MINI PROJECT – BLOOD DONATION FORM

DONE BY

SRIRAM U and AJAY S

AIM:

To create a form for adding details of the blood donors

DBConnection.java(Used to connect MYSQL Database)

Code:

```
import java.sql.*;

public class DBConnection {
    private static final String URL =
"jdbc:mysql://localhost:3306/blooddonation";
```

```
    private static final String USER = "root";
    private static final String PASSWORD = "1973";

    private Connection conn;

    public Connection getConnection() {
        try {
            if (conn == null || conn.isClosed()) {
                conn = DriverManager.getConnection(URL, USER, PASSWORD);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return conn;
    }

    public void closeConnection() {
        try {
            if (conn != null && !conn.isClosed()) {
                conn.close();
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

BloodDonationForm.java(Used as the form page)

Code:

```
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;

public class BloodDonationForm extends JFrame {
    private JTextField nameField, ageField, genderField, bloodGroupField,
contactField, addressField;
    private JButton submitButton;
    private DBConnection dbConnection;

    public BloodDonationForm() {
        dbConnection = new DBConnection();

        // Set frame properties
        setTitle("Blood Donation Form");
        ImageIcon icon = new ImageIcon("resources/logo.png"); // Ensure this
path is correct
        setIconImage(icon.getImage());
```

```
        // Maximize the window on startup
        setExtendedState(JFrame.MAXIMIZED_BOTH); // Fullscreen (maximized)
display
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // Create the custom background panel with image
        BackgroundPanel panel = new
BackgroundPanel("resources/background.png"); // Corrected path to image
        panel.setLayout(new GridBagLayout()); // Set GridBagLayout for form
components

        // Set Layout Manager to GridBagLayout for the panel
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.insets = new Insets(10, 10, 10, 10);

        // Create custom font
        Font labelFont = new Font("Arial", Font.BOLD, 20);
        Font fieldFont = new Font("Arial", Font.BOLD, 20);

        // Name Field
        JLabel nameLabel = new JLabel("Name:");
        nameLabel.setForeground(Color.BLACK);
        nameLabel.setFont(labelFont);
        gbc.gridx = 0;
        gbc.gridy = 0;
        gbc.anchor = GridBagConstraints.WEST; // Align label to the right
        panel.add(nameLabel, gbc);

        nameField = new JTextField(20);
```

```
nameField.setFont(fieldFont);
gbc.gridx = 1;
gbc.anchor = GridBagConstraints.WEST; // Align text box to the left
panel.add(nameField, gbc);

// Age Field
JLabel ageLabel = new JLabel("Age:");
ageLabel.setForeground(Color.BLACK);
ageLabel.setFont(labelFont);
gbc.gridx = 0;
gbc.gridy = 1;
panel.add(ageLabel, gbc);

ageField = new JTextField(20);
ageField.setFont(fieldFont);
gbc.gridx = 1;
panel.add(ageField, gbc);

// Gender Field
JLabel genderLabel = new JLabel("Gender:");
genderLabel.setForeground(Color.BLACK);
genderLabel.setFont(labelFont);
gbc.gridx = 0;
gbc.gridy = 2;
panel.add(genderLabel, gbc);

genderField = new JTextField(10);
genderField.setFont(fieldFont);
gbc.gridx = 1;
panel.add(genderField, gbc);
```

```
// Blood Group Field
JLabel bloodGroupLabel = new JLabel("Blood Group:");
bloodGroupLabel.setForeground(Color.BLACK);
bloodGroupLabel.setFont(labelFont);
gbc.gridx = 0;
gbc.gridy = 3;
panel.add(bloodGroupLabel, gbc);

bloodGroupField = new JTextField(5);
bloodGroupField.setFont(fieldFont);
gbc.gridx = 1;
panel.add(bloodGroupField, gbc);

// Contact Field
JLabel contactLabel = new JLabel("Contact:");
contactLabel.setForeground(Color.BLACK);
contactLabel.setFont(labelFont);
gbc.gridx = 0;
gbc.gridy = 4;
panel.add(contactLabel, gbc);

contactField = new JTextField(15);
contactField.setFont(fieldFont);
gbc.gridx = 1;
panel.add(contactField, gbc);

// Address Field
JLabel addressLabel = new JLabel("Address:");
addressLabel.setForeground(Color.BLACK);
```

```
        addressLabel.setFont(labelFont);
        gbc.gridx = 0;
        gbc.gridy = 5;
        panel.add(addressLabel, gbc);

        addressField = new JTextField(30);
        addressField.setFont(fieldFont);
        gbc.gridx = 1;
        panel.add(addressField, gbc);

        // Submit Button
        submitButton = new JButton("Submit Donation");
        submitButton.setFont(new Font("Arial", Font.BOLD, 16));
        submitButton.setBackground(new Color(255, 255, 255));
        submitButton.setForeground(Color.RED);
        submitButton.setFocusPainted(false);
        submitButton.setPreferredSize(new Dimension(250, 50));
        submitButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                saveDonationData();
            }
        });

        gbc.gridx = 0;
        gbc.gridy = 6;
        gbc.gridwidth = 2;
        gbc.anchor = GridBagConstraints.CENTER; // Keep the button centered
horizontally
        panel.add(submitButton, gbc);
```

```
        // Set the custom background panel as the content pane
        setContentPane(panel);

        // View Button
        JButton viewButton = new JButton("View Donors");
        viewButton.setFont(new Font("Arial", Font.BOLD, 16));
        viewButton.setBackground(new Color(255, 255, 255));
        viewButton.setForeground(Color.BLUE);
        viewButton.setFocusPainted(false);
        viewButton.setPreferredSize(new Dimension(250, 50));
        viewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                viewDonorData();
            }
        });

// Position View Button under Submit Button in GridBagLayout
        gbc.gridx = 0;
        gbc.gridy = 7;
        gbc.gridwidth = 2;
        gbc.anchor = GridBagConstraints.CENTER; // Keep the button centered
horizontally
        panel.add(viewButton, gbc);
    }

    private void saveDonationData() {
        String name = nameField.getText();
        int age = Integer.parseInt(ageField.getText());
        String gender = genderField.getText();
        String bloodGroup = bloodGroupField.getText();
```

```
        String contact = contactField.getText();
        String address = addressField.getText();

        String query = "INSERT INTO donors (name, age, gender, blood_group,
contact, address, last_donated) VALUES (?, ?, ?, ?, ?, ?, NOW())";

        try (Connection conn = dbConnection.getConnection();
             PreparedStatement stmt = conn.prepareStatement(query)) {

            stmt.setString(1, name);
            stmt.setInt(2, age);
            stmt.setString(3, gender);
            stmt.setString(4, bloodGroup);
            stmt.setString(5, contact);
            stmt.setString(6, address);

            int rowsAffected = stmt.executeUpdate();
            if (rowsAffected > 0) {
                JOptionPane.showMessageDialog(this, "Donation Information
Saved Successfully.");
            } else {
                JOptionPane.showMessageDialog(this, "Error saving donation
information.");
            }
        } catch (SQLException e) {
            e.printStackTrace();
            JOptionPane.showMessageDialog(this, "Database error: " +
e.getMessage());
        }
    }
```

```
    // Custom JPanel class to draw the background image
    class BackgroundPanel extends JPanel {
        private Image backgroundImage;

        public BackgroundPanel(String imagePath) {
            // Load the image using relative path
            backgroundImage = new ImageIcon(imagePath).getImage();
        }

        @Override
        protected void paintComponent(Graphics g) {
            super.paintComponent(g);
            // Draw the background image, scaling it to fit the panel size
            g.drawImage(backgroundImage, 0, 0, getWidth(), getHeight(), this);
        }
    }
    private void viewDonorData() {
        String query = "SELECT name, age, gender, blood_group, contact,
address FROM donors";

        // Create a frame to show donor details
        JFrame viewFrame = new JFrame("Donor Details");
        viewFrame.setSize(900, 600);  // Adjust the size
        viewFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        viewFrame.setLocationRelativeTo(this);  // Center the window

        // Set a background color or image for the frame
        viewFrame.getContentPane().setBackground(new Color(245, 245, 245));
// Light gray background
```

```
        // Create the table model and JTable
        DefaultTableModel model = new DefaultTableModel();
        JTable donorTable = new JTable(model);

        // Customize the table appearance
        donorTable.setFont(new Font("Arial", Font.PLAIN, 14)); // Set font for
table text
        donorTable.setRowHeight(30);  // Set row height
        donorTable.setSelectionBackground(new Color(100, 149, 237));  //
Light blue selection color
        donorTable.setSelectionForeground(Color.WHITE);  // White text for
selected rows
        donorTable.setGridColor(Color.DARK_GRAY);  // Color for table grid
lines
        donorTable.setShowGrid(true);  // Show grid lines between cells
        donorTable.setAutoResizeMode(JTable.AUTO_RESIZE_ALL_COLUMNS);
// Auto resize columns

        // Add columns to the table
        model.addColumn("Name");
        model.addColumn("Age");
        model.addColumn("Gender");
        model.addColumn("Blood Group");
        model.addColumn("Contact");
        model.addColumn("Address");

        // Query database and populate the table
        try (Connection conn = dbConnection.getConnection();
            Statement stmt = conn.createStatement();
```

```
             ResultSet rs = stmt.executeQuery(query)) {

            while (rs.next()) {
                // Add each row from the database into the table
                model.addRow(new Object[] {
                        rs.getString("name"),
                        rs.getInt("age"),
                        rs.getString("gender"),
                        rs.getString("blood_group"),
                        rs.getString("contact"),
                        rs.getString("address")
                });
            }
        } catch (SQLException e) {
            e.printStackTrace();
            JOptionPane.showMessageDialog(this, "Error fetching donor data: "
+ e.getMessage());
        }

        // Add the table to a scroll pane and set it in the frame
        JScrollPane scrollPane = new JScrollPane(donorTable);
        viewFrame.add(scrollPane, BorderLayout.CENTER);

        // Add a header label for the table (optional)
        JLabel headerLabel = new JLabel("Donor Information",
SwingConstants.CENTER);
        headerLabel.setFont(new Font("Arial", Font.BOLD, 24));
        headerLabel.setForeground(new Color(60, 60, 60));  // Dark gray color
        headerLabel.setBorder(BorderFactory.createEmptyBorder(20, 0, 10, 0));
// Add some padding
```

```
        viewFrame.add(headerLabel, BorderLayout.NORTH);

        // Make the view window visible
        viewFrame.setVisible(true);
    }
}
```

BloodDonationApp.java(which connects form page and Database)

Code:

```
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;

public class BloodDonationForm extends JFrame {
    private JTextField nameField, ageField, genderField, bloodGroupField,
contactField, addressField;
    private JButton submitButton;
    private DBConnection dbConnection;
```

```
    public BloodDonationForm() {
        dbConnection = new DBConnection();

        // Set frame properties
        setTitle("Blood Donation Form");
        ImageIcon icon = new ImageIcon("resources/logo.png"); // Ensure this
path is correct
        setIconImage(icon.getImage());

        // Maximize the window on startup
        setExtendedState(JFrame.MAXIMIZED_BOTH); // Fullscreen (maximized)
display
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // Create the custom background panel with image
        BackgroundPanel panel = new
BackgroundPanel("resources/background.png"); // Corrected path to image
        panel.setLayout(new GridBagLayout()); // Set GridBagLayout for form
components

        // Set Layout Manager to GridBagLayout for the panel
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.insets = new Insets(10, 10, 10, 10);

        // Create custom font
        Font labelFont = new Font("Arial", Font.BOLD, 20);
        Font fieldFont = new Font("Arial", Font.BOLD, 20);

        // Name Field
        JLabel nameLabel = new JLabel("Name:");
```

```
nameLabel.setForeground(Color.BLACK);
nameLabel.setFont(labelFont);
gbc.gridx = 0;
gbc.gridy = 0;
gbc.anchor = GridBagConstraints.WEST; // Align label to the right
panel.add(nameLabel, gbc);

nameField = new JTextField(20);
nameField.setFont(fieldFont);
gbc.gridx = 1;
gbc.anchor = GridBagConstraints.WEST; // Align text box to the left
panel.add(nameField, gbc);

// Age Field
JLabel ageLabel = new JLabel("Age:");
ageLabel.setForeground(Color.BLACK);
ageLabel.setFont(labelFont);
gbc.gridx = 0;
gbc.gridy = 1;
panel.add(ageLabel, gbc);

ageField = new JTextField(20);
ageField.setFont(fieldFont);
gbc.gridx = 1;
panel.add(ageField, gbc);

// Gender Field
JLabel genderLabel = new JLabel("Gender:");
genderLabel.setForeground(Color.BLACK);
genderLabel.setFont(labelFont);
```

```
gbc.gridx = 0;
gbc.gridy = 2;
panel.add(genderLabel, gbc);

genderField = new JTextField(10);
genderField.setFont(fieldFont);
gbc.gridx = 1;
panel.add(genderField, gbc);

// Blood Group Field
JLabel bloodGroupLabel = new JLabel("Blood Group:");
bloodGroupLabel.setForeground(Color.BLACK);
bloodGroupLabel.setFont(labelFont);
gbc.gridx = 0;
gbc.gridy = 3;
panel.add(bloodGroupLabel, gbc);

bloodGroupField = new JTextField(5);
bloodGroupField.setFont(fieldFont);
gbc.gridx = 1;
panel.add(bloodGroupField, gbc);

// Contact Field
JLabel contactLabel = new JLabel("Contact:");
contactLabel.setForeground(Color.BLACK);
contactLabel.setFont(labelFont);
gbc.gridx = 0;
gbc.gridy = 4;
panel.add(contactLabel, gbc);
```

```
contactField = new JTextField(15);
contactField.setFont(fieldFont);
gbc.gridx = 1;
panel.add(contactField, gbc);

// Address Field
JLabel addressLabel = new JLabel("Address:");
addressLabel.setForeground(Color.BLACK);
addressLabel.setFont(labelFont);
gbc.gridx = 0;
gbc.gridy = 5;
panel.add(addressLabel, gbc);

addressField = new JTextField(30);
addressField.setFont(fieldFont);
gbc.gridx = 1;
panel.add(addressField, gbc);

// Submit Button
submitButton = new JButton("Submit Donation");
submitButton.setFont(new Font("Arial", Font.BOLD, 16));
submitButton.setBackground(new Color(255, 255, 255));
submitButton.setForeground(Color.RED);
submitButton.setFocusPainted(false);
submitButton.setPreferredSize(new Dimension(250, 50));
submitButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        saveDonationData();
    }
});
```

```
        gbc.gridx = 0;
        gbc.gridy = 6;
        gbc.gridwidth = 2;
        gbc.anchor = GridBagConstraints.CENTER; // Keep the button centered
horizontally
        panel.add(submitButton, gbc);

        // Set the custom background panel as the content pane
        setContentPane(panel);

        // View Button
        JButton viewButton = new JButton("View Donors");
        viewButton.setFont(new Font("Arial", Font.BOLD, 16));
        viewButton.setBackground(new Color(255, 255, 255));
        viewButton.setForeground(Color.BLUE);
        viewButton.setFocusPainted(false);
        viewButton.setPreferredSize(new Dimension(250, 50));
        viewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                viewDonorData();
            }
        });

// Position View Button under Submit Button in GridBagLayout
        gbc.gridx = 0;
        gbc.gridy = 7;
        gbc.gridwidth = 2;
        gbc.anchor = GridBagConstraints.CENTER; // Keep the button centered
horizontally
```

```
        panel.add(viewButton, gbc);
    }

    private void saveDonationData() {
        String name = nameField.getText();
        int age = Integer.parseInt(ageField.getText());
        String gender = genderField.getText();
        String bloodGroup = bloodGroupField.getText();
        String contact = contactField.getText();
        String address = addressField.getText();

        String query = "INSERT INTO donors (name, age, gender, blood_group,
contact, address, last_donated) VALUES (?, ?, ?, ?, ?, ?, NOW())";

        try (Connection conn = dbConnection.getConnection();
             PreparedStatement stmt = conn.prepareStatement(query)) {

            stmt.setString(1, name);
            stmt.setInt(2, age);
            stmt.setString(3, gender);
            stmt.setString(4, bloodGroup);
            stmt.setString(5, contact);
            stmt.setString(6, address);

            int rowsAffected = stmt.executeUpdate();
            if (rowsAffected > 0) {
                JOptionPane.showMessageDialog(this, "Donation Information
Saved Successfully.");
            } else {
                JOptionPane.showMessageDialog(this, "Error saving donation
```

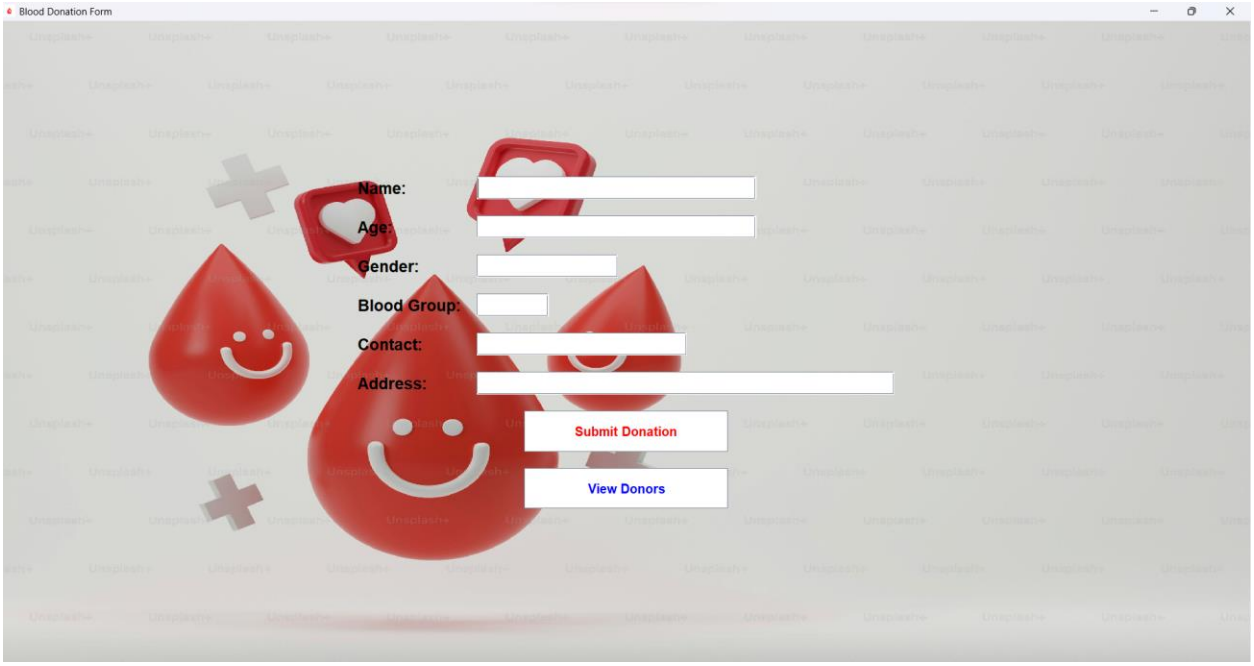
```
information.");
            }
        } catch (SQLException e) {
            e.printStackTrace();
            JOptionPane.showMessageDialog(this, "Database error: " +
e.getMessage());
        }
    }

    // Custom JPanel class to draw the background image
    class BackgroundPanel extends JPanel {
        private Image backgroundImage;

        public BackgroundPanel(String imagePath) {
            // Load the image using relative path
            backgroundImage = new ImageIcon(imagePath).getImage();
        }

        @Override
        protected void paintComponent(Graphics g) {
            super.paintComponent(g);
            // Draw the background image, scaling it to fit the panel size
            g.drawImage(backgroundImage, 0, 0, getWidth(), getHeight(), this);
        }
    }
    private void viewDonorData() {
        String query = "SELECT name, age, gender, blood_group, contact,
address FROM donors";

        // Create a frame to show donor details
```

```
        JFrame viewFrame = new JFrame("Donor Details");
        viewFrame.setSize(900, 600);  // Adjust the size
        viewFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        viewFrame.setLocationRelativeTo(this);  // Center the window

        // Set a background color or image for the frame
        viewFrame.getContentPane().setBackground(new Color(245, 245, 245));
// Light gray background

        // Create the table model and JTable
        DefaultTableModel model = new DefaultTableModel();
        JTable donorTable = new JTable(model);

        // Customize the table appearance
        donorTable.setFont(new Font("Arial", Font.PLAIN, 14)); // Set font for
table text
        donorTable.setRowHeight(30);  // Set row height
        donorTable.setSelectionBackground(new Color(100, 149, 237));  //
Light blue selection color
        donorTable.setSelectionForeground(Color.WHITE);  // White text for
selected rows
        donorTable.setGridColor(Color.DARK_GRAY);  // Color for table grid
lines
        donorTable.setShowGrid(true);  // Show grid lines between cells
        donorTable.setAutoResizeMode(JTable.AUTO_RESIZE_ALL_COLUMNS);
// Auto resize columns

        // Add columns to the table
        model.addColumn("Name");
        model.addColumn("Age");
```

```
        model.addColumn("Gender");
        model.addColumn("Blood Group");
        model.addColumn("Contact");
        model.addColumn("Address");

        // Query database and populate the table
        try (Connection conn = dbConnection.getConnection();
             Statement stmt = conn.createStatement();
             ResultSet rs = stmt.executeQuery(query)) {

            while (rs.next()) {
                // Add each row from the database into the table
                model.addRow(new Object[] {
                        rs.getString("name"),
                        rs.getInt("age"),
                        rs.getString("gender"),
                        rs.getString("blood_group"),
                        rs.getString("contact"),
                        rs.getString("address")
                });
            }
        } catch (SQLException e) {
            e.printStackTrace();
            JOptionPane.showMessageDialog(this, "Error fetching donor data: "
+ e.getMessage());
        }

        // Add the table to a scroll pane and set it in the frame
        JScrollPane scrollPane = new JScrollPane(donorTable);
        viewFrame.add(scrollPane, BorderLayout.CENTER);
```

```
        // Add a header label for the table (optional)
        JLabel headerLabel = new JLabel("Donor Information",
SwingConstants.CENTER);
        headerLabel.setFont(new Font("Arial", Font.BOLD, 24));
        headerLabel.setForeground(new Color(60, 60, 60));  // Dark gray color
        headerLabel.setBorder(BorderFactory.createEmptyBorder(20, 0, 10, 0));
// Add some padding
        viewFrame.add(headerLabel, BorderLayout.NORTH);

        // Make the view window visible
        viewFrame.setVisible(true);
    }
}
```

OUTPUT:



Donor Details

Donor Information

Name	Age	Gender	Blood Group	Contact	Address
janit	18	M	B+	7339533382	college
sriram	19	male	o	9600791354	46/2 ramnagar
Ram	18	Male	A+	123456789	College-1