Blood Donation Management System

Backend code with Database :-

-- Step 1: Create the Database

CREATE DATABASE blood\_donation\_system;

USE blood\_donation\_system;

-- Step 2: Create Tables

-- Donors Table

CREATE TABLE donors (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

blood\_group ENUM('A+', 'A-', 'B+', 'B-', 'O+', 'O-', 'AB+', 'AB-') NOT NULL,

contact\_number VARCHAR(15),

email VARCHAR(255) UNIQUE,

registration\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

-- Blood Requests Table

CREATE TABLE blood\_requests (

id INT AUTO\_INCREMENT PRIMARY KEY,

requester\_name VARCHAR(255) NOT NULL,

blood\_group ENUM('A+', 'A-', 'B+', 'B-', 'O+', 'O-', 'AB+', 'AB-') NOT NULL,

quantity INT NOT NULL,

request\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

-- Blood Inventory Table

CREATE TABLE blood\_inventory (

id INT AUTO\_INCREMENT PRIMARY KEY,

blood\_group ENUM('A+', 'A-', 'B+', 'B-', 'O+', 'O-', 'AB+', 'AB-') NOT NULL UNIQUE,

quantity INT NOT NULL,

last\_updated TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

-- Step 3: Insert Initial Data

-- Sample Donors

INSERT INTO donors (name, blood\_group, contact\_number, email)

VALUES

('John Doe', 'A+', '1234567890', 'john@example.com'),

('Jane Smith', 'O-', '0987654321', 'jane@example.com'),

('Alice Johnson', 'B+', '5555555555', 'alice@example.com');

-- Sample Blood Inventory

INSERT INTO blood\_inventory (blood\_group, quantity)

VALUES

('A+', 10),

('O-', 5),

('B+', 7),

('AB-', 2);

-- Step 4: Stored Procedures

-- Add Donor

DELIMITER //

CREATE PROCEDURE AddDonor(

IN donorName VARCHAR(255),

IN donorBloodGroup ENUM('A+', 'A-', 'B+', 'B-', 'O+', 'O-', 'AB+', 'AB-'),

IN donorContact VARCHAR(15),

IN donorEmail VARCHAR(255)

)

BEGIN

INSERT INTO donors (name, blood\_group, contact\_number, email)

VALUES (donorName, donorBloodGroup, donorContact, donorEmail);

END //

DELIMITER ;

-- Request Blood

DELIMITER //

CREATE PROCEDURE RequestBlood(

IN requester VARCHAR(255),

IN requestedBloodGroup ENUM('A+', 'A-', 'B+', 'B-', 'O+', 'O-', 'AB+', 'AB-'),

IN requestedQuantity INT

)

BEGIN

DECLARE availableQuantity INT;

-- Check available quantity

SELECT quantity INTO availableQuantity

FROM blood\_inventory

WHERE blood\_group = requestedBloodGroup;

-- If enough blood is available, update inventory and log request

IF availableQuantity >= requestedQuantity THEN

UPDATE blood\_inventory

SET quantity = quantity - requestedQuantity

WHERE blood\_group = requestedBloodGroup;

INSERT INTO blood\_requests (requester\_name, blood\_group, quantity)

VALUES (requester, requestedBloodGroup, requestedQuantity);

ELSE

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Insufficient blood quantity in inventory';

END IF;

END //

DELIMITER ;

-- Add Blood to Inventory

DELIMITER //

CREATE PROCEDURE AddBloodToInventory(

IN bloodGroup ENUM('A+', 'A-', 'B+', 'B-', 'O+', 'O-', 'AB+', 'AB-'),

IN quantityAdded INT

)

BEGIN

INSERT INTO blood\_inventory (blood\_group, quantity)

VALUES (bloodGroup, quantityAdded)

ON DUPLICATE KEY UPDATE quantity = quantity + quantityAdded;

END //

DELIMITER ;

-- Fetch Blood Inventory

DELIMITER //

CREATE PROCEDURE GetBloodInventory()

BEGIN

SELECT blood\_group, quantity, last\_updated

FROM blood\_inventory;

END //

DELIMITER ;

-- Step 5: Triggers

DELIMITER //

CREATE TRIGGER UpdateInventoryOnRequest

AFTER INSERT ON blood\_requests

FOR EACH ROW

BEGIN

UPDATE blood\_inventory

SET quantity = quantity - NEW.quantity

WHERE blood\_group = NEW.blood\_group;

END //

DELIMITER ;

-- Step 6: Example Queries

-- Fetch All Donors

DELIMITER //

CREATE PROCEDURE GetAllDonors()

BEGIN

SELECT id, name, blood\_group, contact\_number, email, registration\_date

FROM donors;

END //

DELIMITER ;