****

**Tribhuvan University**

**Faculty of Humanities and Social Sciences**

# BLOOD BANK MANAGEMENT SYSTEM

**A PROJECT REPORT**

**Submitted to**

## Department of Computer Application

**Ambition College**

**Mid-Baneshwor, Kathmandu**

***In partial fulfillment of the requirements for the Bachelors in Computer Application***

Submitted by

Jaison Maharjan (TU Roll No.: - 1810581098)

September 2021

Under the Supervision of

**Mr. Dipak Prasad Bhatt**

****

**Tribhuvan University**

**Faculty of Humanities and Social Sciences**

**Ambition College**

**Mid-Baneshwor, Kathmandu**

## Supervisor’s Recommendation

I hereby recommend that this project prepared under my supervision by JAISON MAHARJAN and MENUKA WAGLE entitled “**BLOOD BANK MANAGEMENT SYSTEM”** in partial fulfillment of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

**SIGNATURE**

Dipak Prasad Bhatt

**SUPERVISOR**

Lecturer

BCA Department

Ambition College

Mid-Baneshwor, Kathmandu

****

**Tribhuvan University**

## Faculty of Humanities and Social Sciences

**Ambition College**

**Mid-Baneshwor, Kathmandu**

**LETTER OF APPROVAL**

This is to certify that this project prepared by JAISON MAHARJAN AND MENUKA WAGLE entitled “**BLOOD BANK MANAGEMENT SYSTEM”** in partial fulfillment of the requirements for the degree of Bachelor in Computer Application has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

|  |  |
| --- | --- |
| Dipak Prasad Ghimire  Lecturer  BCA Department  Ambition College  Mid-Baneshwor, Kathmandu | Bishnu Kumar Ghimire  Campus Chief  Ambition College  Mid-Baneshwor, Kathmandu |

|  |  |
| --- | --- |
| **Internal Examiner** | **External Examiner** |

**ACKNOWLEDGEMENT**

We would like to express our sincere gratitude to Project Supervisor **Mr. Dipak Prasad Bhatt**, who continuously helped and guided us through different problems faced during this whole coding process. We would like to thank our BCA coordinator **Mr. Shreeram Bohara** for assisting in the different scope and helping us to solve obstacles. We are very grateful to our friends who helped us despite of their busy schedule.

Finally, we would like to thank **Ambition College** for providing all the resources and ideal environment for the project development process and encourage us to develop such a project. We would also like to share our gratitude to Tribhuvan University for providing the opportunity to evaluate our programming skills through this project.

**ABSTRACT**

Blood Bank Management System (BBMS) is a web-based system that is designed to store, process, retrieve and analyze information concerned with the administrative and inventory management within a blood bank. This project aims at maintaining all the information pertaining to blood donors, different blood groups available in each blood bank and helps them manage in a better way. Aim of this blood bank is to provide transparency in this field, make the process of obtaining blood from a blood hassle free and corruption free and make the system of blood bank management effective

***Keywords:*** *blood bank, blood bank management System*

**TABLE OF CONTENTS**

Acknowledgment..................................................................................................i

Abstract....................................................................................................................ii

List of Tables ..........................................................................................................vi

List of Figures………………………………………………………………….… vii

List of Abbreviation.................................................................................................viii

**Chapter 1: Introduction.......................................................................1**

1.1. Introduction .......................................................................................................1

1.2. Problem Statement ...........................................................................................2

1.3. Objectives..........................................................................................................2

1.4. Scope and Limitation .......................................................................................2

1.4.1 Scope……………………………………………………………….…2

1.4.2 Limitation……………………………………………………….…….2

1.5. Report Organization...........................................................................................3

**Chapter 2: Literature Review………………………………….…....4**

2.1 Background Study ............................................................................................4

2.2 Literature Review……………………………………………………………...5

**Chapter 3: System Analysis…………………………………………6**

3.1. System Analysis …………………...................................................................6

3.2 Requirement Analysis……................................................................................6

3.2.1 Functional Requirement..................................................................................6

3.2.2 Non-Functional Requirement .........................................................................6

3.3 Feasibility Analysis............................................................................................6

3.3.1Technical feasibility ............................................................................6

3.3.2Economic feasibility ...........................................................................7

3.3.3Operational feasibility..........................................................................7

3.3.4Schedule feasibility..............................................................................7

**Chapter 4: System Design…………………………...........................8**

4.1 System Architecture…………………………………………………………...8

4.2 Entity Relation Diagram……………………………………………………….9

4.3 Data Flow Diagram……………………………………………………………10

4.3.1 Level 0 Diagram………………………………………………….….10

4.3.2 Level 1 Diagram………………………………………………….….11

4.4 Use Case Diagram ………………………………………………………....….12

4.5 Class Diagram……………………………………………………...…...….….13

4.6 Activity Diagram………………………………………………………….......,14

**Chapter 5: System Development and Implementation…................15**

5.1 Programming Platform……………………………………………………….,.15

5.2 Operating Requirement………………………………………………….….….15

5.2.1 Hardware Requirement …………………………………….….…….15

5.2.2 Software Requirement ……………………………………………….15

5.3 Testing and Debugging ……………………………………………………….,.15

5.3.1 Unit and Integration Testing …………………………………………………15

5.3.2 Equivalence and Boundary value……………………………………….……16

**Chapter 6: Conclusion and Future Enhancement ............................18**

6.1 Conclusion………………………………………………………………………18

6.2 Limitations…………………………………………………………….………...18

6.3 Future Enhancement …………………………………………………………....18

REFERENCE……………………………………………………………………….19

APPENDIX I System Screenshots

APPENDIX II-Source Code

**List of Figures**

Figure 3.3.4: Planned Gantt chart …………………………………………….…………7

Figure 4.1: System Architecture………………………………………………….….......8

Figure 4.2: Entity Relationship Diagram…………….......................................................9

Figure 4.3.1: DFD Level 0…………………………………………………………...….10

Figure 4.3.2 DFD Level 1 ……………............................................................................11

Figure 4.4: Use Case Diagram …………….....................................................................12

Figure 4.5: Class Diagram ……………............................................................................13

Figure 4.6: Activity Diagram …………….......................................................................14

**List of Tables**

Table 4.1: Test Cases for Password Id …….………………………………………….16

Table 4.2: Test Cases for Phone no Field …………………………………………......17

Table 4.4: Test Cases for Small Field………………………………………………….17

**LIST OFABBREVIATIONS**

CSS Cascading Style Sheet

DFD Data Flow Diagram

ERD Entity Relationship Diagram

HTML Hypertext Markup Language

JS JavaScript

PHP Hypertext Preprocessor

SQL Structured Query Language\

# Chapter 1: Introduction

# 1.1 Introduction

Blood Bank Management System (BBMS) is the basically a platform where donor can donate blood which is used when necessary and donor are called and asked for it. There is the database, which contains all the information of donor. Whenever new donor will register its name, update is made in database. Since, there are large people who want profit by making blood bank system as a business that is why this type of blood bank system is created. The main aim of this System is to provide service without corruption. This System is designed to keep the information of the donor in advanced form so that it can be easily managed. There will be relation between three parties. Admin, Donor and Hospital. Admin will be responsible for all the task of blood bank system. The donors who are interested in donating blood will register in the database. The software will be fully integrated with CRM (Customer Relationship Management) as well as CMS (Content Management System). It will be developed in a manner that will be easily manageable, timesaving. The requirement of the blood has to be requested and after that Admin will supply the information of donor.

The donor will update their status either they are available or not. There will be three panel in the system, Admin, Hospital and User. The admin is responsible for every updates of details. It can change the data if there is any blood request or registration for blood donation. Admin can access every data of the system. Similarly, Hospital can access user data and if emergency can call user for blood request or email to all in case user is not available. Similarly, the third panel i.e. User Panel, login the system, browse through the available items , go to available item and get schedule for donating or receiving blood and if the queries are full. His/her should wait for few days until they get call or mail.

**1.2 Problem Statement**

The current situation of the blood bank and the blood bank management system were analyzed and the following problems were found such as Lack of computerized Database Management System, the information /data which are in file format are hard to access and it takes a lot of time.

There may be chance of data stole and loss of data in file system. Since, database management system is the modern/advance form for storing, retrieving data whenever necessary. There was also lack of workers to operate legacy software; Graphics are not in proper way, improper graphical interface. In today’s time, no one has time to visit every hospital and blood bank system for query of blood so our system will help people.

**1.3 Objectives**

The objectives of this project are:

* To develop a system that deals with the day-to-day data management of blood, user’s record and receiver’s record in the bank.
* To store a proper computerized database of blood, donors and receiver.

**1.4 Scope and Limitations**

**1.4.1 Scope**

Blood Bank system will be a web-based application whose main language of programming will be HTML. Its main aim is to simplify and improve the efficiency of the blood request process for both users and the hospital, minimize manual data entry and ensure data accuracy and security during or blood request placement process. Users will also be able to view blood groups and their availability and be able to have a visual confirmation that request was placed correctly.

**1.4.2 Limitations**

Requires internet connection and the user must be computer literate. The set back  of the system is that the user  targeted are adults with access to computer  systems while the minors might have to go physically to the adults  to request  the blood that they want or request  blood with the help of an adult. The other limitation is that the system will only be convenient to the people in good geographical region, where there is internet facility.

**1.5 Report Organization**

This document contains six chapters which are described below:

**The Chapter One:** Deals about introduction to the system, it answers “what the result of the system is?” what does it does?” It also states problem of statement what existing problem lags objectives of system slopes, limitation of the system.

**The Chapter Two:** It overview existing system. It studies the features advantage and disadvantage of similar system that existed on past.

**The Chapter Three:** Is system analysis it describes the method on which basis system should be build. It basically causes on the requirement analysis after that requirement is specified, both functionally and nonfunctional. It also discusses about the feasibility of the system.

**The Chapter Four:** Is System design. It consists of soul part of the system. Everything lies here did, flowchart, system architecture, use care diagram, database design. UI/U mechanisms are drawn here.

**The Chapter Five:** Is system development and implementation. It shows how system is developed? Till the coding activities, frontend, backend design is explained in this unit operating environment and hardware requirement is mentioned here. After product is built it passes through various tested. After being tested, it is implemented and result is analyzed

**The Chapter Six:** Is conclusion and future enhancement. It consists of what we concluded and future enhancement consists of additional features in coming days.

**Chapter 2: Literature Review**

The online blood bank management system is one of the latest servicers most hospitals in the world are adopting. With this method, blood is requested online and delivered to the patient. This is made possible through the use of electronic web based system. It will not charge users but they can donate to serve system efficient so that we can serve more people like them. Therefore, the system designed in this project will enable users go online and place request for their blood. Due to the great increase in the awareness of internet and the technologies associated with it, several opportunities are coming up on the web. So many social platforms now venture into their service with ease because of the internet.

**2.1 Background Study:**

What we propose is an online requesting system originally designed for use in primary hospital, but just as applicable in any blood bank system. The main advantage of this system is that it greatly simplifies the requesting process for both the customer and the hospital. The system also greatly lightens the load on the hospitals end, as the entire process of taking requests is automated. Once a request is placed on the webpage that will be designed, it is placed into the database and then retrieved, in pretty much real-time, by a desktop application on the hospitals end. Within this application, all items in the request are displayed, along with their corresponding options and details, in a concise and easy to read manner. This allows the hospital employees to quickly go through the requests as they are placed and provide the service with minimal delay and confusion. The greatest advantage of this system is its flexibility.

**2.2 Literature Review**

Teena, C.A, Sankar, K. and Kannan, S. (2014) in their study entitled “A Study on Blood Bank Management”, they defined Blood Bank Information System as an information management system that contributes to the management of donor records and blood bank. Their system allowed an authorized blood bank administrator to sign in with a password to manage easily the records of donors and patients who need blood.

The system provided many features including the central database, quick access to the system content through the login, includes the search code to find donors on a given basis, and the ease of adding and updating donor data. The main aim of the system was to complete0the process of the blood bank. This system was designed to suit all types of blood banks.

Once successful in the implementation of the application, it can be applied and rolled out in several blood banks. This application contains User Login Screen, Blood Management, Menu Form, Blood Stock, Donor Management, Donor Registration, Blood Reservation, Donor Blood Test, Recipient Management and Blood Reservation. In similar manner, the researchers planned in their application to have hospital administrator, doctors, and blood bank receptionists as users.

The authors did not mention the research method they used, and failed to provide screenshots of the system prototypes, making difficult for the researchers to visualize their application. No discussion also for their respondents, samples and sampling techniques used. Subsequently, the researchers planned to provide figures to explain the system, screenshots of system prototypes, and other diagrams that can help other researchers to visualize the development of web-based blood bank management system. In addition, the researchers will explicitly discuss its research methods, sampling procedures, and statistical treatment to be used for analyzing the gathered data. [1]

**Chapter 3: System Analysis**

**3.1 System Development Model**

Traditional waterfall methodology is used for the development of this system waterfall method is a linear project management approach, where stakeholder and customer requirements are gathered of the beginning of project, and then a sequential project plan is created to accommodate those requirements. The model is named so because each phase of project cascades into the next following steadily down like a waterfall. It is built with required and specific documentation, fixed requirement and enough time, thus the project is as easily developed following the waterfall methodology.

**3.2 Requirement specification**

**3.2.1 Functional Requirements:**

Blood Bank Management will have following functional requirement

* User registration(Donor)
* User Login/Logout/Update Profile
* Request for a blood campaign
* Change personal contact

**3.2.2 Non-functional Requirements**

* Capacity, Scalability
* Availability
* Security
* Maintainability
* Performance Requirement

**3.3 Feasibility study**

**3.3.1 Technical feasibility**

The system is technically feasible as the requirement for the development of system is easily accessible. Necessary hardware and software required for the development and implementation of the system is available. For the technical knowledge different programming tasks is being studied.

**3.3.2 Economy feasibility**

The system is economically feasible and cost effective. It will be cost effective and beneficial for the organization. A computerized system works equivalent to multiple manual workers. And the system secures task which ultimately reduces cost of organization.

**3.3.3 Operation feasibility**

System is easy to operate with the basic knowledge of computer and internet. Well trained man power is not necessary. User can also easily access the system of it is user friendly in many problems which are faced in manual reservation.

**3.3.4 Schedule feasibility**

The project is feasible within given schedule. As the project is well understand and predictable, requirement gathering and analysis process can be completed properly on time and hence in short. Similarly the modules of the project, its front and backend task can be completed within the deadline.

**Gantt chart**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Stages | Start Date | End Date | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Status |
| 1.Initation/Planning | 31-Jan | 15-Feb |  |  |  |  |  |  |  |  |  | Completed |
| 2.Requirement Analysis | 16-Feb | 22-Feb |  |  |  |  |  |  |  |  |  | Completed |
| 3. Design | 23-Mar | 28-Mar |  |  |  |  |  |  |  |  |  | Completed |
| 4.Development | 1-Apr | 25-May |  |  |  |  |  |  |  |  |  | Completed |
| 5.Testing | 26-Jun | 31-Jun |  |  |  |  |  |  |  |  |  | Completed |
| 6.Implementation | 1-July | 20-July |  |  |  |  |  |  |  |  |  | Completed |

***Figure 3.3.4*:** *Planned Gantt chart*

**Chapter 4: System Design**

**4.1 System Architecture**

Database

Server

Internet

Web element

-Location update service(LUS)

-Distress call fetch service (DCFS)

Control Server

-Record current volunteer

-Find nearby volunteer

-Forwarding Distress calls (SMS)

-Forwarding Help Confirmation

-Send notification manually

-Search for volunteer in a specific area

-Registration

-Sending Distress Call

-Receive Confirmation

-Registration

-Send Location Update

-Receive Distress call notification

Send Help Confirmation

Citizen Mobile Application (CMS)

Volunteer Mobile Application (VMP)

***Figure4.1****: System Architecture*

**4.2 Entity Relation Diagram**

Donor

Hospital

Blood Bank

Receptionist

Blood Bank Manager

Blood

Donate

Registers

Stored

Works

Manages

Order

***Figure 4.2:*** *Entity Relationship Diagram*

**4.2 Fig: ER Diagram.**

**4.3.1 Level 0 Diagram**

System User Management

Donor Management

Request Management

Login Management

Group Management

Blood Management

***Figure 4.3.1:*** *DFD Level 0*

**4.3.2 Level 1 Diagram**

Blood Bank Management

Group Management

Donor Management

Blood Stock Management

Login Management

System User Management

Generate Blood Report

Generate Group Report

Generate Donor Report

Generate Blood Stock Report

Check User Login Details

Generate System User Reports

***Figure 4.3.2:*** *DFD Level 1*

**4.4 Use Case Diagram**

***Figure 4.4:*** *Use Case Diagram*

**4.5 Class Diagram**

DONOR

DONATION CAMP

BLOOD BANK

HOSPITAL

Register name

Check user

Donate blood

Providereceipt

Sent blood for testing and storing

Acknowledgement

Give order for blood

Provide blood

Acknowledgment

***Figure 4.5:*** *Class Diagram*

**4.6 Activity Case Diagram**

Admin is registered

Admin login and password

Check Login ID password

Set username and permissions

Admin login and password

Access the internal functionalities according permission

***Figure 4.6:*** *Activity Diagram*

**Chapter 5: System Development and Implementation**

**5.1 Programming Platform**

The entire system is built on Html, Css, Java& PHP. The interfaces and front end designs are made using Java Swing. The application uses MySQL database to store data and information. The system executes on any windows and/or Linux environment with apache server installed in it.

**5.2 Operating Requirement**

**5.2.1 Hardware Requirement**

* Processor: Intel Pentium IV 2.0 GHz and above
* RAM: 512 MB and above
* Hard disk: 80 GB and above
* Monitor: CTR or LED or LCD
* Keyboard: Normal

**5.2.2 Software Requirement**

* Server: XAMPP
* Language:Html, Css, Java, PHP.
* Back End: PHP,MySQL
* Operating System: Windows and/or Linux

**5.3 Testing and debugging**

**5.3.1 Unit and Integration Testing**

Unit and integration is done in each module. In unit testing small testable parts of application are tested individually and independently, unit test was don’t by team members themselves and the test was used for database test, records of each table, basic function test, add donor, add blood.

In integration testing each module is combined and tested as a group. This test verified that the system maintained data integrity and can operate in coordination with other systems in the same environment. The test was carried out by ourselves.

**5.3.2 Equivalence and Boundary value**

* Password field accept minimum 6-16 character, must include at least one numeric value, one uppercase letter, one lowercase letter and one special character.

**Test Cases**

|  |  |  |
| --- | --- | --- |
| Test Scenario | Test Scenario Description | Expected Outcome |
| **1** | Enter less than 6 characters in the password field | System should not accept |
| **2** | Enter more than 16 characters in the password field | System should not accept |
| **3** | Enter 6-16 characters in the password field with at least 1 uppercase letter, one lowercase letter, 1 digit and 1 special character | System should accept |
| **4** | Else | System should not accept |

***Table 5.3.2*:** *Test Cases for Password Id*

* Phone number field should accept only 10-13 digits’ number.

**Test Cases**

|  |  |  |
| --- | --- | --- |
| Test Scenario | Test Scenario Description | Expected Outcome |
| 1 | Enter less than 10 digits number | System should not accept |
| 2 | Enter more than 13 digits numbers | System should not accept |
| 3 | Enter 10-13 digits numbers | System should accept |

***Table 5.3.2:*** *Test Cases for Phone no Field*

* Email field should accept only those addresses that have @ and a ‘.’ in it.

Test Cases

|  |  |  |
| --- | --- | --- |
| Test Scenario | Test Scenario Description | Expected Outcome |
| 1 | Address without an @ and a dot | System should not accept |
| 2 | Address with an @ but without a dot | System should not accept |
| 3 | Address with an @ but with a dot | System should not accept |
| 4 | Address with an @ and a dot | System should accept |

***Table 5.3.2:*** *Test Cases for Small Field*

**Chapter 6: Conclusion and Future Enhancement**

**6.1 Conclusion**

In conclusion, a blood bank management system is a website which will help to optimize and have control over people’s health at same extent. It is a process of blood donation using a trick of modern technology though webpage. If reduces the manual work and improves the efficiency of hospital. This system helps to request for blood. Software keeps the day to day and person to person record so that there is less chance of corruption. It is very useful in keeping the record of database. The system has been developed using different tools such as HTML, CSS, PHP, JAVASCRIPT AND MYSQL.

It is always better to make a manual system automated in order to make blood bank well manage, keep track of records, donor history generate report as required to increase the efficiency of hospital a web based ‘Blood bank management system’ is required. We will be continuously working in this process to make it more user friendly and more secure. We sure implementing our system in hospital, will make the data much managed than the traditional file system.

**6.2 Limitation**

* Patient can’t have direct access to the system.
* We couldn’t implement advance setting like sending automatic mails to donor whenever there is blood request.
* It is web-based application and cannot be used offline.

**6.3 Future enhancement**

Following future enhancement wil be made to this system.

* Patient can make query offline.
* People can visit the site except for admin panel.
* The system will be made easily available
* Many other features as per requirement.

# References

[1]Nawaf Hamed Al-Washahi 66j137

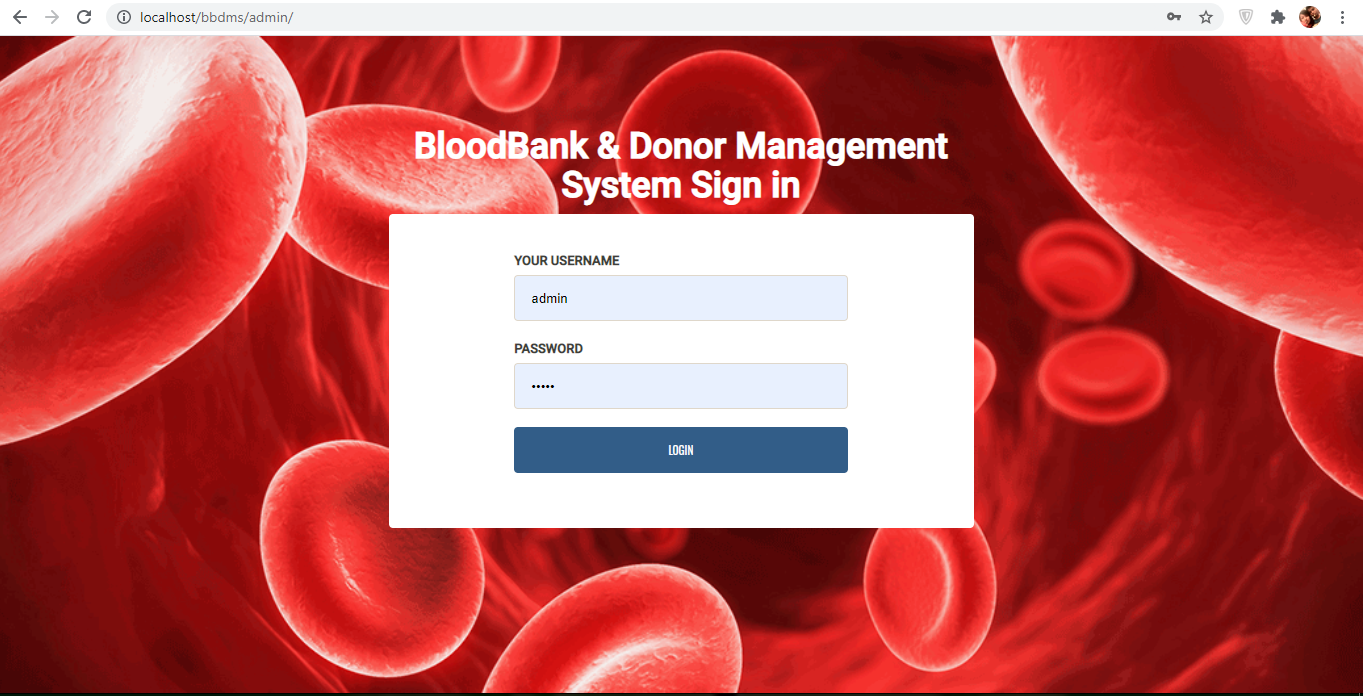
2 Alsafa Ahmed Aal-abdulsalam 66j1468

3 AfrahSulaiman Al-Mamari

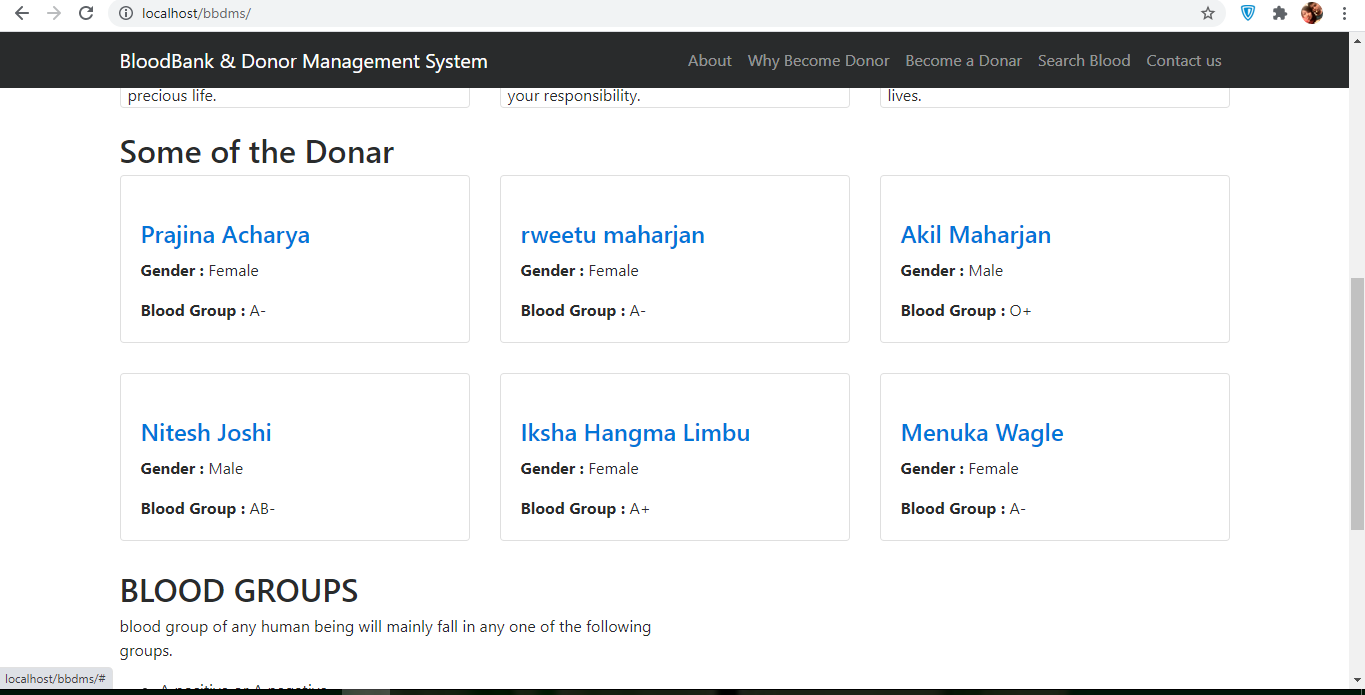
[1] HamedAl-WashahiNawaf, Ahmed Aal-abdulsalamAlsafa, Al-MamariAfrahSulaiman, (October 2018), Applied Research Project Report*,*[*Enhancing Blood Transfusion Safety Through the Use of Online Blood Bank Management System in Oman*](https://www.researchgate.net/project/Enhancing-Blood-Transfusion-Safety-Through-the-Use-of-Online-Blood-Bank-Management-System-in-Oman)[online] Available at:

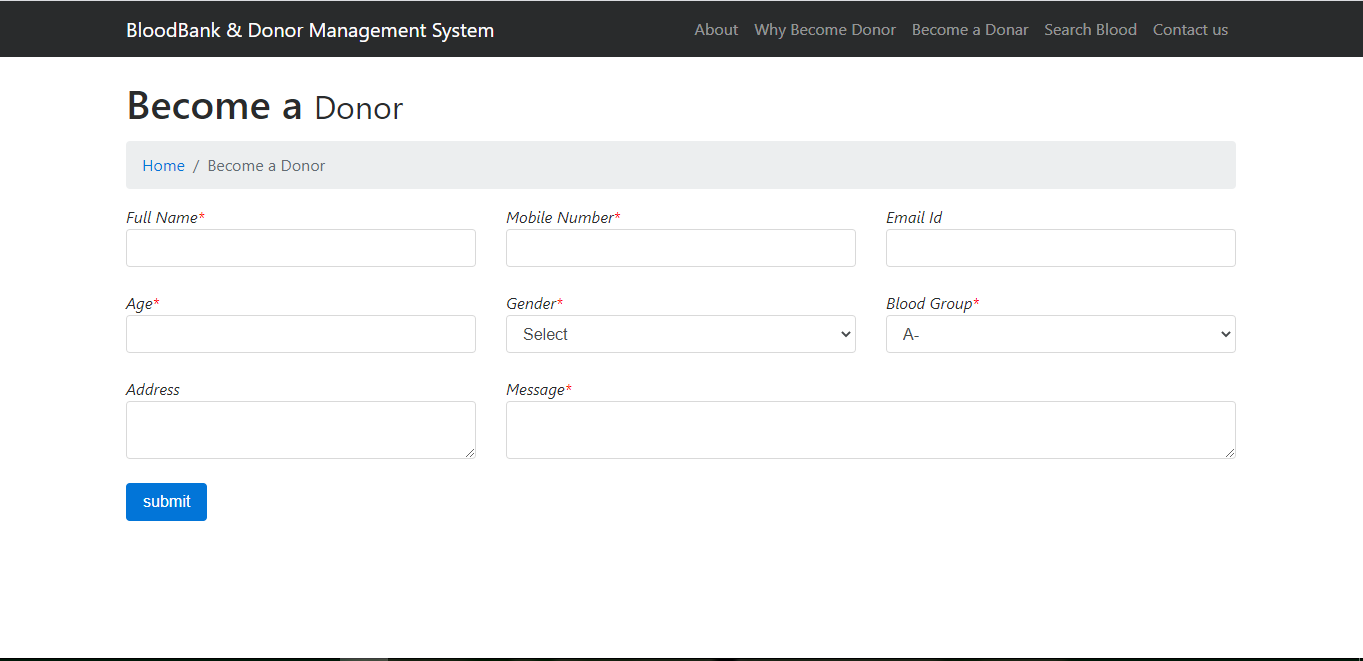
https://www.researchgate.net/publication/328305822\_Blood\_Bank\_System/link/5bc59176458515f7d9bf4a58/download

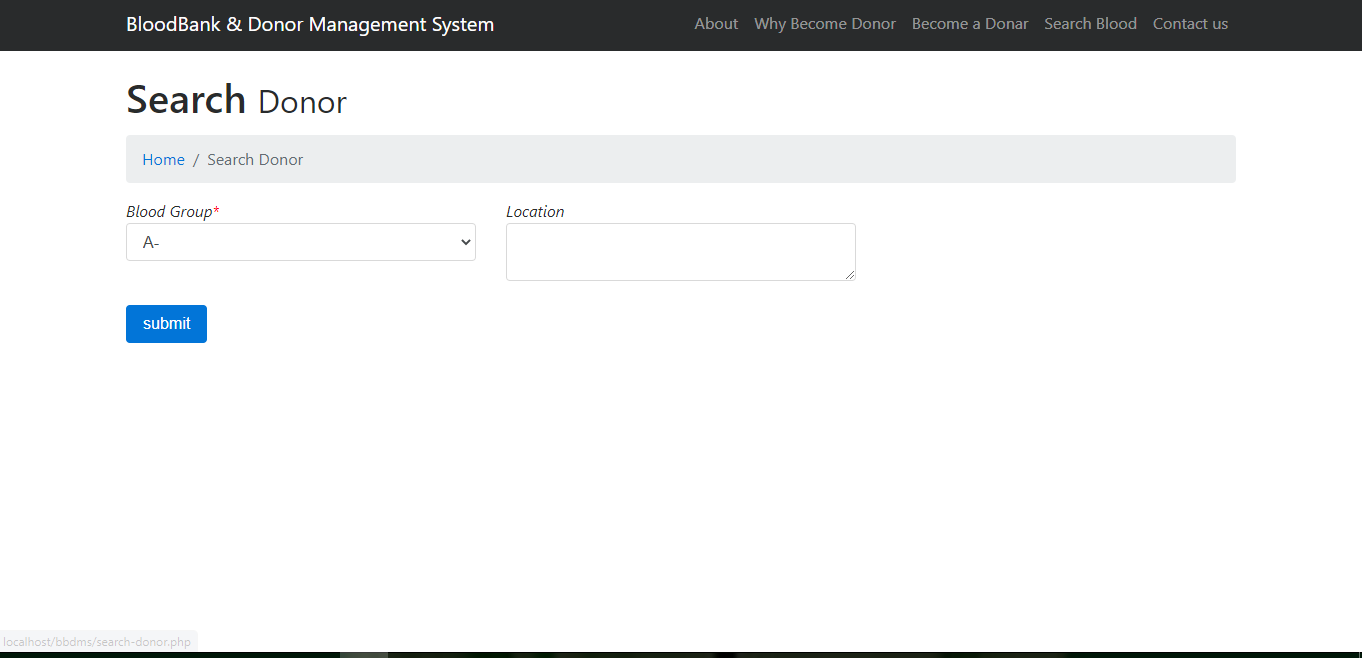
**Appendix I- System Screenshots**

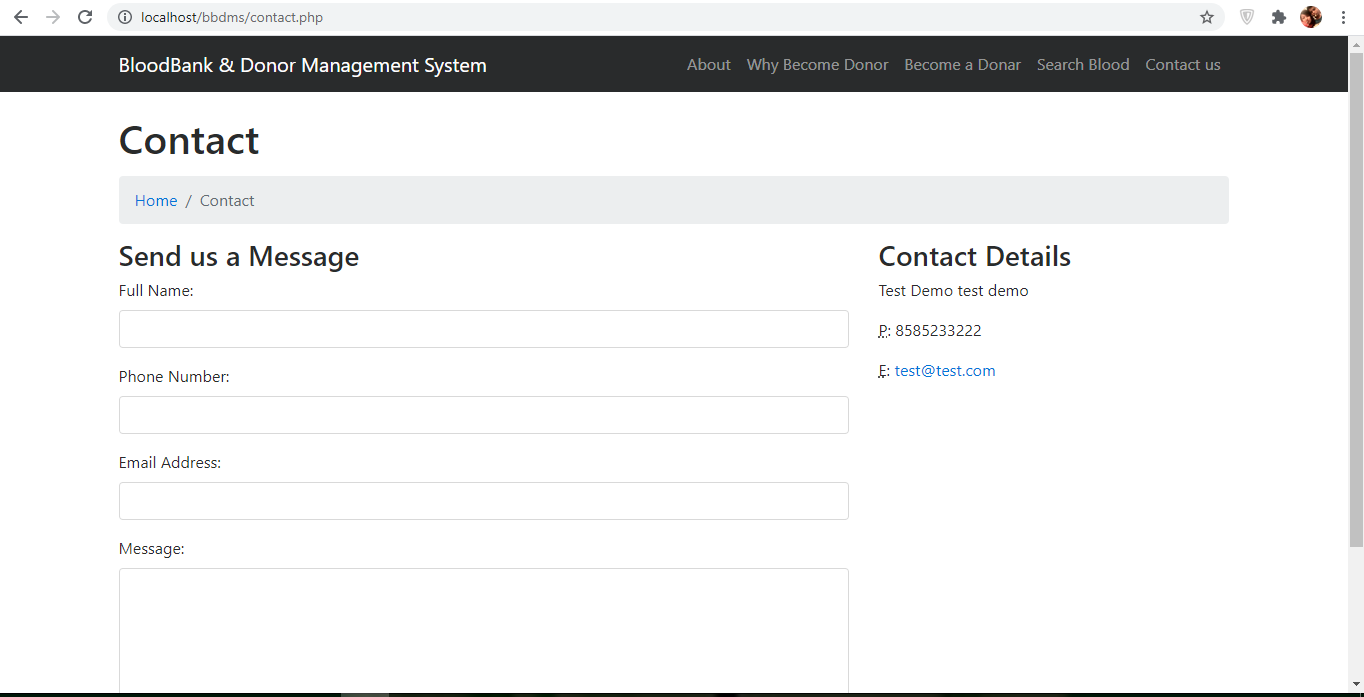
****

****

****

****

****

****

**Appendix II- Source Code**

Db.php

<?php

//Data Source Network(DSN) takes host and dbname as parametersw

$DSN = 'mysql:host=localhost; dbname=sports\_cms';

$conn = new PDO($DSN,'root','');

?>

Functions.php

<?php require\_once("includes/db.php"); ?>

<?php

// Function to redirect the user to same page if field is empty

function redirect($Location){

header("Location: $Location");

exit;

}

function login($UserName,$Password){

//Check for valid username and password from DB

global $conn;

$sql = "SELECT username,password FROM admin

WHERE username=:userName AND password=:passWord";

$stmt = $conn->prepare($sql);

$stmt->bindValue(':userName' ,$UserName);

$stmt->bindValue(':passWord' ,$Password);

$stmt->execute();

//Counting each row which hold sql query result

$show = $stmt->rowcount();

if($show==1){

//fetch result storedin fetch\_login variable

return $fetch\_login = $stmt->fetch();

}else{

return null;

}

}

function login\_confirm(){

//password protecting backend pages from accessing after logging out

if(isset($\_SESSION["Id"])){

return true;

}else{

$\_SESSION["errorMsg"] = "Please enter your login details";

redirect("login.php");

}

}

?>

Sessions.php

<?php

session\_start();

function errorMsg(){

if(isset($\_SESSION["errorMsg"])){

$show = "<div class=\"alert alert-danger\">";

//Don't allow html syntax to be broken

$show .= htmlentities($\_SESSION["errorMsg"]);

$show .= "</div>";

//Clearing the session

$\_SESSION["errorMsg"] = NULL;

return $show;

}

}

function successMsg(){

if(isset($\_SESSION["successMsg"])){

$show = "<div class=\"alert alert-success\">";

$show .= htmlentities($\_SESSION["successMsg"]);

$show .= "</div>";

$\_SESSION["successMsg"] = NULL;

return $show;

}

}

?>

Categories.php

<?php require\_once("includes/db.php"); ?>

<?php require\_once("includes/functions.php"); ?>

<?php require\_once("includes/sessions.php"); ?>

<?php //login\_confirm(); ?>

<?php

if(isset($\_POST["Submit"])){

$categoryName = $\_POST["CategoryName"];

$Admin = $\_SESSION["Name"];

//Get current date and time

date\_default\_timezone\_set("Asia/Kathmandu");

$date=time();

$get\_time=strftime("%d-%m-%Y %H:%M:%S",$date);

if(empty($categoryName)){

//Making the error into session variaoble using super global variable

$\_SESSION["errorMsg"] = "Field(s) cannot be empty.";

redirect("Categories.php");

//Check for category length

}else if(strlen($categoryName)<4){

$\_SESSION["errorMsg"] = "Category must be atleast 2 characters long.";

redirect("Categories.php");

}else{

//Inserting title when validation is true

$sql = "INSERT into category(name, author, datetime)";

//PDO named dummy paramerter to prevent sql injection

$sql .= "VALUES(:nameCategory, :authorName, :dateTime)";

//PDO object notation to call prepare mathod

$stmt = $conn->prepare($sql);

//Bind dummy values to actual values

$stmt->bindValue(':nameCategory', $categoryName);

$stmt->bindValue(':authorName', $Admin);

$stmt->bindValue(':dateTime', $get\_time);

//PDO ,ethod execute called via $stmt object

$execute=$stmt->execute();

//Check for successfull addition to DB

if($execute){

$\_SESSION["successMsg"] = "Category added successfully!";

redirect("Categories.php");

}else{

$\_SESSION["errorMsg"] = "Oops! Something went wrong, please try

again.";

redirect("Categories.php");

}

}

}

?>

Admin.php

<?php require\_once("includes/db.php"); ?>

<?php require\_once("includes/functions.php"); ?>

<?php require\_once("includes/sessions.php"); ?>

<?php //login\_confirm(); ?>

<?php

if(isset($\_POST["Submit"])){

$UserName = $\_POST["username"];

$Password = $\_POST["password"];

$ConfPassword = $\_POST["confirmPassword"];

$Admin = $\_SESSION["Name"];

//Get current date and time

date\_default\_timezone\_set("Asia/Kathmandu");

$date=time();

$get\_time=strftime("%d-%m-%Y %H:%M:%S",$date);

if(empty($UserName) || empty($Password) || empty($ConfPassword)){

//Making the error into session variaoble using super global variable

$\_SESSION["errorMsg"] = "Field(s) cannot be empty.";

redirect("admin.php");

//Check for category length

}else if(strlen($Password) AND strlen($ConfPassword)<=7){

$\_SESSION["errorMsg"] = "Password must be atleast 8 characters long.";

redirect("admin.php");

}else if($Password !== $ConfPassword){

$\_SESSION["errorMsg"] = "Password mismatch error.";

redirect("admin.php");

//Check if udername already exists or not

}else{

//Inserting admin details when validation is true

$sql = "INSERT into admin(datetime, username, password)";

//PDO named dummy paramerter to prevent sql injection

$sql .= "VALUES(:dateTime, :user, :pass)";

//PDO object notation to call prepare mathod

$stmt = $conn->prepare($sql);

//Bind dummy values to actual values

$stmt->bindValue(':dateTime', $get\_time);

$stmt->bindValue(':user', $UserName);

$stmt->bindValue(':pass', $Password);

//PDO ,ethod execute called via $stmt object

$execute=$stmt->execute();

//Check for successfull addition to DB

if($execute){

$\_SESSION["successMsg"] = "Admin added successfully!";

redirect("admin.php");

}else{

$\_SESSION["errorMsg"] = "Oops! Something went wrong, please try

again.";

redirect("admin.php");

}

}

}

?>

Login.php

<?php

if(isset($\_POST["login\_button"])){

$UserName = $\_POST["Username"];

$Password = $\_POST["Password"];

if(empty($UserName) || empty($Password)){

$\_SESSION["errorMsg"] = "Field(s) cannot be empty.";

redirect("login.php");

}else{

//function call to validate user credentials from DB

$fetch\_login=login($UserName, $Password);

if($fetch\_login){

//returning column data from fetch() method

$\_SESSION["Id"] = $fetch\_login["id"];

$\_SESSION["Name"] = $fetch\_login["username"];

$\_SESSION["successMsg"] = "Welcome " .$\_SESSION["Name"];

redirect("Dashboard.php");

}else{

$\_SESSION["errorMsg"] = "Sorry! Incorrect login details.";

redirect("login.php");

}

}

}

?>

Logout.php

<?php

//making all user credential to null before logging out

$\_SESSION["Id"] = NULL;

$\_SESSION["Name"] = NULL;

$\_SESSION["successMsg"] = NULL;

session\_destroy();

redirect("login.php");

?>

DeleteCategory.php

<?php

if(isset($\_GET["id"])){

$searchParam = $\_GET["id"];

global $conn;

$sql = "DELETE FROM category WHERE id='$searchParam'";

$execute = $conn->query($sql);

if($execute){

$\_SESSION["successMsg"] = "Category deleted sucessfully.";

redirect("Categories.php");

}else{

$\_SESSION["errorMsg"] = "Oops! Something went wrong.";

redirect("Categories.php");

}

}

?>

DeleteComment.php

<?php

if(isset($\_GET["id"])){

$searchParam = $\_GET["id"];

global $conn;

$sql = "DELETE FROM comment WHERE id='$searchParam'";

$execute = $conn->query($sql);

if($execute){

$\_SESSION["successMsg"] = "Comment deleted sucessfully.";

redirect("comment.php");

}else{

$\_SESSION["errorMsg"] = "Oops! Something went wrong.";

redirect("comment.php");

}

}

?>

DeletePost.php

<?php

$searchParam = $\_GET["id"];

global $conn;

$sql = "SELECT \* FROM posts WHERE id='$searchParam'";

$stmtUpdate = $conn->query($sql);

while($data\_rows = $stmtUpdate->fetch()) {

$deleteTitle = $data\_rows['name'];

$deleteCategory = $data\_rows['category'];

$deleteImage = $data\_rows['image'];

$deleteContent = $data\_rows['content'];

}

if(isset($\_POST["Submit"])){

global $conn;

//SQL query to delete post

$sql = "DELETE FROM posts

WHERE id='$searchParam'";

$execute = $conn->query($sql);

//Check for successfull deletion to DB

if($execute){

//Deleting image of particular content

$PathDeleteImage = "upload/$deleteImage";

unlink($PathDeleteImage);

$\_SESSION["successMsg"] = "Post deleted successfully!";

redirect("posts.php");

}else{

$\_SESSION["errorMsg"] = "Oops! Something went wrong, please try

again.";

redirect("posts.php");

}

}

?>

EditPost.php

<?php

if(isset($\_POST["Submit"])){

$PostName = $\_POST["postName"];

$PostCategory = $\_POST["selectCategory"];

$PostImage = $\_FILES["image"]["name"];

$ImgTarget = "upload/".basename($\_FILES["image"]["name"]);

$PostContent = $\_POST["postContentArea"];

$Admin = "admin";

//Get current date and time

date\_default\_timezone\_set("Asia/Kathmandu");

$date = time();

$get\_time = strftime("%d-%m-%Y %H:%MM:%S",$date);

if(empty($PostName)){

//Making the error into session variaoble using super global variable

$\_SESSION["errorMsg"] = "Field(s) cannot be empty.";

redirect("posts.php");

//Check for category length

}else if(strlen($PostName)<8){

$\_SESSION["errorMsg"] = "Post title must be atleast 4 characters long.";

redirect("posts.php");

}else if(strlen($PostContent)>1199){

$\_SESSION["errorMsg"] = "Post content cannot be more than 1200 words.";

redirect("posts.php");

}else{

global $conn;

if(!empty($\_FILES["image"]["name"])){

//Update query post when validation is true and image is not empty

$sql = "UPDATE posts

SET name = '$PostName', category = '$PostCategory', image =

'$PostImage', content = '$PostContent'

WHERE id = '$searchParam' ";

}else{

$sql = "UPDATE posts

SET name = '$PostName', category = '$PostCategory', content =

'$PostContent'

WHERE id = '$searchParam' ";

}

$execute = $conn->query($sql);

//Move uplaoded file into local folder

move\_uploaded\_file($\_FILES["image"]["tmp\_name"],$ImgTarget);

//Check for successfull addition to DB

if($execute){

$\_SESSION["successMsg"] = "Post updated successfully!";

redirect("posts.php");

}else{

$\_SESSION["errorMsg"] = "Oops! Something went wrong, please try

again.";

redirect("posts.php");

}

}

}

?>

NewPost.php

<?php

if(isset($\_POST["Submit"])){

$PostName = $\_POST["postName"];

$PostCategory = $\_POST["selectCategory"];

$PostImage = $\_FILES["image"]["name"];

$ImgTarget = "upload/".basename($\_FILES["image"]["name"]);

$PostContent = $\_POST["postContentArea"];

$Admin = $\_SESSION["Name"];

//Get current date and time

date\_default\_timezone\_set("Asia/Kathmandu");

$date = time();

$get\_time = strftime("%d-%m-%Y %H:%MM:%S",$date);

if(empty($PostName)){

//Making the error into session variaoble using super global variable

$\_SESSION["errorMsg"] = "Field(s) cannot be empty.";

redirect("newpost.php");

//Check for category length

}else if(strlen($PostName)<8){

$\_SESSION["errorMsg"] = "Post title must be atleast 4 characters long.";

redirect("newpost.php");

}else if(strlen($PostContent)>1199){

$\_SESSION["errorMsg"] = "Post content cannot be more than 1200 words.";

redirect("newpost.php");

}else{

global $conn;

//Inserting title when validation is true

$sql = "INSERT into posts(datetime, name, category, author, image, content)";

//PDO named dummy paramerter to prevent sql injection

$sql .= "VALUES(:dateTime, :postName, :catName, :authorName, :imgName,

:postContent)";

//PDO object notation to call prepare mathod

$stmt = $conn->prepare($sql);

//Bind dummy values to actual values

$stmt->bindValue(':dateTime', $get\_time);

$stmt->bindValue(':postName', $PostName);

$stmt->bindValue(':catName', $PostCategory);

$stmt->bindValue(':authorName', $Admin);

$stmt->bindValue(':imgName', $PostImage);

$stmt->bindValue(':postContent', $PostContent);

//PDO ,ethod execute called via $stmt object

$execute=$stmt->execute();

//Move uplaoded file into local folder

move\_uploaded\_file($\_FILES["image"]["tmp\_name"],$ImgTarget);

//Check for successfull addition to DB

if($execute){

$\_SESSION["successMsg"] = "Post created successfully!";

redirect("newpost.php");

}else{

$\_SESSION["errorMsg"] = "Oops! Something went wrong, please try

again.";

redirect("newpost.php");

}

}

}

?>