MINI PROJECT

BLOOD DONATION MANAGEMENT SYSTEM

AIM:

The aim of this project is to develop a Blood Donation Management System using Java and MySQL to efficiently store, manage, and display donor information.

ALGORITHM:

1) Start.
2) Establish a connection to the MySQL database using JDBC.
3) Show the main menu with options to add a donor, display donors, or exit.
4) Prompt the user for donor details (name, age, blood group, and contact number).
5) Insert the entered details into the Donors table in the database.
6) Retrieve all records from the Donors table.
7) Format and display the donor details (ID, name, age, blood group, contact number).
8) Loop back to the main menu until the user chooses to exit.
9) Based on user input, call the corresponding function (add or display donors).
10) Close the database connection and terminate the program.
11) Stop.

PROGRAM:

SQL CODE:

```
CREATE DATABASE BloodDonationDB;

USE BloodDonationDB;

CREATE TABLE Donors (

   id INT AUTO_INCREMENT PRIMARY KEY,

   name VARCHAR(100) NOT NULL,

   age INT NOT NULL,

   blood_group VARCHAR(5) NOT NULL,

   contact VARCHAR(15) NOT NULL

);
```

JAVA CODE:

```
import java.sql.*;
import java.util.Scanner;

public class BloodDonationSystem {
    static final String DB_URL = "jdbc:mysql://localhost:3306/BloodDonationDB";
    static final String USER = "root"; // Update with your MySQL username
    static final String PASS = "password"; // Update with your MySQL password

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        try (Connection conn = DriverManager.getConnection(DB_URL, USER, PASS)) {
            System.out.println("Connected to the database.");

            while (true) {
                System.out.println("\nBlood Donation System:");
                System.out.println("1. Add Donor");
                System.out.println("2. Display Donors");
                System.out.println("3. Exit");
                System.out.print("Enter your choice: ");
                int choice = scanner.nextInt();
                scanner.nextLine(); // Consume newline

                switch (choice) {
                    case 1:
                        addDonor(conn, scanner);
                        break;
                    case 2:
                        displayDonors(conn);
                        break;
                    case 3:
                        System.out.println("Exiting... Goodbye!");
                        return;
                    default:
                        System.out.println("Invalid choice. Please try again.");
                }
            }
        } catch (SQLException e) {
            System.out.println("Database error: " + e.getMessage());
        }
    }

    private static void addDonor(Connection conn, Scanner scanner) {
        try {
            System.out.print("Enter donor name: ");
            String name = scanner.nextLine();
            System.out.print("Enter donor age: ");
            int age = scanner.nextInt();

            scanner.nextLine(); // Consume newline

            System.out.print("Enter blood group (e.g., A+, O-): ");
            String bloodGroup = scanner.nextLine();

            System.out.print("Enter contact number: ");
            String contact = scanner.nextLine();

            String query = "INSERT INTO Donors (name, age, blood_group, contact) VALUES (?, ?, ?,
?)";
            try (PreparedStatement pstmt = conn.prepareStatement(query)) {
                pstmt.setString(1, name);
                pstmt.setInt(2, age);
                pstmt.setString(3, bloodGroup);
                pstmt.setString(4, contact);
                pstmt.executeUpdate();
                System.out.println("Donor added successfully.");
            }
        } catch (SQLException e) {
            System.out.println("Error adding donor: " + e.getMessage());
        }
    }

    private static void displayDonors(Connection conn) {
        String query = "SELECT * FROM Donors";
        try (Statement stmt = conn.createStatement();
             ResultSet rs = stmt.executeQuery(query)) {

            System.out.println("\nRegistered Donors:");
            System.out.printf("%-5s %-20s %-5s %-10s %-15s\n", "ID", "Name", "Age", "Blood
Group", "Contact");
            System.out.println("-------------------------------------------------------");
            while (rs.next()) {
                System.out.printf("%-5d %-20s %-5d %-10s %-15s\n",
                        rs.getInt("id"),
                        rs.getString("name"),
                        rs.getInt("age"),
                        rs.getString("blood_group"),
                        rs.getString("contact"));
            }
        } catch (SQLException e) {
            System.out.println("Error displaying donors: " + e.getMessage());
        }
    }
}
```

OUTPUT:

markdown

```
Blood Donation System:
1. Add Donor
2. Display Donors
3. Exit
Enter your choice: 1
```

mathematica

```
Enter donor name: John Doe
Enter donor age: 28
Enter blood group (e.g., A+, O-): O+
Enter contact number: 9876543210
Donor added successfully.
```

markdown

```
Blood Donation System:
1. Add Donor
2. Display Donors
3. Exit
Enter your choice: 2
```

markdown

```
Registered Donors:
ID    Name                  Age    Blood Group    Contact
------------------------------------------------------------
1     John Doe              28     O+             9876543210
```

markdown

```
Blood Donation System:
1. Add Donor
2. Display Donors
3. Exit
Enter your choice: 3


Exiting... Goodbye!
```

sql

```sql
SELECT * FROM Donors;
```

diff

```
+----+----------+-----+------------+-------------+
| id | name     | age | blood_group | contact     |
+----+----------+-----+------------+-------------+
| 1  | John Doe |  28 | O+          | 9876543210  |
+----+----------+-----+------------+-------------+
```

RESULT:

The Blood Donation Management System efficiently stores and retrieves donor information, allowing users to add new donor details and display all registered donors in a structured format. The system achieves its objective of managing blood donor data using Java and MySQL, ensuring accuracy and ease of use002E