when changes made to the software are incorrect, incomplete, wrongly communicated, or the change request is misunderstood. Logical errors result from invalid tests and conclusions, incorrect implementation of design specifications, faulty logic flow, or incomplete test of data. All these errors, referred to as residual errors, prevent the software from conforming to its agreed specifications. Note that the need for corrective maintenance is usually initiated by bug reports drawn by the users.



In the event of a system failure due to an error, actions are taken to restore the operation of the software system. The approach in corrective maintenance is to locate the original specifications in order to determine what the system was originally designed to do. However, due to pressure from management, the maintenance team sometimes resorts to emergency fixes known as patching. Corrective maintenance accounts for 20% of all the maintenance activities.

## **Adaptive Maintenance**

Adaptive maintenance is the implementation of changes in a part of the system, which has been affected by a change that occurred in some other part of the system. Adaptive maintenance consists of adapting software to changes in the environment such as the hardware or the operating system. The term environment in this context refers to the conditions and the influences which act (from outside) on the system. For example, business rules, work patterns, and government policies have a significant impact on the software system.

For instance, a government policy to use a single 'European currency' will have a significant effect on the software system. An acceptance of this change will require banks in various member countries to make significant changes in their software systems to accommodate this currency. Adaptive maintenance accounts for 25% of all the maintenance activities.

## **Perfective Maintenance**

Perfective maintenance mainly deals with implementing new or changed user requirements. Perfective maintenance involves making functional enhancements to the system in addition to the activities to increase the system's performance even when the changes have not been suggested by faults. This includes enhancing both the function and efficiency of the code and changing the functionalities of the system as per the users' changing needs.

Examples of perfective maintenance include modifying the payroll program to incorporate a new union settlement and adding a new report in the sales analysis system. Perfective maintenance accounts for 50%, that is, the largest of all the maintenance activities.

## **Preventive Maintenance**

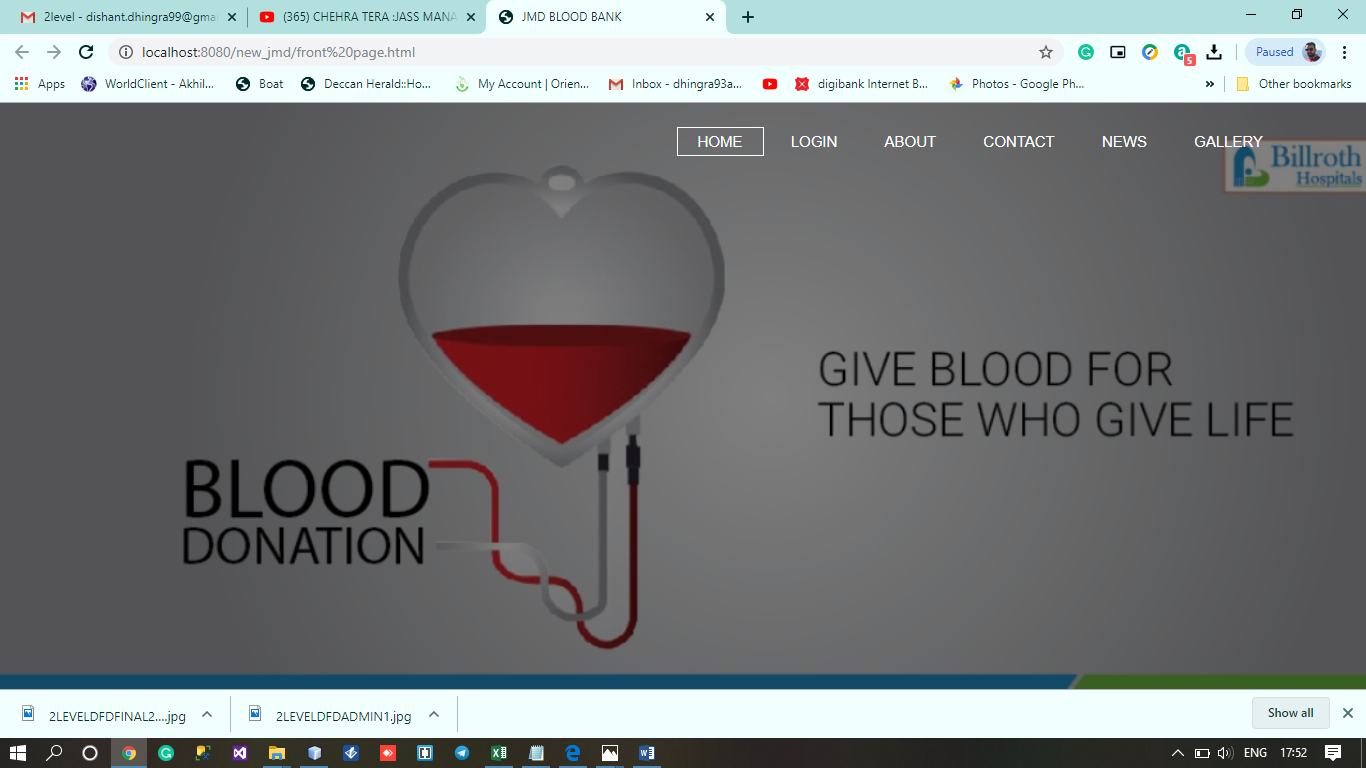
Preventive maintenance involves performing activities to prevent the occurrence of errors. It tends to reduce the software complexity thereby improving program understandability and increasing software maintainability. It comprises documentation updating, code optimization, and code restructuring. Documentation updating involves modifying the documents affected by the changes in order to correspond to the present state of the system. Code optimization involves modifying the programs for faster execution or efficient use of storage space. Code restructuring involves transforming the program structure for reducing the complexity in source code and making it easier to understand.

Preventive maintenance is limited to the maintenance organization only and no external requests are acquired for this type of maintenance. Preventive maintenance accounts for only 5% of all the maintenance activities.

**CHAPTER-7**

**IMPLEMENTATION**

**FrontPage.html(homepage)**



<html>

<head>

<title>JMD BLOOD BANK</title>

<link href="fp.css" rel="stylesheet" type="text/css">

<style>

body{

background-image:url("Pic5.jpg");

background-size:cover; }

</style>

</head>

<body>

<header>

<div class="row">

<ul class="main-nav">

<li class="active"><a href=""> HOME </a></li>

<li><a href="login.html"> LOGIN </a></li>

<li><a href="about us.html"> ABOUT </a></li>

<li><a href="contact us.html"> CONTACT </a></li>

<li><a href="news.html"> NEWS </a></li>

<li><a href="gallery.html"> GALLERY </a></li>

</ul>

</div>

</header>

</body>

</html>

**FP.CSS**

\*

{

margin: 0;

padding: 0;

}

body

{

font-family: monospace;

}

.row

{

max-width: 1200px;

margin: auto;

}

.main-nav

{

float: right;

list-style: none;

margin-top: 30px;

}

.main-nav li

{

display: inline-block;

}

.main-nav li.active a

{

border: 1px solid white;

}

.main-nav li a:hover

{

border: 1px solid white;

}

.main-nav li a

{

color: white;

text-decoration: none;

padding: 5px 20px;

font-family: "Roboto", sans-serif;

font-size: 15px;

}

.hero

{

position: absolute;

width: 1200px;

margin-left: 0px;

margin-top: 0px;

}

h1

{

color: white;

text-transform: uppercase;

font-size: 70px;

text-align: center;

margin-top: 275px;

}

header

{

background-image:linear-gradient(rgba(0,0,0,0.5),rgba(0,0,0,0.5)),url(pic1.jpg);

height: 100vh;

background-size: cover;

background-position: center;

}

.button

{

margin-top: 30px;

margin-left: 440px;

}

.btn

{

border: 1px solid white;

padding: 10px 30px;

color: white;

text-decoration: none;

margin-right: 5px;

font-size: 13px;

text-transform: uppercase;

}

.btn-one

{

background-color: darkorange;

font-family: "Roboto", sans-serif;

}

.btn-two

{

font-family: "Roboto", sans-serif;

}

.btn-two:hover

{

background-color: darkorange;

transition: all 0.5s ease-in;

}

.logo img

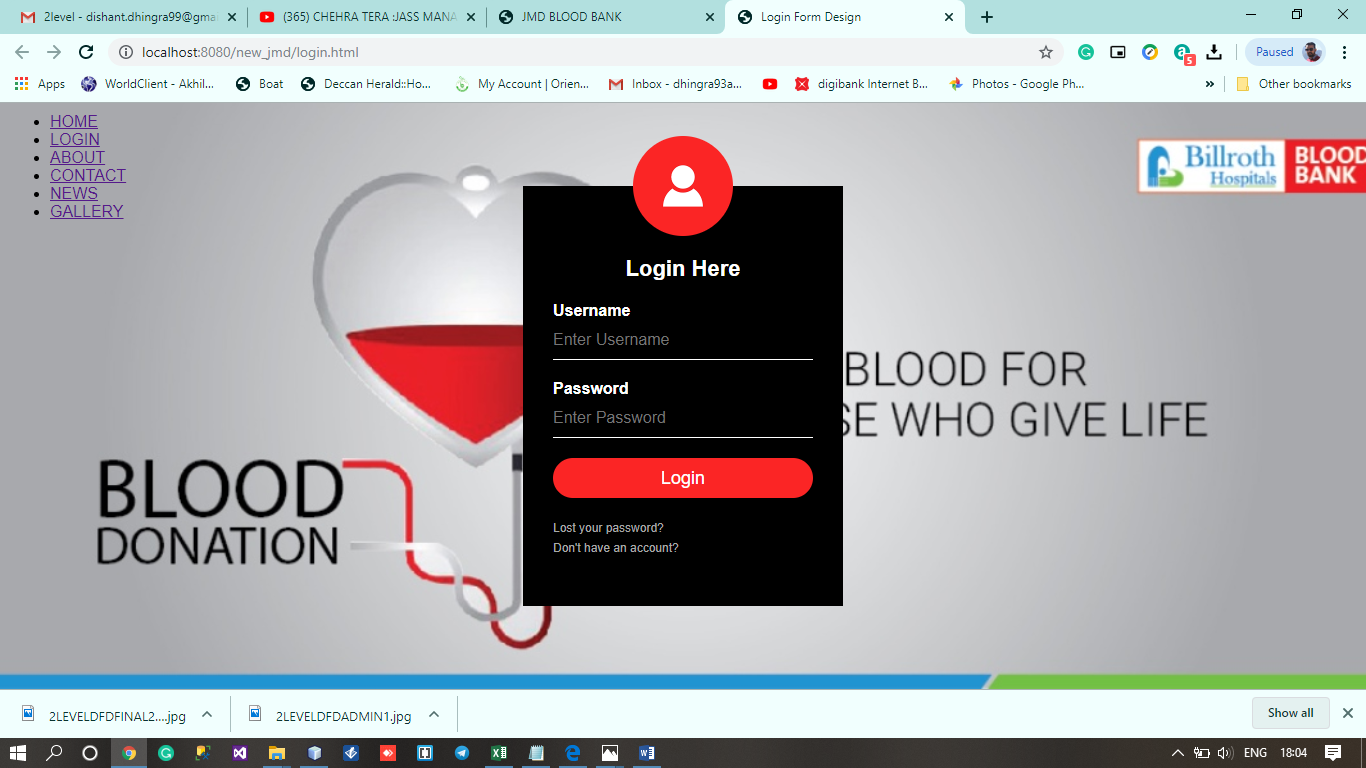
{

width: 150px;

height: auto;

}

**LOGIN.HTML**



<html>

<head>

<title>Login Form Design</title>

<link rel="stylesheet" type="text/css" href="loginstyle.css">

<body>

<head>

<div class="row">

<ul class="main-nav">

<li class="active"><a href="front page.html"> HOME </a></li>

<li><a href="login.html"> LOGIN </a></li>

<li><a href="about us.html"> ABOUT </a></li>

<li><a href="contact us.html"> CONTACT </a></li>

<li><a href=""> NEWS </a></li>

<li><a href="gallery.html"> GALLERY </a></li>

</ul>

</div>

</head>

<div class="loginbox">

<img src="avatar.png" class="avatar">

<h1>Login Here</h1>

<form method=get action="login.jsp">

<p>Username</p>

<input type="text" name="ID" placeholder="Enter Username">

<p>Password</p>

<input type="password" name="PASSWORD" placeholder="Enter Password">

<input type="submit" name="" value="Login">

<a href="#">Lost your password?</a><br>

<a href="signup.html">Don't have an account?</a>

</form>

</div>

</body>

</html>

**LOGIN.JSP**

<%--

Document : login

Created on : 25 Nov, 2019, 3:33:02 PM

Author : Akhil

--%>

<%@page import="java.sql.ResultSet"%>

<%@page import="java.sql.PreparedStatement"%>

<%@page import="java.sql.Connection"%>

<%@page import="java.sql.DriverManager"%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP Page</title>

</head>

<body>

<%

String un= request.getParameter("ID");

String p= request.getParameter("PASSWORD");

System.out.print(un);

System.out.print(p);

try

{

Class.forName("org.apache.derby.jdbc.ClientDriver");

Connection cn= DriverManager.getConnection("jdbc:derby://localhost:1527/mk");

PreparedStatement ps=cn.prepareStatement("select \* from LOGIN where ID=? and PASSWORD=?");

ps.setString(1,un);

ps.setString(2,p);

ResultSet rs= ps.executeQuery();

if(rs.next())

{

session.setAttribute("ID", un);

String x= rs.getString("USERTYPE");

if(x.equalsIgnoreCase("DONOR"))

{

response.sendRedirect("donorprofile.html");

}

else if(x.equals("ADMIN"))

{

response.sendRedirect("adminprofile.html");

}

}

else

{

response.sendRedirect("Login.html");

}

cn.close();

}

catch(Exception ex)

{

out.print(ex);

}

%>

</body>

</html>

**LOGINSTYLE.CSS**

body{

margin: 0;

padding: 10px;

background: url(Pic5.jpg);

background-size: cover;

background-position: center;

font-family: sans-serif;

}

.loginbox{

width: 320px;

height: 420px;

background: #000;

color: #fff;

top: 50%;

left: 50%;

position: absolute;

transform: translate(-50%,-50%);

box-sizing: border-box;

padding: 70px 30px;

}

.avatar{

width: 100px;

height: 100px;

border-radius: 50%;

position: absolute;

top: -50px;

left: calc(50% - 50px);

}

h1{

margin: 0;

padding: 0 0 20px;

text-align: center;

font-size: 22px;

}

.loginbox p{

margin: 0;

padding: 0;

font-weight: bold;

}

.loginbox input{

width: 100%;

margin-bottom: 20px;

}

.loginbox input[type="text"], input[type="password"]

{

border: none;

border-bottom: 1px solid #fff;

background: transparent;

outline: none;

height: 40px;

color: #fff;

font-size: 16px;

}

.loginbox input[type="submit"]

{

border: none;

outline: none;

height: 40px;

background: #fb2525;

color: #fff;

font-size: 18px;

border-radius: 20px;

}

.loginbox input[type="submit"]:hover

{

cursor: pointer;

background: #ffc107;

color: #000;

}

.loginbox a{

text-decoration: none;

font-size: 12px;

line-height: 20px;

color: darkgrey;

}

.loginbox a:hover

{

color: #ffc107;

}

/\*

To change this license header, choose License Headers in Project Properties.

To change this template file, choose Tools | Templates

and open the template in the editor.

\*/

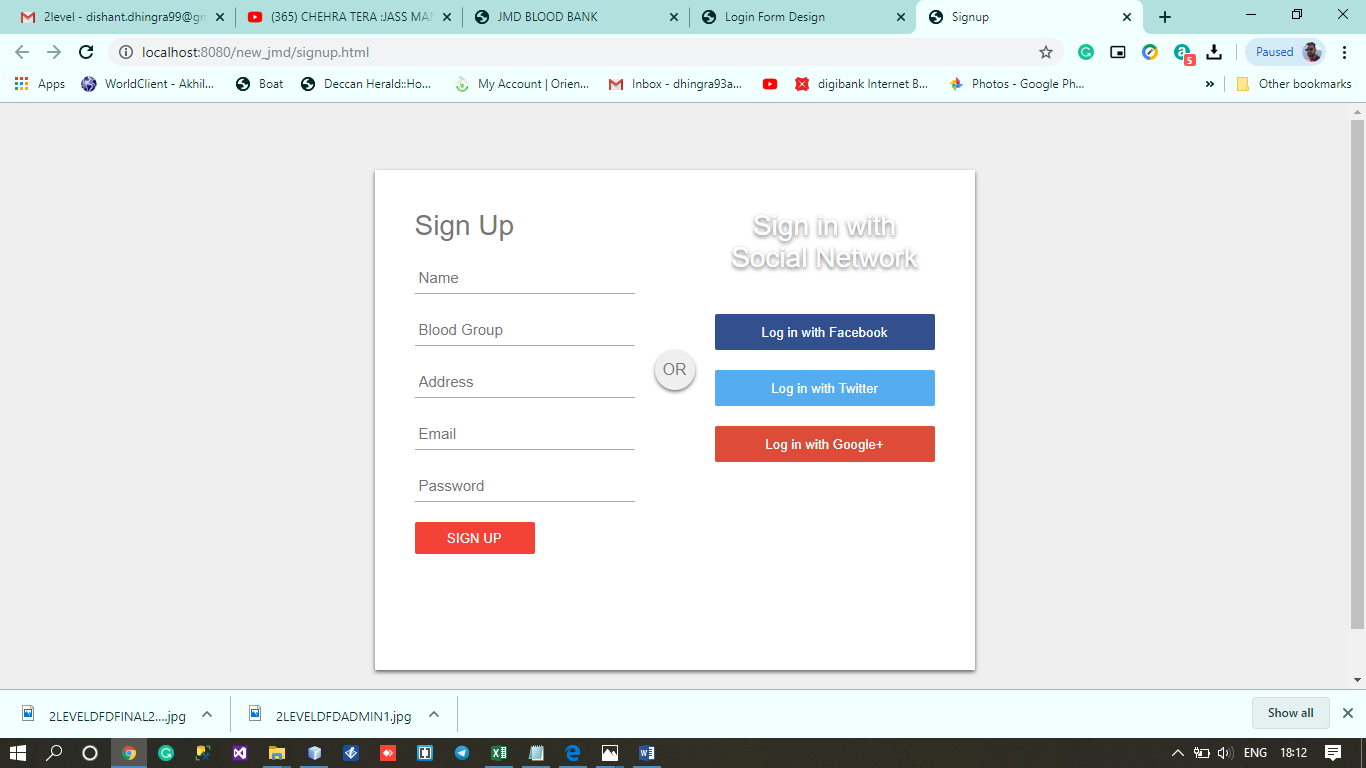
/\*

Created on : 18 Nov, 2019, 5:21:08 PM

Author : Akhil

\*/

**SIGNUP.HTML**



<html>

<head>

<title> Signup </title>

<link rel="stylesheet" type="text/css" href="style.css">

</head>

<body>

<form method=get action="signup.jsp">

<div id="login-box">

<div class="left-box">

<form method=get action="signup.jsp">

<h1> Sign Up</h1>

<input type="text" name="name" placeholder="Name"/>

<input type="text" name="bloodgroup" placeholder="Blood Group"/>

<input type="text" name="address" placeholder="Address"/>

<input type="text" name="email" placeholder="Email"/>

<input type="password" name="password" placeholder="Password"/>

<input type="submit" name="signup-button" value="Sign Up"/>

</form>

</div>

<div class="right-box">

<span class="signinwith">Sign in with<br/>Social Network </span>

<button class="social facebook">Log in with Facebook</button>

<button class="social twitter">Log in with Twitter</button>

<button class="social google">Log in with Google+</button>

</div>

<div class="OR">OR</div>

</div>

</body>

</html>

**SIGNUP.JSP**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP Page</title>

</head>

<%@page import="java.sql.\*"%>

<body>

<%

String nm= request.getParameter("name");

String add= request.getParameter("address");

String em= request.getParameter("email");

String p= request.getParameter("password");

//String mn= request.getParameter("phonenumber");

String bg=request.getParameter("bloodgroup");

//out.print(nm+ " "+ add + " " + em + " "+ p + " "+mn + " "+ bg + " ");

try

{

Class.forName("org.apache.derby.jdbc.ClientDriver");

Connection cn= DriverManager.getConnection("jdbc:derby://localhost:1527/mk");

PreparedStatement ps=cn.prepareStatement("select max(d\_id) from donor");

ResultSet rs= ps.executeQuery();

int x=1001;

if(rs.next())

{

if(rs.getString(1)!=null)

{

x=rs.getInt(1)+1;

}

}

session.setAttribute("d", x);

ps=cn.prepareStatement("insert into donor values(?,?,?,?,?)");

ps.setString(2, nm);

ps.setString(3, add);

ps.setString(1, em);

//ps.setInt(5, Integer.parseInt(mn));

ps.setString(4,bg);

ps.setInt(5, x);

ps.executeUpdate();

ps=cn.prepareStatement("insert into login values(?,?,'Donor')");

ps.setString(1, nm);

ps.setString(2, p);

ps.executeUpdate();

cn.close();

response.sendRedirect("frontpage.html");

}

catch(ClassNotFoundException ex)

{

out.print(ex);

}

catch(SQLException ex)

{

out.print(ex);

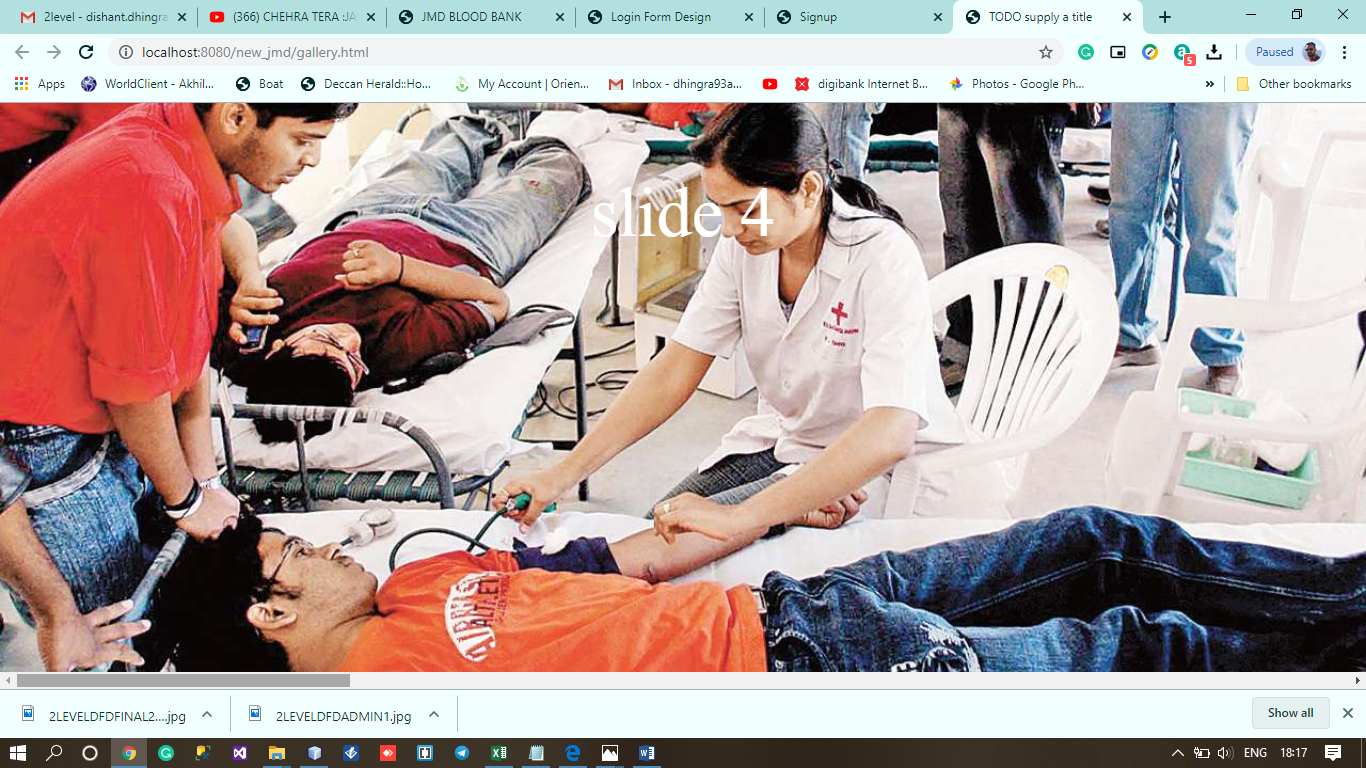
}

%>

</body>

</html>

**GALLERY.HTML**



<!DOCTYPE html>

<!--

To change this license header, choose License Headers in Project Properties.

To change this template file, choose Tools | Templates

and open the template in the editor.

-->

<html>

<head>

<title>TODO supply a title</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width">

<link rel="stylesheet" href="gallery.css">

</head>

<body>

<slider>

<slide><p>slide 1</p></slide>

<slide><p>slide 2</p></slide>

<slide><p>slide 3</p></slide>

<slide><p>slide 4</p></slide>

</slider>

</body>

</html>

**GALLERY.CSS**

html,body

{

width:200%;

height:100%;

padding:0;

margin:0px;

}

slider{

display : block;

width: 100%;

height: 100%;

background-color: #1f1f1f;

overflow: hidden;

position: absolute;

}

slider > \* {

position: absolute;

display: block;

width: 100%;

height: 100%;

background: #1f1f1f;

animation: slide 12s infinite;

overflow: hidden;

}

slide:nth-child(1)

{

left: 0%;

animation-delay: 1s;

background-image: url(1.jpg);

background-size: cover;

background-position:center;

}

slide:nth-child(2)

{

animation-delay: 2s;

background-image: url(2.jpg);

background-size: cover;

background-position:center;

}

slide:nth-child(3)

{

animation-delay: 5s;

background-image: url(3.jpg);

background-size: cover;

background-position:center;

}

slide:nth-child(4)

{

animation-delay: 8s;

background-image: url(4.jpg);

background-size: cover;

background-position:center;

}

slide p{

font\_family:comfortaa;

font-size:70px;

text-align:center;

display:inline-block;

width:100%;

margin\_top:340px;

color:#fff;

}

@keyframe slide

{

0%{left:100%;width:100%;}

5%{left:left:0%;}

25%{left:0%;}

30%{left:-100% width:100%;}

30.0001%{left:-100%; width:0%;}

100%{left:100%;width:0%;}

}

/\*

To change this license header, choose License Headers in Project Properties.

To change this template file, choose Tools | Templates

and open the template in the editor.

\*/

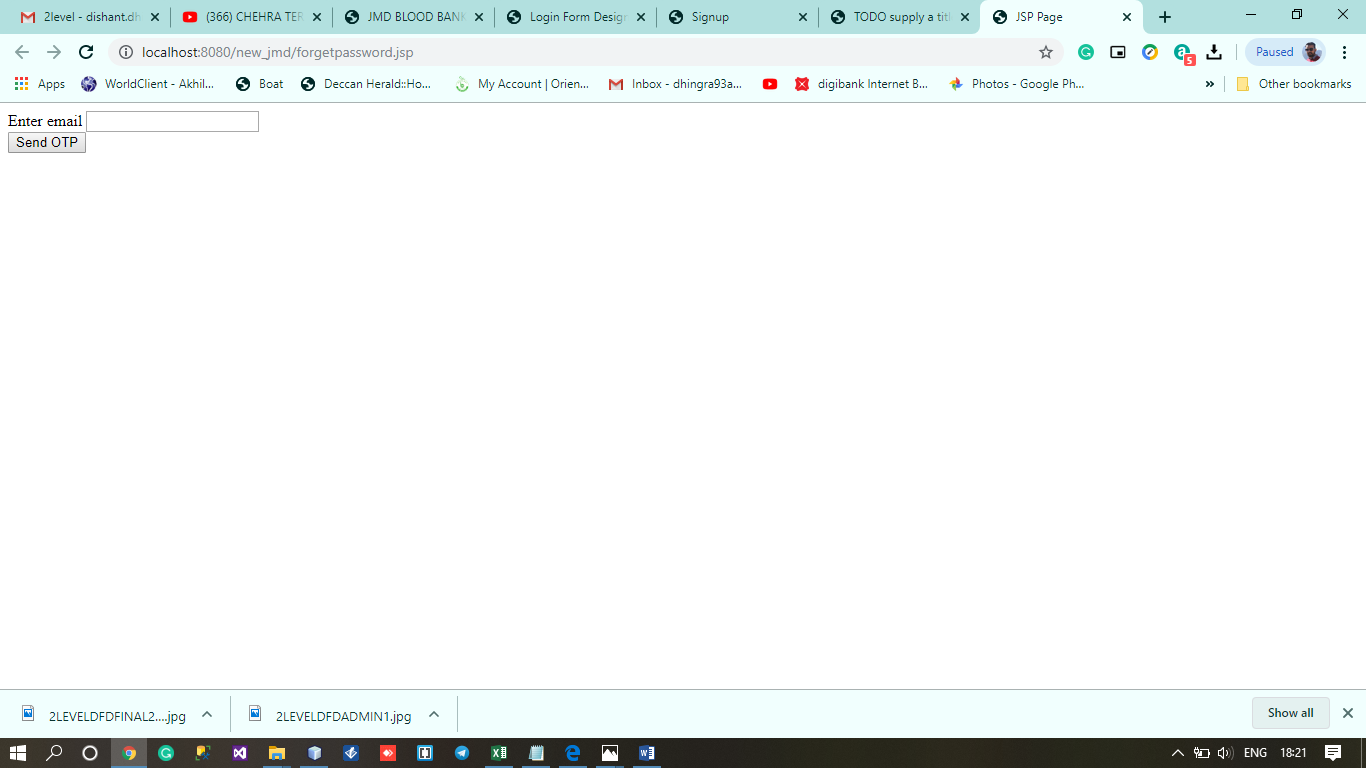
/\*

Created on : 25 Nov, 2019, 5:22:46 PM

Author : Akhil

\*/

**FORGOT PASSWORD**



<%--

Document : forgetpassword

Created on : Nov 22, 2019, 4:02:22 PM

Author : abhishek

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP Page</title>

</head>

<body>

<form action="forgot.jsp">

Enter email <input type="email" name="email"><br>

<input type="submit" value="Send OTP">

</form>

</body>

</html>

**FORGOT.JSP**

<%--

Document : forgot pass word

Created on : 18 Nov, 2019, 6:48:28 PM

Author : abhishek

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Booking</title>

<link href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">

<link href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css"

rel="stylesheet" integrity="sha384-wvfXpqpZZVQGK6TAh5PVlGOfQNHSoD2xbE+QkPxCAFlNEevoEH3Sl0sibVcOQVnN" crossorigin="anonymous">

<link href="https://fonts.googleapis.com/css?family=Montserrat&display=swap" rel="stylesheet">

<link rel="stylesheet" type="text/css" href="joinus.css">

<link rel="stylesheet" type="text/css" href="slider.css">

<link rel="stylesheet" type="text/css" href="CSS/payment.css">

<%@page import="java.sql.\*"%>

</head>

<style type="text/css">

.main-container{

display: flex;

flex-direction: row;

}

.main-container .sidebar-left{

flex: 1;

}

.main-container .sidebar-right{

flex: 5;

display: flex;

justify-content: center;

align-items: center;

}

body {

font-family: Arial, Helvetica, sans-serif;

}

/\* Style inputs \*/

input[type=text], select, textarea {

width: 100%;

padding: 12px;

border: 1px solid #ccc;

margin-top: 6px;

margin-bottom: 16px;

resize: vertical;

}

input[type=submit] {

background-color: #4CAF50;

color: white;

padding: 12px 20px;

border: none;

cursor: pointer;

}

input[type=submit]:hover {

background-color: #45a049;

}

/\* Style the container/contact section \*/

.container {

border-radius: 2px;

background-color: #f2f2f2;

padding: 0px;

}

/\* Create two columns that float next to eachother \*/

.column {

float: left;

width: 50%;

margin-top: 6px;

padding: 20px;

}

/\* Clear floats after the columns \*/

.row:after {

content: "";

display: table;

clear: both;

}

/\* Responsive layout - when the screen is less than 600px wide, make the two columns stack on top of each other instead of next to each other \*/

@media screen and (max-width: 100%) {

.column, input[type=submit] {

width: 100%;

margin-top: 0;

}

}

</style>

<body>

<div class="main-container">

<!-- Left Sidebar for Coolapsing navbar -->

<div class="sidebar-left">

<nav class="menu">

<header>Menu <span>×</span></header>

<ol>

<li class="menu-item"><a href="home.html">Home</a></li>

<li class="menu-item">

<a href="joinus.html ">Join us</a>

<ol class="sub-menu">

<li class="menu-item"><a href="joinus.html">Login</a></li>

<li class="menu-item"><a href="joinus.html">Sign-up</a></li>

</ol>

</li>

<li class="menu-item">

<a href="#0">Booking</a>

<ol class="sub-menu">

<li class="menu-item"><a href="hotel.jsp">Hotel</a></li>

<li class="menu-item"><a href="package.jsp">Package</a></li>

</ol>

</li>

<li class="menu-item">

<a href="#0">Cancel</a>

<ol class="sub-menu">

<li class="menu-item"><a href="cancelHotel.jsp">Hotel</a></li>

<li class="menu-item"><a href="cancelpackage.jsp">Package</a></li>

</ol>

</li>

<li class="menu-item"><a href="aboutus.html">About us</a></li>

<li class="menu-item"><a href="contactus.html">Contact us</a></li>

<li class="menu-item"><a href="gallery.html">Gallery</a></li>

</ol>

<footer><button aria-label="Toggle Menu">Toggle</button></footer>

</nav>

</div>

<!-- Right sidebar for content -->

<%

String usnm=request.getParameter("ID");

try

{

String im="";

Class.forName("org.apache.derby.jdbc.ClientDriver");

Connection cn= DriverManager.getConnection("jdbc:derby://localhost:1527/mk");

PreparedStatement ps= cn.prepareStatement("select \* from LOGIN where ID=?");

ps.setString(1,usnm);

ResultSet rs=ps.executeQuery();

if(rs.next())

{

session.setAttribute("uem", usnm);

response.sendRedirect("sendmail.jsp");

}

else

{

response.sendRedirect("forgetpassword.jsp");

}

}

catch(Exception ex) {

out.print(ex);

}

%>

<script>

function myFunction() {

var x = document.getElementById("myDate").value;

document.getElementById("demo").innerHTML = x;

}

</script>

<script src="https://code.jquery.com/jquery-3.3.1.js"></script>

<script type="text/javascript" src="collapsenav.js"></script>

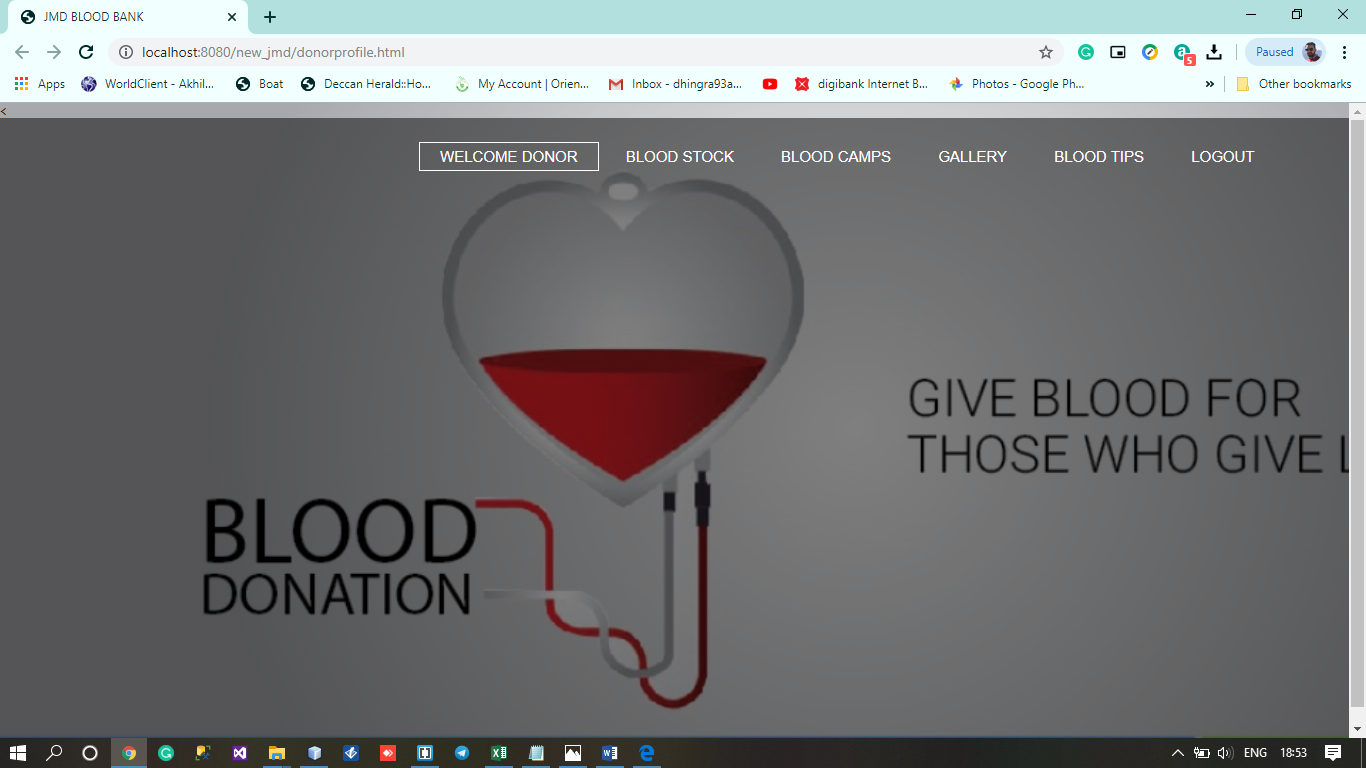
<script type="text/javascript" src="slider.js"></script>

</body>

</body>

</html>

**DONORPROFILE.HTML**



<!DOCTYPE html>

<!--

To change this license header, choose License Headers in Project Properties.

To change this template file, choose Tools | Templates

and open the template in the editor.

-->

< <html>

<head>

<title>JMD BLOOD BANK</title>

<link href="fp.css" rel="stylesheet" type="text/css">

<style>

body{

background-image:url("Pic5.jpg");

background-size:cover; }

</style>

</head>

<body>

<header>

<div class="row">

<ul class="main-nav">

<li class="active"><a href="donorprofile.html"> WELCOME DONOR</a></li>

<li><a href="bloodstock.html">BLOOD STOCK </a></li>

<li><a href="bloodcamps.html"> BLOOD CAMPS </a></li>

<li><a href="gallery.html"> GALLERY </a></li>

</ul>

</div>

</header>

</body>

</html>

**LOGOUT.JSP**

<%--

Document : logout

Created on : 3 Jul, 2019, 12:51:53 AM

Author : ABHIJEET

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP Page</title>

</head>

<body>

<%

session.invalidate();

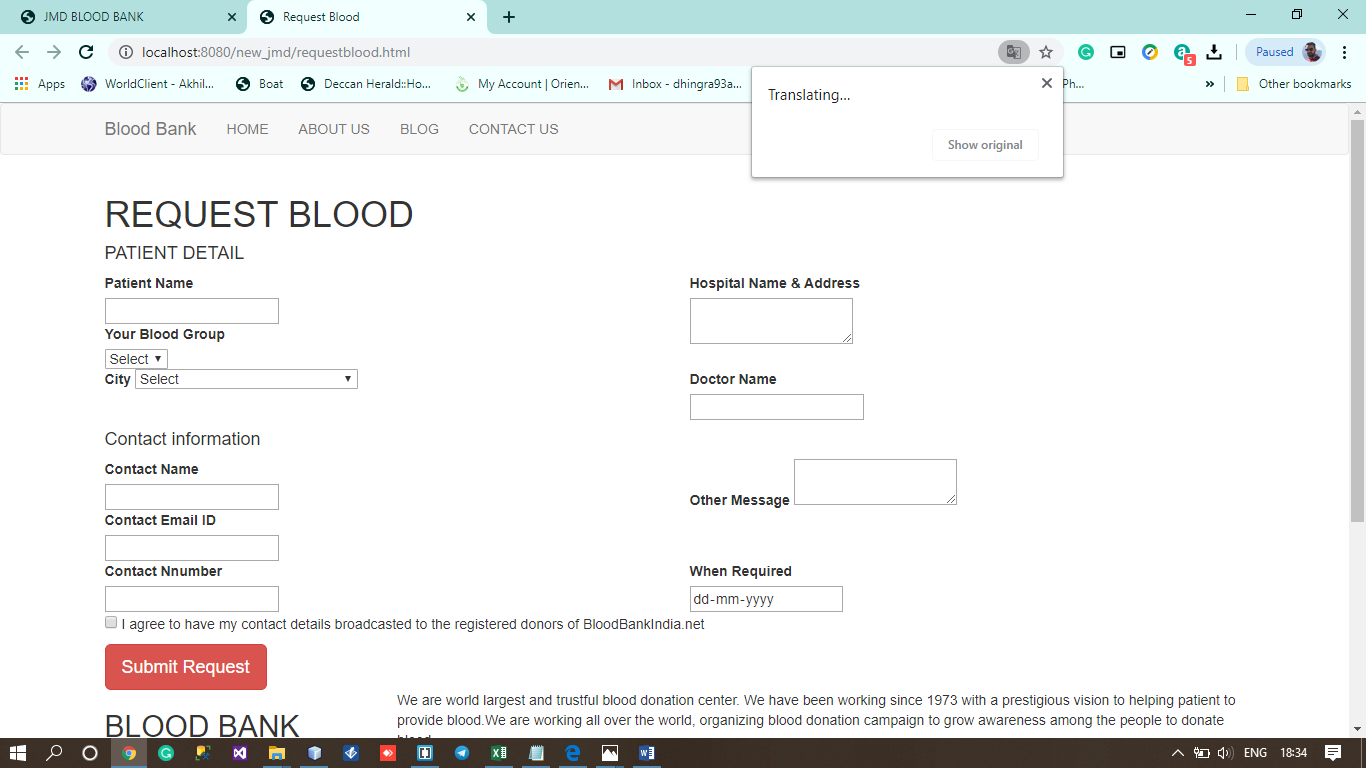
%>

<jsp:forward page="frontpage.html"/>

</body>

</html>

**REQUESTBLOOD.HTML**



<!DOCTYPE html>

<html>

<head>

<title>Request Blood</title>

<link rel="stylesheet" type="text/css" href="bootstrap.css">

<link rel="stylesheet" type="text/css" href="Request\_Blood.css">

</head>

<body>

<div class="navbar navbar-default">

<div class="container">

<div class="navbar-header">

<a href="#" class="navbar-brand">Blood Bank</a>

</div>

<ul class="nav navbar-nav">

<li><a href="frontnew.html">HOME</a></li>

<li><a href="about.html">ABOUT US</a></li>

<li><a href="#">BLOG</a></li>

<li><a href="contactus.html">CONTACT US</a></li>

</ul>

</div>

</div>

<div>

<div class="container signup">

<h1 class="h1-one">REQUEST BLOOD</h1>

<form class="form-one">

<h4 class="h4-one">PATIENT DETAIL</h4>

<div class="row">

<div class="col-lg-6">

<div class="row">

<div class="col-lg-12">

<label class="label-one" for="pName">Patient Name</label>

<br>

<input class="input-one" id="pname" type="text">

<br>

</div>

</div>

<div class="row">

<div class="col-lg-12">

<label class="label-one" for="blood">Your Blood Group</label>

<br>

<select id="blood" class="select-one">

<option>Select</option>

<option>A+</option>

<option>B+</option>

<option>O+</option>

<option>A-</option>

<option>B-</option>

<option>O-</option>

<option>AB+</option>

<option>AB-</option>

<option>A</option>

</select>

</div>

</div>

</div>

<div class="col-lg-6">

<label class="label-one" for="Hospital">Hospital Name & Address</label>

<br>

<textarea col=10 row=5 class="textarea-one"></textarea>

</div>

</div>

<div class="row">

<div class="col-lg-6">

<label for="city" class="label-one">City</label>

<select id="city" class="select-one">

<option value="0">Select </option>

<option value="1">Agartala</option>

<option value="2">Agra</option>

<option value="3">Ahmedabad</option>

<option value="4">Ahmednagar</option>

<option value="5">Aizwal</option>

<option value="6">Ajmer</option>

<option value="7">Akola</option>

<option value="8">Aligarh</option>

<option value="9">Allahabad</option>

<option value="10">Alwar</option>

<option value="11">Ambala</option>

<option value="12">Amravati</option>

<option value="13">Amreli</option>

<option value="14">Amritsar</option>

<option value="15">Anand</option>

<option value="16">Anantapur</option>

<option value="17">Anklesvar</option>

<option value="18">Anuppur</option>

<option value="19">Araria</option>

<option value="20">Arcot</option>

<option value="21">Arrah</option>

<option value="22">Aruppukkottai</option>

<option value="23">Asansol</option>

<option value="24">Ashok Nagar</option>

<option value="25">Aurangabad, Bihar</option>

<option value="26">Aurangabad, Maharashtra</option>

<option value="27">Azamgarh</option>

<option value="28">Bahadurgarh</option>

<option value="29">Baharampur</option>

<option value="30">Bahraich</option>

<option value="31">Balaghat</option>

<option value="32">Balangir Orissa</option>

<option value="33">Balasore</option>

<option value="34">Balia</option>

<option value="35">Ballabhgarh</option>

<option value="36">Ballarpur</option>

<option value="37">Balrampur</option>

<option value="38">Balurghat West Bengal</option>

<option value="39">Banda</option>

<option value="40">Bangalore</option>

<option value="41">Banganapalle</option>

<option value="42">Banswara</option>

<option value="43">Banswara</option>

<option value="44">Banur</option>

<option value="45">baran</option>

<option value="46">Bardhaman</option>

<option value="47">Bareilly</option>

<option value="48">Barh</option>

<option value="49">Baripada</option>

<option value="50">Barmer</option>

<option value="51">Barrackpur</option>

<option value="52">Barwani</option>

<option value="53">Beawar</option>

<option value="54">Belgaum</option>

<option value="55">Bellary</option>

<option value="56">Bengaluru</option>

<option value="57">Betul</option>

<option value="58">Bhagalpur</option>

<option value="59">Bhandara</option>

<option value="60">Bharatpur</option>

<option value="61">Bharuch</option>

<option value="62">Bhavani</option>

<option value="63">Bhavnagar</option>

<option value="64">Bhilai Nagar</option>

<option value="65">Bhimavaram</option>

<option value="66"Bhiwandi</option>

<option value="67">Bhopal</option>

<option value="68">Bhubaneswar</option>

<option value="69">Bhuj</option>

<option value="70">Bidar</option>

<option value="71">Bihar Sharif</option>

<option value="72">Bijnaur, UP</option>

<option value="73">Bikaner</option>

<option value="74">Bilaspur, Chhattisgarh</option>

<option value="75">Bilaspur, Himachal Pradesh</option>

<option value="76">Bilgha, Punjab</option>

<option value="77">Bodh Gaya</option>

<option value="78">Bokaro Steel City</option>

<option value="79">Bongaigaon</option>

<option value="80">Bulandshahr</option>

<option value="81">Buldana</option>

<option value="82">Burhanpur</option>

<option value="83">Buxar</option>

<option value="84">Cambay</option>

<option value="85">Caryobys</option>

<option value="86">Chamoli Gopeshwar</option>

<option value="87">Champawat</option>

<option value="88">Chamrajnagar</option>

<option value="89">Chandannagar</option>

<option value="90">Chandigarh</option>

<option value="91">Chandrapur</option>

<option value="92">Chapra</option>

<option value="93">Charkhari</option>

<option value="94">Chengalpattu</option>

<option value="95">Chennai</option>

<option value="96">Chhatarpur</option>

<option value="97">Chhindwara</option>

<option value="98">Chikmagalur</option>

<option value="99">Chiplun</option>

<option value="100">Chitradurga</option>

<option value="101">Chitrakoot Dham Karwi</option>

<option value="102">Chittoor</option>

<option value="103">Coimbatore</option>

<option value="104">Contai</option>

<option value="105">Coonoor</option>

<option value="106">Cuddalore</option>

<option value="107">Cuddapah</option>

<option value="108">Cuttack</option>

<option value="109">Dabra</option>

<option value="110">Dadra</option>

<option value="111">Dahod</option>

<option value="112">Daksh</option>

<option value="113">Daltonganj</option>

<option value="114">Daman</option>

<option value="115">Damoh</option>

<option value="116">Darbhanga</option>

<option value="117">Darjeeling</option>

<option value="118">Datia</option>

<option value="119">Davanagere</option>

<option value="120">Dehgam</option>

<option value="121">Dehradun</option>

<option value="122">Delhi</option>

<option value="123">Deoghar</option>

<option value="124">Dewas</option>

<option value="125">Dhanbad</option>

<option value="126">Dhar</option>

<option value="127">Dharampur</option>

<option value="128">Dharamsala</option>

<option value="129">Dharwad</option>

<option value="130">Dholka</option>

<option value="131">Dhule</option>

<option value="132">Dhulian</option>

<option value="133">Dibrugarh</option>

<option value="134">Dindigul</option>

<option value="135">Dispur</option>

<option value="136">Diu</option>

<option value="137">Diu</option>

<option value="138">Dombivli</option>

<option value="139">Duhbai</option>

<option value="140">Dumdum</option>

<option value="141">Durg</option>

<option value="142">Durgapur</option>

<option value="143">Dwarka</option>

<option value="144">Ernakulam</option>

<option value="145">Erode</option>

<option value="146">Etah</option>

<option value="147">Etawah</option>

<option value="148">Faizabad</option>

<option value="149">Faridabad</option>

<option value="150">Faridkot</option>

<option value="151">Farrukhabad</option> <option value="152">Fatehgarh</option> <option value="153">Fatehpur Sikri</option> <option value="154">Ferozepur Cantt.</option> <option value="155">Ferozepur City</option> <option value="156">Firozabad</option> <option value="157">Gadchiroli</option> <option value="158">Gandhinagar</option> <option value="159">Gangtok</option> <option value="160">Ganjam</option> <option value="161">Garcha, Punjab</option> <option value="162">Gaya</option> <option value="163">Ghaziabad</option> <option value="164">Ghazipur</option> <option value="165">Goa Velha</option> <option value="166">Godhra</option> <option value="167">Gondiya</option> <option value="168">Gorakhpur</option> <option value="169">Greater Noida</option> <option value="170">Gudalur</option> <option value="171">Gudivada</option> <option value="172">Gulbarga</option> <option value="173">Gumla</option> <option value="174">Guna</option> <option value="175">Gundlupet</option> <option value="176">Guntur</option> <option value="177">Gurgaon</option> <option value="178">Guruharsahai</option> <option value="179">Guwahati</option> <option value="180">Gwalior</option> <option value="181">Gwalior</option> <option value="182">Haldia</option> <option value="183">Haldwani</option> <option value="184">Hamir, Uttar Pradesh</option> <option value="185">Hamirpur, Himachal Pradesh</option> <option value="186">Hansi</option> <option value="187">Hanumangarh</option> <option value="188">Harda</option> <option value="189">Haridwar</option> <option value="190">Harsawa</option> <option value="191">Hassan</option> <option value="192">Hastinapur</option> <option value="193">Hathras</option> <option value="194">Hatigudda, Karnataka</option> <option value="195">Himatnagar</option> <option value="196">hindupur</option> <option value="197">Hisar</option> <option value="198">Hoshiarpur</option> <option value="199">Howrah</option> <option value="200">Hubli</option> <option value="201">Hyderabad, Andhra Pradesh</option> <option value="202">Indore</option> <option value="203">Irinjalakuda</option> <option value="204">Jabalpur</option> <option value="205">Jagraon</option> <option value="206">Jagtial</option> <option value="207">Jaipur</option> <option value="208">Jais</option> <option value="209">Jaisalmer</option> <option value="210">Jalalabad</option> <option value="211">Jalandhar</option> <option value="212">Jalgaon</option> <option value="213">Jammu</option> <option value="214">Jamnagar</option> <option value="215">Jamshedpur</option> <option value="216">Jaunpur</option> <option value="217">Jhabua</option> <option value="218">Jhalawar</option> <option value="219">Jhansi</option> <option value="220">Jhunjhunu</option> <option value="221">Jodhpur</option> <option value="222">Jorhat</option> <option value="223">Junagadh</option> <option value="224">Kakinada</option> <option value="225">Kalimpong</option> <option value="226">Kalwa</option> <option value="227">Kalyan-Dombivali</option> <option value="228">Kalyani</option> <option value="229">Kanauj</option> <option value="230">Kancheepuram</option> <option value="231">Kandla</option> <option value="232">Kangazha</option> <option value="233">Kannur</option> <option value="234">Kanpur</option> <option value="235">Kanyakumari</option> <option value="236">Karaikal</option> <option value="237">Karaikudi</option> <option value="238">Karamsad</option> <option value="239">Karimnagar</option> <option value="240">Karjat</option> <option value="241">Karnal</option> <option value="242">Karur</option> <option value="243">Karwar</option> <option value="244">Kavaratti</option> <option value="245">Khamanon</option> <option value="246">Khammam</option> <option value="247">Khandwa</option> <option value="248">Khanna</option> <option value="249">Kharagpur</option> <option value="250">Khargone</option> <option value="251">Kheda</option> <option value="252">Khilchipur</option> <option value="253">Khopoli</option> <option value="254">Khuldabad</option> <option value="255">Kirandul</option> <option value="256">Kochi</option> <option value="257">Kohima</option> <option value="258">Kokrajhar</option> <option value="259">Kolar</option> <option value="260">Kolhapur</option> <option value="261">Kolkata</option> <option value="262">Kollam</option> <option value="263">Konark</option> <option value="264">Korba</option> <option value="265">Kota</option> <option value="266">Kotdwara</option> <option value="267">Kothagudem</option> <option value="268">Kothamangalam</option> <option value="269">Kottarakara</option> <option value="270">Kottayam</option> <option value="271">Kovilpatti</option> <option value="272">Koyampattur</option> <option value="273">Kozhencherry</option> <option value="274">Kozhikode</option> <option value="275">Krishnagiri</option> <option value="276">Kulpahar</option> <option value="277">Kumbakonam</option> <option value="278">Kumbhraj</option> <option value="279">Kurnool</option> <option value="280">kurukshetra]</option> <option value="281">Kushinagar</option> <option value="282">Lalitpur</option> <option value="283">Lamka</option> <option value="284">Leh</option> <option value="285">Leh</option> <option value="286">Lucknow</option> <option value="287">Ludhiana</option> <option value="288">Machilipatnam</option> <option value="289">Madanapalle</option> <option value="290">Madgaon</option> <option value="291">Madikeri</option> <option value="292">Madurai</option> <option value="293">Mahabaleswar</option> <option value="294">Mahabubnagar</option> <option value="295">Mahad</option> <option value="296">Mahe</option> <option value="297">Mahoba</option> <option value="298">Mahwa</option> <option value="299">Malout</option> <option value="300">Mandsaur</option> <option value="301">Mangalagiri</option> <option value="302">Mangalore</option> <option value="303">Mapusa</option> <option value="304">Marmagao</option> <option value="305">Mathura</option> <option value="306">Meerut</option> <option value="307">Melattur</option> <option value="308">Mirzapur</option> <option value="309">Moga</option> <option value="310">Mohali</option> <option value="311">Mokama</option> <option value="312">Moradabad</option> <option value="313">Motihari</option> <option value="314">Mount Abu</option> <option value="315">Mukatsar</option> <option value="316">Mullanpur</option> <option value="317">Mumbai</option> <option value="318">Murshidabad</option> <option value="319">Murwara</option> <option value="320">Mussoorie</option> <option value="321">Muzaffarnagar</option> <option value="322">Muzaffarpur</option> <option value="323">Mysore</option> <option value="324">Nadiad</option> <option value="325">Nagapattinam</option> <option value="326">Nagarkurnool</option> <option value="327">Nagercoil</option> <option value="328">Nagina, UP</option> <option value="329">Nagpur</option> <option value="330">Nainital</option> <option value="331">Nalgonda</option> <option value="332">Nanded</option> <option value="333">Nandurbar</option> <option value="334">Nandyal</option> <option value="335">Nanital</option> <option value="336">Narasaraopet</option> <option value="337">Narnol</option> <option value="338">Narsimhapur</option> <option value="339">Narsinghgarh</option> <option value="340">Nashik</option> <option value="341">Navi Mumbai</option> <option value="342">Navsari</option> <option value="343">Nawalgarh</option> <option value="344">Neemuch</option> <option value="345">Nellore</option> <option value="346">New Delhi\* or Delhi</option> <option value="347">New Guntur</option> <option value="348">Nizamabad</option> <option value="349">NOIDA</option> <option value="350">Nurmahal, Punjab</option> <option value="351">Nurpur, Himachal Pradesh</option> <option value="352">Palghat</option> <option value="353">Palwal</option> <option value="354">Panaji</option> <option value="355">Panchkula</option> <option value="356">Pandharpur</option> <option value="357">Panipat</option> <option value="358">Panna</option> <option value="359">Panvel</option> <option value="360">Pasla, Punjab</option> <option value="361">Patan</option> <option value="362">Pathankot</option> <option value="363">Patiala</option> <option value="364">Patna</option> <option value="365">Patratu,Jharkhand</option> <option value="366">Pimpri Chinchwad</option> <option value="367">Ponda</option> <option value="368">Porbandar</option> <option value="369">Port Blair</option> <option value="370">Pratapgarh</option> <option value="371">Puducherry</option> <option value="372">Punalur</option> <option value="373">Pune</option> <option value="374">Puri</option> <option value="375">Pushkar</option> <option value="376">Quilon</option> <option value="377">Rae Bareli</option> <option value="378">Raichur</option> <option value="379">Raigarh</option> <option value="380">Raipur</option> <option value="381">Raisen</option> <option value="382">Rajahmundry</option> <option value="383">Rajampet</option> <option value="384">Rajgarh</option> <option value="385">Rajkot</option> <option value="386">Rajnandgaon</option> <option value="387">Rajsthan</option> <option value="388">Ramanathapuram</option> <option value="389">Rameshwaram</option> <option value="390">Rampur</option> <option value="391">Ranchi</option> <option value="392">Rangapara</option> <option value="393">Ranikhet</option> <option value="394">Rasheed</option> <option value="395">Ratangarh</option> <option value="396">Ratlam</option> <option value="397">Ratnagiri</option> <option value="398">Raurkela</option> <option value="399">Ravulapalem</option> <option value="400">Rewa</option> <option value="401">Ringawa</option> <option value="402">Rishikesh</option> <option value="403">Roorkee</option> <option value="404">Sagars</option> <option value="405">Saharanpur</option> <option value="406">Salem</option> <option value="407">Samastipur</option> <option value="408">Sanawad</option> <option value="409">Sangamner</option> <option value="410">Sangli</option> <option value="411">Satara</option> <option value="412">Sathyamangalam</option> <option value="413">Satna</option> <option value="414">Sehore</option> <option value="415">Seoni</option> <option value="416">Shajapur</option> <option value="417">Shegaon</option> <option value="418">Sheopur</option> <option value="419">Shevgaon</option> <option value="420">Shillong</option> <option value="421">Shimla</option> <option value="422">Shimoga</option> <option value="423">Shirala</option> <option value="424">Shivani</option> <option value="425">Shivpuri</option> <option value="426">Shoaib</option> <option value="427">Sholapur</option> <option value="428">shrigonda</option> <option value="429">shrirampur</option> <option value="430">Siddipet</option> <option value="431">Sikar</option> <option value="432">Silchar</option> <option value="433">Siliguri</option> <option value="434">Silvassa</option> <option value="435">Sindhanur</option> <option value="436">Singrauli</option> <option value="437">Sirohi</option> <option value="438">Sironj</option> <option value="439">Sitamarhi</option> <option value="440">Sitapur</option> <option value="441">Siwan</option> <option value="442">Sonipat</option> <option value="443">Sriganganagar</option> <option value="444">Srikakulam</option> <option value="445">Srinagar</option> <option value="446">Surat</option> <option value="447">Suratgarh</option> <option value="448">Surendranagar</option> <option value="449">Tamluk</option> <option value="450">Tandur</option> <option value="451">Tenali</option> <option value="452">Thane</option> <option value="453">Thanjavur</option> <option value="454">Thathawata</option> <option value="455">Thiruvallur</option> <option value="456">Thiruvananthapuram</option> <option value="457">Thoothukudi,</option> <option value="458">Thrikkannamangal</option> <option value="459">Thrissur</option> <option value="460">Tikamgarh</option> <option value="461">Tinsukia</option> <option value="462">Tiruchirappalli</option> <option value="463">Tirunelveli</option> <option value="464">Tirupathi</option> <option value="465">Tirupattur</option> <option value="466">Tirupur</option> <option value="467">Tiruvarur</option> <option value="468">Tzudikong</option> <option value="469">Udaipur in Rajasthan</option> <option value="470">Udaipur in Tripura</option> <option value="471">Udhagamandalam</option> <option value="472">Udhampur in Jammu & Kashmir</option> <option value="473">Udupi</option> <option value="474">Ujjain</option> <option value="475">Ulhasnagar</option> <option value="476">Unnao</option> <option value="477">Uttarpara in West Bengal</option> <option value="478">Vadodara</option> <option value="479">Vallabh Vidhyanagar</option> <option value="480">Valsad</option> <option value="481">Vandavasi</option> <option value="482">Vapi</option> <option value="483">Varanasi</option> <option value="484">Vasai</option> <option value="485">Vasco da Gama, Goa</option> <option value="486">Vellore</option> <option value="487">Vidisha</option> <option value="488">Vijayawada</option> <option value="489">Viluppuram</option> <option value="490">Virar</option> <option value="491">Virudhachalam</option> <option value="492">Visakhapatnam</option> <option value="493">Vizianagaram</option> <option value="494">Vyara</option> <option value="495">Wani</option> <option value="496">Warangal</option> <option value="497">Wardha</option> <option value="498">Wayanad</option>

</select>

</div>

<div class="col-lg-6">

<label class="label-one" for="doctor">

Doctor Name

</label>

<br>

<input class="input-one" id="doctor" type="text">

</div>

</div>

</form>

<form class="form-one">

<h4 class="h4-one">Contact information</h4>

<div class="row">

<div class="col-lg-6">

<div class="row">

<div class="col-lg-12">

<label class="label-one" for="ContactName">Contact Name</label>

<br>

<input class="input-one" id="ContactName" type="text">

</div>

<div class="col-lg-12">

<label class="label-one" for="ContactEmail">Contact Email ID</label>

<br>

<input class="input-one" id="ContactEmail" type="text">

</div>

</div>

</div>

<div class="col-lg-6">

<label for="Other" class="label-one">Other Message</label>

<textarea col="10" row="10" class="textarea-one"></textarea>

</div>

</div>

<div class="row">

<div class="col-lg-6">

<label class="label-one" for="ContactNo">Contact Nnumber</label>

<br>

<input type="number" class="input-one" id="ContactNo">

</div>

<div class="col-lg-6">

<label class="label-one" for="date">When Required</label>

<br>

<input type="date" class="input-one" id="date">

</div>

</div>

</form>

<div style="width:90%;">

<p><input type="checkbox" required> I agree to have my contact details broadcasted to the registered donors of BloodBankIndia.net

</p>

</div>

<div class="submit">

<button class="btn btn-danger btn-lg" id="form-btn">Submit Request</button>

</div>

</div>

</div>

<footer>

<div class="container">

<div class="row">

<div class="col-lg-3 footer-h2">

<h2>BLOOD BANK</h2>

</div>

<div class="col-lg-9 footer-p1" >

<p>

We are world largest and trustful blood donation center. We have been working since 1973 with a prestigious vision to helping patient to provide blood.We are working all over the world, organizing blood donation campaign to grow awareness among the people to donate blood.

</p>

</div>

</div>

<hr>

<div class="col-lg-4">

<h4 class="footer-h4">SUBSCRIBE US</h4>

<p class="footer-p1" style="padding:10px 0;">Signup for regular newsletter and stay up to date with our lastest news.</p>

<input type="email" placeholder="Enter Your Email" class="footer-input">

<br>

<button class="btn btn-danger btn-lg btn-footer"><a href="Blood%20Donation.docx">JOIN NEWSLETTER</a></button>

</div>

<div class="col-lg-4">

</div>

<div class="col-lg-4">

</div>

</div>

<hr>

<div class="container">

<div class="row">

<div class="col-lg-6">

<p class="footer-p1">Copyright 2018-Blood Bank by GreenLight.All Rights Reserved.</p>

</div>

<div class="col-lg-6 footer-a-all">

<a href="#" class="footer-a1">Cookies</a>

<a href="#" class="footer-a1">Sitemap</a>

<a href="#" class="footer-a1">Feedback</a>

<a href="#" class="footer-a1 footer-a1-l">Privacy Policy</a>

</div>

</div>

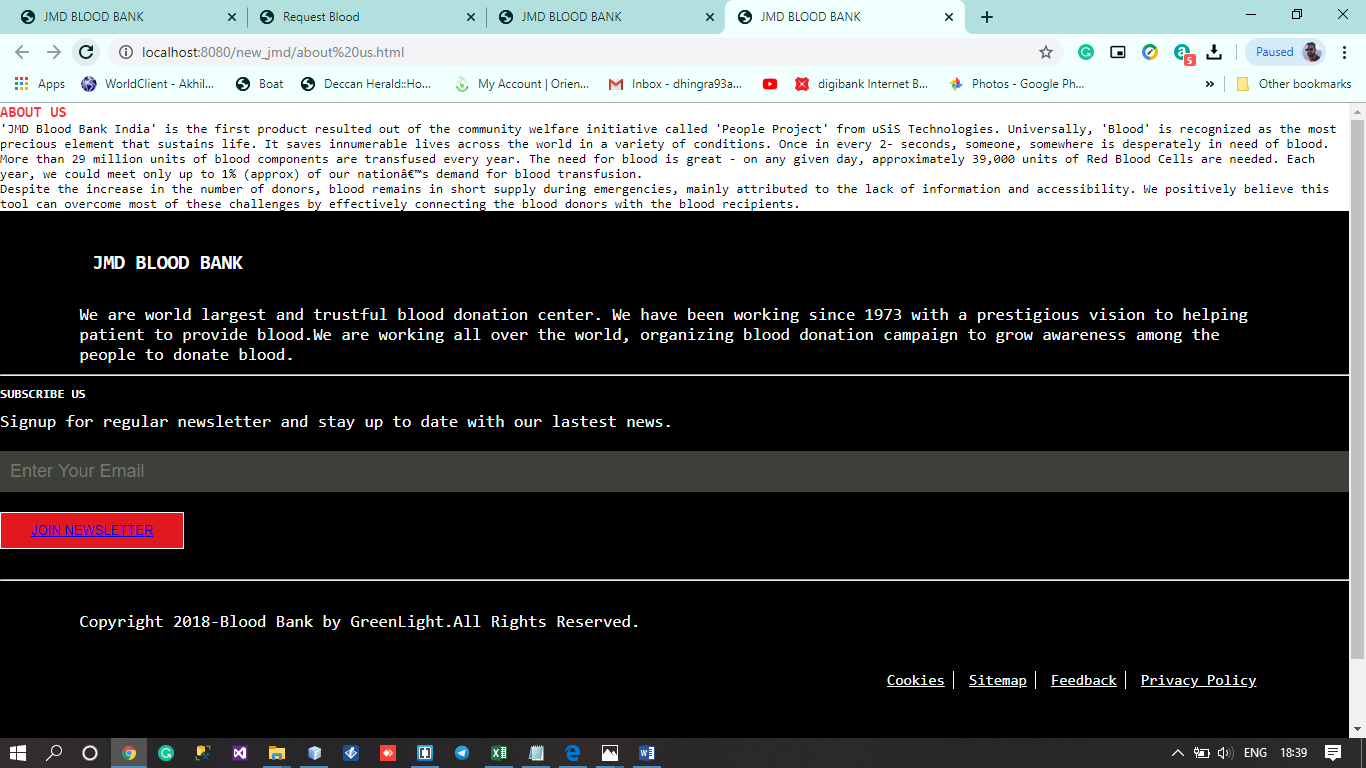
</div>

</footer>

</body>

</html>

**ABOUTUS.HTML**



<html>

<head>

<title>JMD BLOOD BANK</title>

<link href="fp.css" rel="stylesheet" type="text/css">

<link href="aboutus.css" rel="stylesheet" type="text/css">

<!--

<div id="h-image">

</div> -->

<div class="main">

<div class="container">

<div class="col-lg-6">

<h3 class="main-h3">ABOUT US</h3>

<p class="main-p">'JMD Blood Bank India' is the first product resulted out of the community welfare initiative called 'People Project' from uSiS Technologies. Universally, 'Blood' is recognized as the most precious element that sustains life. It saves innumerable lives across the world in a variety of conditions. Once in every 2- seconds, someone, somewhere is desperately in need of blood. More than 29 million units of blood components are transfused every year. The need for blood is great - on any given day, approximately 39,000 units of Red Blood Cells are needed. Each year, we could meet only up to 1% (approx) of our nation’s demand for blood transfusion.</p>

<p>Despite the increase in the number of donors, blood remains in short supply during emergencies, mainly attributed to the lack of information and accessibility. We positively believe this tool can overcome most of these challenges by effectively connecting the blood donors with the blood recipients.</p>

</div>

</div>

</div>

<footer>

<div class="container">

<div class="row">

<div class="col-lg-3 footer-h2">

<h2>JMD BLOOD BANK</h2>

</div>

<div class="col-lg-9 footer-p1" >

<p>

We are world largest and trustful blood donation center. We have been working since 1973 with a prestigious vision to helping patient to provide blood.We are working all over the world, organizing blood donation campaign to grow awareness among the people to donate blood.

</p>

</div>

</div>

<hr>

<div class="col-lg-4">

<h4 class="footer-h4">SUBSCRIBE US</h4>

<p class="footer-p1" style="padding:10px 0;">Signup for regular newsletter and stay up to date with our lastest news.</p>

<input type="email" placeholder="Enter Your Email" class="footer-input">

<br>

<button class="btn btn-danger btn-lg btn-footer"><a href="Blood%20Donation.docx">JOIN NEWSLETTER</a></button>

</div>

<div class="col-lg-4">

</div>

<div class="col-lg-4">

</div>

</div>

<hr>

<div class="container">

<div class="row">

<div class="col-lg-6">

<p class="footer-p1">Copyright 2018-Blood Bank by GreenLight.All Rights Reserved.</p>

</div>

<div class="col-lg-6 footer-a-all">

<a href="#" class="footer-a1">Cookies</a>

<a href="#" class="footer-a1">Sitemap</a>

<a href="#" class="footer-a1">Feedback</a>

<a href="#" class="footer-a1 footer-a1-l">Privacy Policy</a>

</div>

</div>

</div>

</footer>

</body>

</html>

**ABOUTUS.CSS**

/\* navbar \*/

.navbar-default{

background-color: #fcf8e3;

color: cyan;

height: 90px;

padding-top: 20px;

}

.navbar-default .navbar-nav > li > a{

color: black;

font-size: 15px;

}

.navbar-default .navbar-brand{

color: #f03737;

text-transform: uppercase;

font-size: 29px;

font-weight: bolder;

}

/\* navbar \*/

/\* Footer\*/

/\* FOOTER \*/

footer{

background: black;

height: 600px;

}

.footer-p1{

/\*padding-top: 30px;\*/

padding: 30px 0px 10px 5px;

color: white;

font-size: 17px;

}

.footer-h2{

color: snow;

padding: 40px 0 0 18px;

}

.footer-a-all{

text-align: right;

padding: 30px 10px 10px 10px;

}

.footer-a1{

border-right: 1px solid white;

padding-left: 8px;

padding-right: 8px;

font-size: 15px;

color: white;

}

.footer-a1-l{

border-right: 0px;

}

.footer-input{

padding: 10px;

/\*margin-left: 5px;\*/

width: 100%;

margin-top: 10px;

margin-bottom:20px;

font-size: 18px;

background: #3D4039;

border: 0;

color: snow;

}

.footer-h4{

color: white;

padding-top: 10px;

}

.btn-footer{

background-color: rgb(225,25,30);

margin-bottom: 30px;

}

/\* Footer \*/

/\* Image - background \*/

#h-image{

background-image: url("685987201c28d78.jpg");

height: 500px;

background-size: cover;

}

/\* Image - background \*/

/\* main\*/

.main-h3{

color : #f03737;

}

/\*

To change this license header, choose License Headers in Project Properties.

To change this template file, choose Tools | Templates

and open the template in the editor.

\*/

/\*

Created on : 26 Nov, 2019, 12:57:13 PM

Author : Akhil

\*/

**BLOODTIPS.HTML**

<!DOCTYPE html>

<!--

To change this license header, choose License Headers in Project Properties.

To change this template file, choose Tools | Templates

and open the template in the editor.

-->

<!DOCTYPE html>

<!--

To change this license header, choose License Headers in Project Properties.

To change this template file, choose Tools | Templates

and open the template in the editor.

-->

< <html>

<head>

<title>JMD BLOOD BANK</title>

<link href="fp.css" rel="stylesheet" type="text/css">

<link rel="stylesheet" type="text/css" href="bootstrap.css">

<link rel="stylesheet" type="text/css" href="BloodTip.css">

<style>

body{

background-image:url("Pic5.jpg");

background-size:cover; }

</style>

</head>

<body>

<header>

<div class="row">

<ul class="main-nav">

<li class="active"><a href="donorprofile.html"> WELCOME DONOR</a></li>

<li><a href="bloodstock.html">BLOOD STOCK </a></li>

<li><a href="bloodcamps.html"> BLOOD CAMPS </a></li>

<li><a href="gallery.html"> GALLERY </a></li>

<li><a href="logout.jsp"> LOGOUT </a></li>

</ul>

</div>

<div class="Blood">

<div class="container" id="blood-container">

<h2 class="blood-h2">Blood Tips</h2>

<div class="blood-block">

<h3 class="blood-h3">Beat the myth</h3>

<div class="blood-block-p">

<p class="blood-p">Donating blood is safe and simple. It takes approximately 10-15 minutes to complete the blood donation process. Any healthy adult between 18 years and 60 years of age can donate blood. This is what you can expect when you are ready to donate blood:</p>

<ul>

<li>You walk into a reputed and safe blood donation centre or a mobile camp organized by a reputed institution.</li>

<li>A few questions will be asked to determine your health status (general questions on health, donation history etc). Usually you will be asked to fill out a short form.</li>

<li>Then a quick physical check will be done to check temperature, blood pressure, pulse and hemoglobin content in blood to ensure you are a healthy donor.</li>

<li>If found fit to donate, then you will be asked to lie down on a resting chair or a bed. Your arm will be thoroughly cleaned. Then using sterile equipments blood will be collected in a special plastic bag. Approximately 350 ml of blood will be collected in one donation. Those who weigh more than 60 Kg can donate 450 ml of blood.</li>

<li>Then you must rest and relax for a few minutes with a light snack and something refreshing to drink. Some snacks and juice will be provided.</li>

<li>Blood will be separated into components within eight hours of donation</li>

<li>The blood will then be taken to the laboratory for testing.</li>

<li>Once found safe, it will be kept in special storage and released when required.</li>

<li>The blood is now ready to be taken to the hospital, to save lives.</li>

</ul>

</div>

</div>

<div class="blood-block">

<h3 class="blood-h3">Blood Groups</h3>

<div class="blood-block-p">

<p>Blood type is determined by which antibodies and antigens the person's blood produces. An individual has two types of blood groups namely ABO-grouping and Rh-grouping. Rh is called as the Rhesus factor that has come to us from Rhesus monkeys.</p>

<p>Most humans are in the ABO blood group. The ABO group has four categories namely<br>

1) A group 2) B group 3) O group and 4) AB group</br>

In the Rh- group, either the individual is said to be Rh- Negative or Rh- Positive.</p>

<p>Thus blood group of any human being will mainly fall in any one of the following groups.<br>

A positive or A negative<br>

B positive or B negative<br>

O positive or O negative<br>

AB positive or AB negative</p>

</div>

</div>

<div class="blood-block">

<h3 class="blood-h3">Universal Donors and Recipients</h3>

<div class="blood-block-p">

<p>The most common blood type is O, followed by type A.</p>

<p>Type O individuals are often called "universal donors" since their blood can be transfused into persons with any blood type. Those with type AB blood are called "universal recipients" because they can receive blood of any type.</p>

<p>However, since approximately twice as many people in the general population have O and A blood types, a blood bank's need for this type of blood increases exponentially.</p>

</div>

</div>

<div class="blood-block">

<h3 class="blood-h3">DO donate blood,only if you satisfy all of the following conditions</h3>

<div class="blood-block-p">

<ul>

<li>You are between age group of 18-60 years.</li>

<li>Your weight is 45 kgs or more.</li>

<li>Your hemoglobin is 12.5 gm% minimum.</li>

<li>Your last blood donation was 3 or more months earlier.</li>

<li>You are healthy and have not suffered from malaria, typhoid or other transmissible disease in the recent past.</li>

</ul>

</div>

</div>

<div class="blood-block">

<h3 class="blood-h3">Do NOT donate blood,if you have any of the following conditions</h3>

<div class="blood-block-p">

<ul>

<li>Cold / fever in the past 1 week.</li>

<li>Under treatment with antibiotics or any other medication.</li>

<li>Cardiac problems, hypertension, epilepsy, diabetes (on insulin therapy), history of cancer, chronic kidney or liver disease, bleeding tendencies, venereal disease etc.</li>

<li>Major surgery in the last 6 months.</li>

<li>Vaccination in the last 24 hours.</li>

<li>Had a miscarriage in the last 6 months or have been pregnant / lactating in the last one year.</li>

<li>Had fainting attacks during last donation.</li>

<li>Have regularly received treatment with blood products.</li>

<li>Shared a needle to inject drugs/ have history of drug addiction.</li>

<li>Had sexual relations with different partners or with a high risk individual.</li>

<li>Been tested positive for antibodies to HIV.</li>

</ul>

</div>

</div>

<div class="blood-block">

<h3 class="blood-h3">A Healthy Donor</h3>

<div class="blood-block-p">

<p>A healthy diet helps ensure a successful blood donation, and also makes you feel better! Check out the following recommended foods to eat prior to your donation.</p>

<ul>

<li>Low fat foods</li>

<li>Iron rich foods</li>

</ul>

</div>

</div>

<div class="blood-block">

<h3 class="blood-h3">Blood Bank</h3>

<div class="blood-block-p">

<p>A blood bank is a place designed especially for the storage of blood and blood products. Large coolers hold these products at a constant temperature and they are available at a moment's notice.</p>

</div>

</div>

<div class="blood-block">

<h3 class="blood-h3">About Blood Donation</h3>

<div class="blood-block-p">

<p>Donating blood is a life saving measure especially when you have a rare blood type. Blood donation is safe and simple. It takes only about 10 minutes to donate blood - less than the time of an average telephone call. We can save upto 3 to 4 precious lives by donating blood.</p>

</div>

</div>

<div class="blood-block">

<h3 class="blood-h3">World Blood Donor Day</h3>

<div class="blood-block-p">

<p>The day is celebrated to raise awareness globally about the need for regular and voluntary blood donation.</p>

</div>

</div>

</div>

</div>

<footer>

<div class="container">

<div class="row">

<div class="col-lg-3 footer-h2">

<h2>BLOOD BANK</h2>

</div>

<div class="col-lg-9 footer-p1" >

<p>

We are world largest and trustful blood donation center. We have been working since 1973 with a prestigious vision to helping patient to provide blood.We are working all over the world, organizing blood donation campaign to grow awareness among the people to donate blood.

</p>

</div>

</div>

<hr>

<div class="col-lg-4">

<h4 class="footer-h4">SUBSCRIBE US</h4>

<p class="footer-p1" style="padding:10px 0;">Signup for regular newsletter and stay up to date with our lastest news.</p>

<input type="email" placeholder="Enter Your Email" class="footer-input">

<br>

<button class="btn btn-danger btn-lg btn-footer"><a href="Blood%20Donation.docx">JOIN NEWSLETTER</a></button>

</div>

<div class="col-lg-4">

</div>

<div class="col-lg-4">

</div>

</div>

<hr>

<div class="container">

<div class="row">

<div class="col-lg-6">

<p class="footer-p1">Copyright 2018-Blood Bank by GreenLight.All Rights Reserved.</p>

</div>

<div class="col-lg-6 footer-a-all">

<a href="#" class="footer-a1">Cookies</a>

<a href="#" class="footer-a1">Sitemap</a>

<a href="#" class="footer-a1">Feedback</a>

<a href="#" class="footer-a1 footer-a1-l">Privacy Policy</a>

</div>

</div>

</div>

</footer>

</header>

</body>

</html>

**BLOODTIP.CSS**

/\* navbar \*/

.navbar-default{

background-color: #fcf8e3;

color: cyan;

height: 90px;

padding-top: 20px;

}

.navbar-default .navbar-nav > li > a{

color: black;

font-size: 15px;

}

.navbar-default .navbar-brand{

color: #f03737ed;

text-transform: uppercase;

font-size: 29px;

font-weight: bolder;

}

/\* navbar \*/

/\* Blood Tip \*/

.Blood{

background: white;

}

#blood-container{

padding: 140px 300px 150px 300px;

width: auto;

}

.blood-block{

border:3px solid rgb(245,245,245);

border-radius: 3px;

background: white;

margin: 30px 20px;

}

.blood-h3{

color: #f03737ed;

background: rgb(245,245,245);

padding: 20px 40px 20px;

margin-top: 0px;

}

.blood-block-p{

color: rgb(102,102,102);

padding: 20px 40px;

}

/\* Footer\*/

/\* FOOTER \*/

footer{

background: black;

height: 600px;

}

.footer-p1{

/\*padding-top: 30px;\*/

padding: 30px 0px 10px 5px;

color: white;

font-size: 17px;

}

.footer-h2{

color: snow;

padding: 40px 0 0 18px;

}

.footer-a-all{

text-align: right;

padding: 30px 10px 10px 10px;

}

.footer-a1{

border-right: 1px solid white;

padding-left: 8px;

padding-right: 8px;

font-size: 15px;

color: white;

}

.footer-a1-l{

border-right: 0px;

}

.footer-input{

padding: 10px;

/\*margin-left: 5px;\*/

width: 100%;

margin-top: 10px;

margin-bottom:20px;

font-size: 18px;

background: #3D4039;

border: 0;

color: snow;

}

.footer-h4{

color: white;

padding-top: 10px;

}

.btn-footer{

background-color: rgb(225,25,30);

margin-bottom: 30px;

}/\*

To change this license header, choose License Headers in Project Properties.

To change this template file, choose Tools | Templates

and open the template in the editor.

\*/

/\*

Created on : 27 Nov, 2019, 6:51:20 PM

Author : Akhil

\*/

FUTURE SCOPE

The main advantage of the project is that it provides different types of registered users to

be verified by the administrator. Admin can interact with the selected users through

according blood report only. Now this site is free, after few months, registered users have

to pay monthly for better and good services.

In future, we can have the search engine that can provide the result on the basis of

different criteria to search. We can also have one more module of implementing this

project on web.

•

One year maintenance and software support for the proposed application software

will be provided free of cost (provided change of scope is not raised).

•

The proposed software is completely modular and therefore it facilitates for

extendibility i.e. activities for new establishment can be easily added without

affecting the main program with less effort and cost.

•

Changes in GUIs based screen can be easily incorporated but it will be treated as

changed request and it will be charged.

•

It will be portable can be implemented on most of the available web based

platform.

•

Some part of the proposed Application System can be converted to web based

software by using web based scripting languages and tools.

•

Complete software will be easily upgraded (i.e. versioning of software is possible)

and will show downward compatibility.

BIBLIOGRAPHY

1. KK Aggarwal Software Engineering
2. www.uianduxdesign.com
3. www.w3techs.com
4. A Practicitioner Approach Software Engineering