Appendix 2

SOURCE CODE FOR APPLICATION

Java Swing was used to develop the gui therefore the code contains extensive usage of predefined methods

homepage.java

```
import java.awt.*

import javax.swing.*


public class Homepage extends JFrame {


    public static String s=LoginPage.username;

    private JPanel contentPane;


    /**
     * Launch the application.
     */

    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Homepage frame = new Homepage();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the frame.
     */
    public Homepage() {
        pack();
        setLocationRelativeTo(null);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 777, 512);
        this.setPreferredSize(new Dimension(777, 512));
        contentPane = new JPanel();
        contentPane.setBackground(new Color(135, 206, 235));
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);
```

```
             JLabel homepageLbl = new JLabel("Homepage");
             homepageLbl.setFont(new Font("Rockwell", Font.BOLD, 32));
             homepageLbl.setBounds(284, 47, 186, 54);
             contentPane.add(homepageLbl);

             JButton btnNewButton = new JButton("Reciever");
             btnNewButton.setBackground(new Color(255, 228, 225));
             btnNewButton.setFont(new Font("Rockwell", Font.PLAIN, 30));
             btnNewButton.addActionListener(new ActionListener() {
                    public void actionPerformed(ActionEvent e) {
                          recieverHomepage lr=new recieverHomepage();
                          lr.setVisible(true);
                          dispose();



                    }
             });
             btnNewButton.setBounds(63, 167, 274, 237);
             contentPane.add(btnNewButton);

             JButton btnDonor = new JButton("Donor");
             btnDonor.setBackground(new Color(255, 228, 225));
             btnDonor.addActionListener(new ActionListener() {
                    public void actionPerformed(ActionEvent e) {


                          LoginPage lp=new LoginPage();

                            lp.setVisible(true);

                          dispose();


                    }
             });
             btnDonor.setFont(new Font("Rockwell", Font.PLAIN, 30));
             btnDonor.setBounds(415, 167, 274, 237);
             contentPane.add(btnDonor);
       }
}
```

LoginPage.java

```
import java.awt.*
import java.sql.*
import java.util.logging.Level;
import java.util.logging.Logger;

import javax.swing.*

public class LoginPage extends My_CNX {


    public JFrame frame;
    public JPanel contentPane;
    public static String username;
    private JTextField usernameField;
    private JPasswordField passwordField;
    //public static String s=LoginPage.username;
    /**
     * Launch the application.
     */
    public static void main(String[] args) {
               LoginPage window = new LoginPage();

        window.setVisible(true);

    }
    /**
     * Create the application.
     */
    public LoginPage() { // this.username=s;

        initialize();

    }

    /**
     * Initialize the contents of the frame.
     */
    private void initialize() {


        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 700, 622);
        this.setPreferredSize(new Dimension(700, 600));
        this.pack();
        contentPane = new JPanel();
        contentPane.setBackground(new Color(135, 206, 235));
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel usernameLbl = new JLabel("Username");
        usernameLbl.setFont(new Font("Rockwell", Font.PLAIN, 22));
        usernameLbl.setBounds(56, 123, 136, 53);
        //frame.getContentPane().add(usernameLbl);
        contentPane.add(usernameLbl);

        JLabel passwordLbl = new JLabel("Password");
        passwordLbl.setFont(new Font("Rockwell", Font.PLAIN, 22));
        passwordLbl.setBounds(56, 222, 136, 53);
        //frame.getContentPane().add(passwordLbl);
        contentPane.add(passwordLbl);

        usernameField = new JTextField();
        usernameField.setFont(new Font("Rockwell", Font.PLAIN, 20));
        usernameField.setBackground(new Color(255, 240, 245));
        usernameField.addFocusListener(new FocusAdapter() {

        });
        usernameField.setBounds(223, 132, 300, 46);
        //frame.getContentPane().add(usernameField);
        usernameField.setColumns(10);
        contentPane.add(usernameField);
```

```
        JLabel welcomeLbl = new JLabel("Welcome, Please Login In");
        welcomeLbl.setFont(new Font("Rockwell", Font.BOLD, 30));
        welcomeLbl.setHorizontalAlignment(SwingConstants.CENTER);
        welcomeLbl.setBounds(109, 28, 466, 67);
        //frame.getContentPane().add(welcomeLbl);
        contentPane.add(welcomeLbl);

        JButton loginBtn = new JButton("Login");
        loginBtn.setBackground(new Color(255, 228, 225));
        loginBtn.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                PreparedStatement st;
                ResultSet rs;

                //getting username and password

                username = usernameField.getText();
                String password = String.valueOf(passwordField.~~getText()~~);

                //string query to check whether username and password exist in
database
                String query = "SELECT * FROM `users` WHERE `username` = ? AND
`password`= ?";

                if (username.trim().equals("username")) {
                    JOptionPane.showMessageDialog(null, "Enter Your Username",
"Empty Username", 2);
                } else if (password.trim().equals("password")) {
                    JOptionPane.showMessageDialog(null, "Enter Your Password",
"Empty Password", 2);
                } else {


                    try {
                        st = My_CNX.getConnection().prepareStatement(query);

                        st.setString(1, username);
                        st.setString(2, password);

                        rs = st.executeQuery();

                        if (rs.next()) {

                            DonorsPage page = new DonorsPage();
                            page.setVisible(true);
                            page.pack();
                            page.setLocationRelativeTo(null);
                            page.setBounds(289, 329, 800, 800);
                            dispose();



                        } else {
                            JOptionPane.showMessageDialog(null, "Invalid
username/password", "login error", 2);
                        }
                    } catch (SQLException ex) {
                        Logger.getLogger(" Get Connection -> " +
My_CNX.class.getName()).log(Level.SEVERE, null, ex);
                    }
                }
            }
        });
        loginBtn.setFont(new Font("Rockwell", Font.PLAIN, 20));
        loginBtn.setBounds(289, 329, 136, 46);
        //frame.getContentPane().add(loginBtn);
        contentPane.add(loginBtn);
        passwordField = new JPasswordField();
        passwordField.setBackground(new Color(255, 240, 245));
        passwordField.setFont(new Font("Rockwell", Font.PLAIN, 20));
        passwordField.setBounds(223, 227, 300, 46);
        //frame.getContentPane().add(passwordField);
        contentPane.add(passwordField);

        JButton navRegBtn = new JButton("Don't have an account?, Click here to
register");
        navRegBtn.setBackground(new Color(255, 228, 225));
        navRegBtn.setFont(new Font("Rockwell", Font.PLAIN, 20));
        navRegBtn.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent ae) {


                dispose();
                Registration r = new Registration();
                r.setVisible(true);
            }
        });

        navRegBtn.setFont(new Font("Tahoma", Font.PLAIN, 18));
        navRegBtn.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {}
        });
        navRegBtn.setBounds(43, 428, 606, 87);
        //frame.getContentPane().add(navRegBtn);
        contentPane.add(navRegBtn);

        JButton btnNewButton = new JButton("< back");
        btnNewButton.setBackground(new Color(255, 228, 225));
        btnNewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                Homepage hp = new Homepage();
                hp.setVisible(true);
                dispose();


            }
        });
        btnNewButton.setBounds(31, 56, 80, 23);
        contentPane.add(btnNewButton);
        //frame.add(contentPane);
    }
}
```

Registration.java

```
import java.sql.*
import java.util.*
import javax.swing.*
import java.awt.*


public class Registration extends JFrame {

    private JPanel contentPane;
    private JTextField textField_username;
    private JTextField textField_mobNo;
    private JTextField textField_fullname;
    private JPasswordField passwordField_1;
    private JPasswordField passwordField_2;
    ButtonGroup G1;
    public static String s = LoginPage.username;
    // int=LoginPage.id;



    public boolean verifyFields() {

        String fname = textField_fullname.getText();
        String uname = textField_username.getText();
        String phone = textField_mobNo.getText();
        String pass1 = String.valueOf(passwordField_1.getPassword());
        String pass2 = String.valueOf(passwordField_2.getPassword());

        if (fname.trim().equals("") || uname.trim().equals("") ||
phone.trim().equals("") || pass1.trim().equals("") || pass2.trim().equals("")) {
            JOptionPane.showMessageDialog(null, "One or more fields are empty",
"Empty Fields", 2);
            return false;
        } else if (!pass1.equals(pass2)) {
            JOptionPane.showMessageDialog(null, "Passwords do not Match",
"Password Mismatch", 2);
            return false;
        }



        //if everything is correct
        else {
            return true;
        }

    }

    public boolean checkUsername(String username) {

        PreparedStatement st;
        ResultSet rs;
        boolean username_exist = false;

        String query = "SELECT * FROM `users` WHERE `username` = ?";

        try {
            st = My_CNX.getConnection().prepareStatement(query);
            st.setString(1, username);
            rs = st.executeQuery();
            
            if (rs.next()) {
                username_exist = true;
                JOptionPane.showMessageDialog(null, "This username is already 
taken, choose another Username", "Username Creation Failure ", 2);
            }
            
            
        } catch (SQLException e) {
            Logger.getLogger(Registration.class.getName()).log(Level.SEVERE, null, 
e);
        }
        
        return username_exist;
        
    }
    
    
    
    //create a function to check
    
    //
    
    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Registration frame = new Registration();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the frame.
     */
    public Registration() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 700, 730);
        this.setPreferredSize(new Dimension(700, 730));
        this.pack();
        contentPane = new JPanel();
        contentPane.setBackground(new Color(135, 206, 250));
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);
        
        JLabel lblRegistration = new JLabel("Registration");
        lblRegistration.setFont(new Font("Rockwell", Font.PLAIN, 32));
        lblRegistration.setBounds(245, 30, 182, 47);
```

```
contentPane.add(lblRegistration);

JLabel crUsernameLbl = new JLabel("Username");
crUsernameLbl.setFont(new Font("Rockwell", Font.PLAIN, 18));
crUsernameLbl.setBounds(51, 187, 114, 40);
contentPane.add(crUsernameLbl);

JLabel crPasswordLbl = new JLabel("Create Password");
crPasswordLbl.setFont(new Font("Rockwell", Font.PLAIN, 18));
crPasswordLbl.setBounds(49, 226, 175, 40);
contentPane.add(crPasswordLbl);

JRadioButton radiobMale = new JRadioButton("Male");
radiobMale.setBackground(new Color(135, 206, 250));
radiobMale.setFont(new Font("Tahoma", Font.PLAIN, 19));
radiobMale.setBounds(442, 406, 89, 25);
contentPane.add(radiobMale);

JRadioButton radiobFemale = new JRadioButton("Female");
radiobFemale.setBackground(new Color(135, 206, 250));
radiobFemale.setSelected(true);
radiobFemale.setFont(new Font("Tahoma", Font.PLAIN, 19));
radiobFemale.setBounds(265, 406, 127, 25);
contentPane.add(radiobFemale);

JLabel genderLbl = new JLabel("Gender");
genderLbl.setFont(new Font("Rockwell", Font.PLAIN, 18));
genderLbl.setBounds(49, 405, 114, 40);
contentPane.add(genderLbl);

JLabel mobNumberLbl = new JLabel("Mobile Number");
mobNumberLbl.setFont(new Font("Rockwell", Font.PLAIN, 18));
mobNumberLbl.setBounds(49, 330, 161, 40);
contentPane.add(mobNumberLbl);

textField_username = new JTextField();
textField_username.setBackground(new Color(255, 240, 245));
textField_username.setFont(new Font("Rockwell", Font.PLAIN, 18));
textField_username.setColumns(10);
textField_username.setBounds(265, 192, 263, 35);
contentPane.add(textField_username);

textField_mobNo = new JTextField();
textField_mobNo.setBackground(new Color(255, 240, 245));
textField_mobNo.setFont(new Font("Rockwell", Font.PLAIN, 18));
textField_mobNo.addKeyListener(new KeyAdapter() {
    @Override
    public void keyTyped(KeyEvent e) {

        if (!Character.isDigit(e.getKeyChar())) {
            e.consume();

        }
    }
});
textField_mobNo.setColumns(10);
textField_mobNo.setBounds(265, 335, 263, 35);
contentPane.add(textField_mobNo);
```

```
        JButton btnNewButton = new JButton("Submit and Create account");
        btnNewButton.setBackground(new Color(255, 228, 225));
        btnNewButton.setFont(new Font("Rockwell", Font.PLAIN, 18));
        btnNewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent ae) {



                String fname = textField_fullname.getText();
                String username = textField_username.getText();
                String pass1 = String.valueOf(passwordField_1.getPassword());
                String pass2 = String.valueOf(passwordField_2.getPassword());
                String phone = textField_mobNo.getText();
                String gender = "Male";

                if (radiobFemale.isSelected()) {

                    gender = "Female";


                }

                if (verifyFields()) {

                    if (!checkUsername(username)) {
                        PreparedStatement ps;
                        ResultSet rs;
                        String registerUserQuery = "INSERT INTO
`users`(`full_name`, `username`, `password`, `phone`, `gender`) VALUES
(?,?,?,?,?)";

                        try {

                            ps =
My_CNX.getConnection().prepareStatement(registerUserQuery);
                            ps.setString(1, fname);
                            ps.setString(2, username);
                            ps.setString(3, pass1);
                            ps.setString(4, phone);
                            ps.setString(5, gender);

                            if (ps.executeUpdate() != 0) {
                                JOptionPane.showMessageDialog(null, "Your account
has been created succesfully");
                                LoginPage l = new LoginPage();
                                l.frame.setVisible(true);
                                dispose();



                            } else {
                                JOptionPane.showMessageDialog(null, "Error; Check
your details");
                            }

                        } catch (SQLException e) {


Logger.getLogger(Registration.class.getName()).log(Level.SEVERE, null, e);
```

```
                }


            }

        }

    }
});



btnNewButton.setBounds(210, 491, 300, 51);
contentPane.add(btnNewButton);

JLabel lblFullname = new JLabel("Fullname");
lblFullname.setFont(new Font("Rockwell", Font.PLAIN, 18));
lblFullname.setBounds(51, 137, 114, 40);
contentPane.add(lblFullname);

textField_fullname = new JTextField();
textField_fullname.setBackground(new Color(255, 240, 245));
textField_fullname.setFont(new Font("Rockwell", Font.PLAIN, 18));
textField_fullname.setColumns(10);
textField_fullname.setBounds(265, 147, 263, 35);
contentPane.add(textField_fullname);

JLabel lblConfirmPassword = new JLabel("Confirm Password");
lblConfirmPassword.setFont(new Font("Rockwell", Font.PLAIN, 18));
lblConfirmPassword.setBounds(51, 280, 159, 40);
contentPane.add(lblConfirmPassword);

passwordField_1 = new JPasswordField();
passwordField_1.setBackground(new Color(255, 240, 245));
passwordField_1.setFont(new Font("Rockwell", Font.PLAIN, 18));
passwordField_1.setBounds(265, 237, 263, 35);
contentPane.add(passwordField_1);

passwordField_2 = new JPasswordField();
passwordField_2.setBackground(new Color(255, 240, 245));
passwordField_2.setFont(new Font("Rockwell", Font.PLAIN, 18));
passwordField_2.setBounds(265, 291, 263, 35);
contentPane.add(passwordField_2);


ButtonGroup bg = new ButtonGroup();
bg.add(radiobMale);
bg.add(radiobFemale);

JButton btnNewButton_1 = new JButton("Click here to Login");
btnNewButton_1.setBackground(new Color(255, 228, 225));
btnNewButton_1.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {



        LoginPage l = new LoginPage();
        l.setVisible(true);
```

```
                dispose();



            }

        });
        btnNewButton_1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {


            }
        });
        btnNewButton_1.setFont(new Font("Rockwell", Font.PLAIN, 18));
        btnNewButton_1.setBounds(94, 592, 517, 63);
        contentPane.add(btnNewButton_1);






    }
}
```

DonorsPage.java

```
import java.sql.*
import java.util.*
import javax.swing.*
import java.awt.*

public class DonorsPage extends JFrame {

    private JPanel contentPane;
    private JTable table;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {


        DonorsPage frame = new DonorsPage();
        frame.setTitle("Donations Type");
        frame.setVisible(true);
    }

    /**
     * Create the frame.
     */
    public DonorsPage() {

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 600, 595);
             this.setSize(new Dimension(600, 595));
        contentPane = new JPanel();
        contentPane.setBackground(new Color(135, 206, 235));
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);
        contentPane.setSize(new Dimension(600, 595));
        JButton btnNewButton = new JButton("Make a Donation");
        btnNewButton.setBackground(new Color(255, 228, 225));
        btnNewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                DonationForm df = new DonationForm();
                df.setVisible(true);
                dispose();

            }
        });
        btnNewButton.setFont(new Font("Rockwell", Font.PLAIN, 24));
        btnNewButton.setBounds(125, 196, 457, 91);
        contentPane.add(btnNewButton);

        JButton btnDonationRequests = new JButton("Donation Requests");
        btnDonationRequests.setBackground(new Color(255, 228, 225));
        btnDonationRequests.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                Connection con = My_CNX.getConnection();
                ResultSet rs;
                PreparedStatement stmt;
                int a1 = 0, a2 = 0, a3 = 0, a4 = 0, a5 = 0;
                String reQ = "Select * from total";
                try {
                    stmt = con.prepareStatement(reQ);
                    rs = stmt.executeQuery();
                    while (rs.next()) {
                        a1 = rs.getInt(2);
                        a2 = rs.getInt(3);
                        a3 = rs.getInt(4);
                        a4 = rs.getInt(5);
                        a5 = rs.getInt(6);

                    }
                    if (a1 == 0) {
                        JOptionPane.showMessageDialog(null, "Blankets are
Required");
                    } else if (a2 == 0) {
                        JOptionPane.showMessageDialog(null, "Meals are Required");
                    } else if (a3 == 0) {
                        JOptionPane.showMessageDialog(null, "Dry Rations are
Required");
                    } else if (a4 == 0) {
                        JOptionPane.showMessageDialog(null, "Shoes are Required");
                    } else if (a5 == 0) {
                        JOptionPane.showMessageDialog(null, "Clothes are
Required");
                    } else {
                        JOptionPane.showMessageDialog(null, "All Items are
sufficient");
                    }
                } catch (SQLException ex) {
                    System.out.println(ex);
                }
            }
        });
        btnDonationRequests.setFont(new Font("Rockwell", Font.PLAIN, 20));
        btnDonationRequests.setBounds(125, 308, 457, 91);
        contentPane.add(btnDonationRequests);

        JButton btnNewButton_1_1 = new JButton("Top Donor");
        btnNewButton_1_1.setBackground(new Color(255, 228, 225));
        btnNewButton_1_1.setFont(new Font("Rockwell", Font.PLAIN, 24));
        btnNewButton_1_1.addActionListener(new ActionListener() {
            public void
            actionPerformed(ActionEvent e) {
                Connection con = My_CNX.getConnection();
                ResultSet rs, rs1;
                int c = 0, c1 = 0;
                String g = null;
                try {
                    String f = "select
id,username,(MAX(Blankets+Meals+Shoes+Clothes+Dry_Rations)) from donations";
                    PreparedStatement stmt = con.prepareStatement(f);
                    rs = stmt.executeQuery();
                    while (rs.next()) {
                        c = rs.getInt(1);
                        g = rs.getString(2);

                    }
                    String topQ = "Select
SUM(Blankets+Meals+Shoes+Clothes+Dry_Rations) from donations where id=" + c;
                    PreparedStatement stmt1 = con.prepareStatement(topQ);
                    rs1 = stmt1.executeQuery();
                    while (rs1.next()) {
                        c1 = rs1.getInt(1);
                    }
                    JOptionPane.showMessageDialog(null, "Username of Top Donor is
" + g + " and " + "Total Donations by Top Donor is " + c1 + " Items");

                } catch (SQLException ex) {
                    System.out.println(ex);
                }

            }
        });
        btnNewButton_1_1.setFont(new Font("Tahoma", Font.PLAIN, 20));
        btnNewButton_1_1.setBounds(125, 410, 457, 91);
        contentPane.add(btnNewButton_1_1);

        JLabel lblNewLabel = new JLabel("Welcome to Care n Share");
        lblNewLabel.setFont(new Font("Rockwell", Font.BOLD, 28));
        lblNewLabel.setBounds(225, 58, 349, 91);
        contentPane.add(lblNewLabel);


    }
}
```

DonationForm.java

```
import java.sql.*
```

```java
import java.util.*
import javax.swing.*
import java.awt.*

public class DonationForm extends My_CNX {
    private JTextField textField_2;
    private JTextField textField_Rate;
    private JTextField textField_cost;
    int n;
    int cost;
    String value, q, query, t_q;
    int x;
    int quantity, id;
    private JTextField textField;
    int value2;
    String i_d;
    public static String s = LoginPage.username;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {

        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    DonationForm df = new DonationForm();

                    // f.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
```

```
};



public DonationForm() {


    super();
    getContentPane().setBackground(new Color(135, 206, 235));
    System.out.print(s);

    setBounds(100, 100, 700, 660);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    getContentPane().setLayout(null);

    JButton submitBtn = new JButton("Submit");
    submitBtn.setBackground(new Color(255, 240, 245));
    submitBtn.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent event) {
            int pt = 17;
            int tb = 0, td = 0, tm = 0, ts = 0, tc = 0;
            boolean usernameExists = false;
            try {

                String SQL1 = "Select * from total where id=" + pt;
                Connection con = My_CNX.getConnection();
                PreparedStatement st = con.prepareStatement(SQL1);
                ResultSet rs1 = st.executeQuery();

                while (rs1.next()) {
                    tb = rs1.getInt("T_Blankets");
                    tm = rs1.getInt("T_Meals");
                    td = rs1.getInt("T_DryRations");
                    ts = rs1.getInt("T_Shoes");
                    tc = rs1.getInt("T_Clothes");
                    }



                    PreparedStatement pst = My_CNX.getConnection()
                        .prepareStatement("select * from donations where username
= ?");
                    pst.setString(1, s);
                    ResultSet r1 = pst.executeQuery();
                    if (r1.next()) {
                        usernameExists = true;
                        JOptionPane.showMessageDialog(null, "You are a Registered
Donor");

                    } else {
                        JOptionPane.showMessageDialog(null,
                            "You are new here? kindly press enter button to be
Registered Donor");
                    }
                } catch (Exception e) {
                    JOptionPane.showMessageDialog(null, e);
                }

                PreparedStatement st, ps, st1;
                Connection con;
                ResultSet r, r1;
                int rs, t_r;

                query = "select * from donations where username= '" + s + "' "; //
for prev values
                quantity = 0;
                id = 0;
                if (value == "Blankets") {
                    q = "UPDATE donations set Blankets= ? where username= '" + s +
"' ";
                    value2 = Integer.parseInt(textField.getText());
                    t_q = "UPDATE total set T_Blankets= ? where id=" + pt;
                } else if (value == "Meals") {
                    q = "UPDATE donations set Meals= ? where username= '" + s + "'
";
                    value2 = Integer.parseInt(textField.getText());
                    t_q = "UPDATE total set T_Meals= ? where id=" + pt;


                } else if (value == "Dry_Rations") {


                    q = "UPDATE donations set Dry_Rations= ? where username= '" +
s + "' ";
                    value2 = Integer.parseInt(textField.getText());
                    t_q = "UPDATE total set T_DryRations= ? where id=" + pt;
                } else if (value == "Shoes") {
                    q = "UPDATE donations set Shoes= ? where username= '" + s + "'
";
                    value2 = Integer.parseInt(textField.getText());
                    t_q = "UPDATE total set T_Shoes= ? where id=" + pt;
                } else if (value == "Clothes") {
                    q = "UPDATE donations set Clothes= ? where username= '" + s +
"' ";
                    value2 = Integer.parseInt(textField.getText());
                    t_q = "UPDATE total set T_Clothes= ? where id=" + pt;
                } else {
                    JOptionPane.showMessageDialog(null, "input column doesnt
exist");
                }
                try {
                    con = My_CNX.getConnection();
                    st = con.prepareStatement(q);
                    ps = con.prepareStatement(query);
                    st1 = con.prepareStatement(t_q);



                    r = ps.executeQuery();
                    while (r.next()) {
                        if (value == "Blankets") {
                            quantity = r.getInt("Blankets");
                            id = r.getInt("id");
                            st1.setInt(1, (value2 + tb));
                        } else if (value == "Meals") {
                            quantity = r.getInt("Meals");
                            id = r.getInt("id");
                            st1.setInt(1, (value2 + tm));
                        } else if (value == "Dry_Rations") {
                            quantity = r.getInt("Dry_Rations");
                            id = r.getInt("id");
                            st1.setInt(1, (value2 + td));
                        } else if (value == "Shoes") {
                            quantity = r.getInt("Shoes");
                            id = r.getInt("id");
                            st1.setInt(1, (value2 + ts));
                        } else if (value == "Clothes") {
                            quantity = r.getInt("Clothes");
                            id = r.getInt("id");
                            st1.setInt(1, (value2 + tc));
                        } else {
                            JOptionPane.showMessageDialog(null, "input column
doesnt exist");
                        }
                    }

                    System.out.println(quantity);
                    System.out.println(id);
                    st.setInt(1, (value2 + quantity));



                    rs = st.executeUpdate();
                    t_r = st1.executeUpdate();
                    if (st.executeUpdate() != 0) {
                        JOptionPane.showMessageDialog(null, "Your donation has 
been made");

                    } else {
                        JOptionPane.showMessageDialog(null, "Error; try again");
                    }

                } catch (Exception e) {
                    JOptionPane.showMessageDialog(null, e);
                }

            }

        });

        submitBtn.setFont(new Font("Rockwell", Font.PLAIN, 18));
        submitBtn.setBounds(351, 528, 127, 50);
        getContentPane().add(submitBtn);

        JLabel lblRate = new JLabel("Rate/unit ($)");
        lblRate.setFont(new Font("Rockwell", Font.PLAIN, 20));
        lblRate.setBounds(62, 212, 147, 64);
        getContentPane().add(lblRate);

        textField_Rate = new JTextField();
        textField_Rate.setBackground(new Color(255, 240, 245));
        textField_Rate.setEditable(false);
        textField_Rate.setFont(new Font("Rockwell", Font.PLAIN, 20));
        textField_Rate.setBounds(328, 224, 259, 47);

        JComboBox donationsType = new JComboBox();
        donationsType.setBackground(new Color(255, 228, 225));
        donationsType.setModel(
            new DefaultComboBoxModel(new String[] {
                "Blankets",
                "Meals",
                "Dry_Rations",
                "Shoes",
                "Clothes"
            }));
        donationsType.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                value = donationsType.getSelectedItem().toString();

                if (value == "Meals") {
                    String rate = "5";
                    textField_Rate.setText(rate);
                    n = Integer.parseInt(rate);

                } else if (value == "Blankets") {
                    String rate = "20";
                    textField_Rate.setText(rate);
                    n = Integer.parseInt(rate);

                } else if (value == "Shoes") {
                    String rate = "15";
                    textField_Rate.setText(rate);
                    n = Integer.parseInt(rate);

                } else if (value == "Dry_Rations") {
                    String rate = "10";
                    textField_Rate.setText(rate);
                    n = Integer.parseInt(rate);
                } else if (value == "Clothes") {
                    String rate = "20";
                    textField_Rate.setText(rate);
                    n = Integer.parseInt(rate);

                }
            }
        });

        donationsType.setFont(new Font("Rockwell", Font.PLAIN, 20));
        donationsType.setBounds(328, 112, 259, 64);
        getContentPane().add(donationsType);

        JLabel lblDonType = new JLabel("Type of Donation");
        lblDonType.setFont(new Font("Rockwell", Font.PLAIN, 20));
        lblDonType.setBounds(62, 112, 259, 64);
        getContentPane().add(lblDonType);

        JLabel lblQuant = new JLabel("Quantity");
        lblQuant.setFont(new Font("Rockwell", Font.PLAIN, 20));
        lblQuant.setBounds(62, 307, 259, 64);
        getContentPane().add(lblQuant);

        getContentPane().add(textField_Rate);
        textField_Rate.setColumns(10);
        textField_cost = new JTextField();
        textField_cost.setBackground(new Color(255, 240, 245));
        textField_cost.setEditable(false);
        textField_cost.setFont(new Font("Rockwell", Font.PLAIN, 20));
        textField_cost.setBounds(332, 423, 259, 47);
        getContentPane().add(textField_cost);
        textField_cost.setColumns(10);

        JLabel lblCost = new JLabel("Donation Cost");
```

```
lblCost.setBackground(new Color(255, 228, 225));
lblCost.setFont(new Font("Rockwell", Font.PLAIN, 20));
lblCost.setBounds(62, 411, 259, 64);
getContentPane().add(lblCost);

JLabel lblNewLabel_1 = new JLabel("New Donation");
lblNewLabel_1.setFont(new Font("Rockwell", Font.BOLD, 28));
lblNewLabel_1.setBounds(231, 24, 202, 57);
getContentPane().add(lblNewLabel_1);

textField = new JTextField();
textField.setBackground(new Color(255, 240, 245));
textField.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        x = Integer.parseInt(textField.getText());
        if (n == 20 && value == "Blankets") {
            cost = n * x;
            textField_cost.setText("" + cost);
        } else if (n == 5 && value == "Meals") {
            cost = n * x;
            textField_cost.setText("" + cost);
        } else if (n == 10 && value == "Dry_Rations") {
            cost = n * x;
            textField_cost.setText("" + cost);
        } else if (n == 15 && value == "Shoes") {
            cost = n * x;
            textField_cost.setText("" + cost);
        } else if (n == 20 && value == "Clothes") {
            cost = n * x;
            textField_cost.setText("" + cost);
        } else
            textField_cost.setText("Kindly Enter valid input");
            }
        });
        textField.setFont(new Font("Rockwell", Font.PLAIN, 20));
        textField.setEditable(true);
        textField.setColumns(10);
        textField.setBounds(328, 318, 259, 47);
        getContentPane().add(textField);

        JButton btnEnter = new JButton("New Donor");
        btnEnter.setBackground(new Color(255, 240, 245));
        btnEnter.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                Connection con = My_CNX.getConnection();
                ResultSet rs;

                try {
                    String f = "select * from users where username= '" + s + "' ";
                    PreparedStatement stmt = con.prepareStatement(f);
                    rs = stmt.executeQuery();
                    while (rs.next()) {
                        i_d = rs.getString("id");
                        System.out.println(i_d);
                    }

                } catch (SQLException ex) {
                    System.out.println(ex);
                }
                String SQL = "INSERT INTO donations
(id,username,Blankets,Meals,Dry_Rations,Shoes,Clothes) " +
                    "VALUES(?,?,?,?,?,?,?)";

                long id1 = 0;

                try (Connection conn = My_CNX.getConnection(); PreparedStatement
pstmt = conn.prepareStatement(SQL, Statement.RETURN_GENERATED_KEYS)) {
```

```
            pstmt.setString(1, i_d);
            pstmt.setString(2, s);
            pstmt.setInt(3, 0);
            pstmt.setInt(4, 0);
            pstmt.setInt(5, 0);
            pstmt.setInt(6, 0);
            pstmt.setInt(7, 0);

            int affectedRows = pstmt.executeUpdate();

            if (affectedRows > 0) {

                try (ResultSet rs1 = pstmt.getGeneratedKeys()) {
                    if (rs1.next()) {
                        id1 = rs1.getLong(1);
                    }
                } catch (SQLException ex) {
                    System.out.println(ex.getMessage());
                }
            }

        } catch (SQLException ex) {
            System.out.println(ex.getMessage());

        }

    }
});
btnEnter.setFont(new Font("Rockwell", Font.PLAIN, 18));
btnEnter.setBounds(183, 528, 127, 53);
getContentPane().add(btnEnter);

JButton btnNewButton = new JButton("< back");
```

```
        btnNewButton.setBackground(new Color(255, 228, 225));

        btnNewButton.addActionListener(new ActionListener() {

            public void actionPerformed(ActionEvent e) {

                DonorsPage dp = new DonorsPage();

                dp.setVisible(true);

                dispose();

            }

        });

        btnNewButton.setFont(new Font("Rockwell", Font.PLAIN, 12));

        btnNewButton.setBounds(34, 46, 89, 23);

        getContentPane().add(btnNewButton);


    }

}
```

My_CNX.java

```
import java.sql.Connection;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;

import javax.swing.JFrame;

import com.mysql.cj.jdbc.MysqlDataSource;

public class My_CNX extends JFrame {

       private static String servername = "localhost";
       private static String username = "root";
       private static String dbname = "users_db";
       private static Integer portnumber = 3306;
       private static String passsword = "";




             public static Connection getConnection() {
                    Connection  cnx = null;

                    MysqlDataSource datasource = new MysqlDataSource();

                    datasource.setServerName(servername);
                    datasource.setUser(username);
                    datasource.setPassword(passsword);
                    datasource.setDatabaseName(dbname);
                    datasource.setPortNumber(portnumber);
```

```
                     try {
                          cnx= datasource.getConnection();
                    } catch (SQLException ex) {

                          Logger.getLogger(" Get Connection -> " +
My_CNX.class.getName()).log(Level.SEVERE, null, ex);

                    }


                    return cnx;



             }
}
```

ReceiverHomepage.java

```
import java.sql.*
import java.util.*
import javax.swing.*
import java.awt.*




       public class recieverHomepage extends JFrame {

           private JPanel contentPane;
           private JTextField textField_username;
           private JTextField textField_mobNo;
           private JTextField textField_cnic;
           /**
            * Launch the application.
            */

           public boolean verifyFields() {

               String uname = textField_username.getText();
               String phone = textField_mobNo.getText();
               String cnic = textField_cnic.getText();
               //String pass2 = String.valueOf(passwordField_2.getPassword());


               if (uname.trim().equals("") || phone.trim().equals("") ||
phone.trim().equals("")) {
                   JOptionPane.showMessageDialog(null, "One or more fields are
empty", "Empty Fields", 2);
                   return false;
               }

                    else if(cnic.charAt(6)!='-' && cnic.charAt(13)!='-' &&
cnic.length()!=15)
               {
```

```
            JOptionPane.showMessageDialog(null, "CNIC mismatch", "invalid
format ", 2);
            return false;
            }



            //if everything is correct
            else {
                return true;
            }

        }

        public boolean checkUsername(String username) {

            PreparedStatement st;
            ResultSet rs;
            boolean username_exist = false;

            String query = "SELECT * FROM `recievers` WHERE `username` = ?";

            try {
                st = My_CNX.getConnection().prepareStatement(query);
                st.setString(1, username);
                rs = st.executeQuery();

                if (rs.next()) {
                    username_exist = true;
                    JOptionPane.showMessageDialog(null, "This username is
already taken, choose another Username", "Username Creation Failure ", 2);
                }


            } catch (SQLException e) {

Logger.getLogger(Registration.class.getName()).log(Level.SEVERE, null, e);
            }

            return username_exist;

        }



        //create a function to check

        //

        /**
         * Launch the application.
         */
        public static void main(String[] args) {
            EventQueue.invokeLater(new Runnable() {
                public void run() {
                    try {
                        recieverHomepage frame = new recieverHomepage();
                        frame.setVisible(true);
                    } catch (Exception e) {
```

```
                    e.printStackTrace();
                }
            }
        });
    }
    /**
     * Create the frame.
     */
    public recieverHomepage() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 676, 650);
        this.setPreferredSize(new Dimension(675, 650));
        this.pack();
        contentPane = new JPanel();
        contentPane.setBackground(new Color(135, 206, 235));
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel lblRegistration = new JLabel("Registration");
        lblRegistration.setFont(new Font("Rockwell", Font.BOLD, 32));
        lblRegistration.setBounds(221, 28, 250, 47);
        contentPane.add(lblRegistration);

        JLabel crUsernameLbl = new JLabel("Username");
        crUsernameLbl.setFont(new Font("Rockwell", Font.PLAIN, 20));
        crUsernameLbl.setBounds(49, 140, 114, 40);
        contentPane.add(crUsernameLbl);

        JLabel crPasswordLbl = new JLabel("CNIC Number");
        crPasswordLbl.setFont(new Font("Rockwell", Font.PLAIN, 20));
        crPasswordLbl.setBounds(49, 217, 138, 40);
        contentPane.add(crPasswordLbl);

        JLabel mobNumberLbl = new JLabel("Mobile Number");
        mobNumberLbl.setFont(new Font("Rockwell", Font.PLAIN, 20));
        mobNumberLbl.setBounds(50, 306, 161, 40);
        contentPane.add(mobNumberLbl);

        textField_username = new JTextField();
        textField_username.setBackground(new Color(255, 240, 245));
        textField_username.setFont(new Font("Rockwell", Font.PLAIN, 20));
        textField_username.setColumns(10);
        textField_username.setBounds(268, 143, 263, 35);
        contentPane.add(textField_username);

        textField_mobNo = new JTextField();
        textField_mobNo.setBackground(new Color(255, 240, 245));
        textField_mobNo.setFont(new Font("Rockwell", Font.PLAIN, 20));
        textField_mobNo.addKeyListener(new KeyAdapter() {
            @Override
            public void keyTyped(KeyEvent e) {

                if (!Character.isDigit(e.getKeyChar())) {
                    e.consume();

                }
            }
        });
```

```
               textField_mobNo.setColumns(10);
               textField_mobNo.setBounds(268, 306, 263, 35);
               contentPane.add(textField_mobNo);

               JButton btnNewButton = new JButton("Submit and Create account");
               btnNewButton.setBackground(new Color(255, 228, 225));
               btnNewButton.setFont(new Font("Rockwell", Font.PLAIN, 20));
               btnNewButton.addActionListener(new ActionListener() {
                   public void actionPerformed(ActionEvent ae) {



                       String username = textField_username.getText();
                       String cnic = textField_cnic.getText();
                       String phone = textField_mobNo.getText();


                       if (verifyFields()) {

                           if (!checkUsername(username)) {
                               PreparedStatement ps;
                               ResultSet rs;
                               String registerUserQuery = "INSERT INTO
`recievers`(`username`, `cnic`, `phone`) VALUES (?,?,?)";

                               try {

                                   ps =
My_CNX.getConnection().prepareStatement(registerUserQuery);
                                   ps.setString(1, username);
                                   ps.setString(2, cnic);
                                   ps.setString(3, phone);


                                   if (ps.executeUpdate() != 0) {
                                       JOptionPane.showMessageDialog(null, "Your
account has been created succesfully");
                                       recieverLogin rl = new recieverLogin();
                                       rl.setVisible(true);
                                       dispose();


                                   } else {
                                       JOptionPane.showMessageDialog(null, "Error;
Check your details");
                                   }

                               } catch (SQLException e) {


Logger.getLogger(Registration.class.getName()).log(Level.SEVERE, null, e);
                               }


                           }

                       }

                   }
               });



               btnNewButton.setBounds(189, 404, 300, 51);
               contentPane.add(btnNewButton);


               JButton btnNewButton_1 = new JButton("Click here to Login to an 
existing account");
               btnNewButton_1.setBackground(new Color(255, 228, 225));
               btnNewButton_1.addMouseListener(new MouseAdapter() {
                   @Override
                   public void mouseClicked(MouseEvent e) {
                       recieverLogin rl = new recieverLogin();
                       rl.setVisible(true);
                       dispose();
                   }

               });
               btnNewButton_1.addActionListener(new ActionListener() {
                   public void actionPerformed(ActionEvent e) {}
               });
               btnNewButton_1.setFont(new Font("Rockwell", Font.PLAIN, 20));
               btnNewButton_1.setBounds(71, 495, 517, 63);
               contentPane.add(btnNewButton_1);

               textField_cnic = new JTextField();
               textField_cnic.setBackground(new Color(255, 240, 245));
               textField_cnic.setFont(new Font("Rockwell", Font.PLAIN, 20));
               textField_cnic.setBounds(265, 219, 266, 36);
               contentPane.add(textField_cnic);
               textField_cnic.setColumns(10);

               JButton btnNewButton_2 = new JButton("< back");
               btnNewButton_2.setBackground(new Color(255, 228, 225));
               btnNewButton_2.setFont(new Font("Rockwell", Font.PLAIN, 11));
               btnNewButton_2.addActionListener(new ActionListener() {
                    public void actionPerformed(ActionEvent e) {

                          Homepage h = new Homepage();
                          h.setVisible(true);
                          dispose();

                    }
               });
               btnNewButton_2.setBounds(29, 48, 89, 23);
               contentPane.add(btnNewButton_2);


           }
       }
```

ReceiverLogin.java

```
import java.sql.*
import java.util.*
import javax.swing.*
import java.awt.*
```

```
public class recieverLogin extends My_CNX {

    public JFrame frame;
    public static String username;
    public JPanel contentPane;
    private JTextField usernameField;
    private JPasswordField cnicField;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {

        recieverLogin window = new recieverLogin();
        window.setVisible(true);

    }

    /**
     * Create the application.
     */
    public recieverLogin() {
        initialize();
    }

    /**
     * Initialize the contents of the frame.
     */
    private void initialize() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 700, 730);
        this.setPreferredSize(new Dimension(700, 730));
        this.pack();
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel usernameLbl = new JLabel("Username");
        usernameLbl.setFont(new Font("Tahoma", Font.PLAIN, 22));
        usernameLbl.setBounds(56, 123, 136, 53);
        contentPane.add(usernameLbl);

        usernameField = new JTextField();
        usernameField.addFocusListener(new FocusAdapter() {

        });
        usernameField.setBounds(223, 132, 300, 38);
        usernameField.setColumns(10);
        contentPane.add(usernameField);

        JLabel welcomeLbl = new JLabel("Welcome, Please Login In");
        welcomeLbl.setFont(new Font("Tahoma", Font.PLAIN, 26));
        welcomeLbl.setHorizontalAlignment(SwingConstants.CENTER);
        welcomeLbl.setBounds(10, 28, 684, 67);
        contentPane.add(welcomeLbl);

        JButton loginBtn = new JButton("Login");
        loginBtn.addActionListener(new ActionListener() {
```

```
      public void actionPerformed(ActionEvent e) {

        PreparedStatement st;
        ResultSet rs;

        //getting username and cnic

        username = usernameField.getText();
        String cnic = String.valueOf(cnicField.~~getText()~~);

        //string query to check whether username and cnic exist in
database
        String query = "SELECT * FROM `recievers` WHERE `username` = ? AND
`cnic`= ?";

        if (username.trim().equals("username")) {
          JOptionPane.showMessageDialog(null, "Enter Your Username",
"Empty Username", 2);
        } else if (cnic.trim().equals("cnic")) {
          JOptionPane.showMessageDialog(null, "Enter Your CNIC", "Empty
CNIC", 2);
        } else {



          try {
            st = My_CNX.getConnection().prepareStatement(query);

            st.setString(1, username);
            st.setString(2, cnic);

            rs = st.executeQuery();

            if (rs.next()) {

              RecieverPage page = new RecieverPage();
              page.setVisible(true);
              dispose();



            } else {
              JOptionPane.showMessageDialog(null, "Invalid
username/cnic, try again", "login error", 2);
            }


          } catch (SQLException ex) {
            Logger.getLogger(" Get Connection -> " +
My_CNX.class.getName()).log(Level.SEVERE, null, ex);
          }
        }
      }
    });
    loginBtn.setFont(new Font("Tahoma", Font.PLAIN, 18));
    loginBtn.setBounds(289, 329, 136, 46);
    contentPane.add(loginBtn);

    cnicField = new JPasswordField();
```

```
        cnicField.setFont(new Font("Tahoma", Font.PLAIN, 20));
        cnicField.setBounds(223, 227, 300, 46);
        contentPane.add(cnicField);

        JButton navRegBtn = new JButton("Don't have an account?, Click here to 
register");
        navRegBtn.setFont(new Font("Tahoma", Font.PLAIN, 18));
        navRegBtn.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent ae) {

             recieverHomepage rh = new recieverHomepage();
                rh.setVisible(true);
                dispose();
            }
        });

        navRegBtn.setFont(new Font("Tahoma", Font.PLAIN, 18));
        navRegBtn.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {}
        });
        navRegBtn.setBounds(43, 428, 606, 87);
        contentPane.add(navRegBtn);
        JLabel lblCnic = new JLabel("CNIC");
        lblCnic.setFont(new Font("Tahoma", Font.PLAIN, 22));
        lblCnic.setBounds(56, 227, 136, 53);
        contentPane.add(lblCnic);
    }
}
```

ReceiverPage.java

```
import java.sql.*
import java.util.*
import javax.swing.*
import java.awt.*

public class RecieverPage extends My_CNX {

    private JPanel contentPane;
    String value,q,query,value2,t_q,query1;
    int quantity;
    int V;
    String id="1";
    int b,b1,c,c1,m,m1,p,p1,s,s1;
    private JTable table;
    private JTextField itemQuantity;
    int item=0;
    int count=3;
    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    RecieverPage frame = new RecieverPage();
                    frame.setVisible(true);

                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the frame.
     */
    public RecieverPage() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 887, 612);
        contentPane = new JPanel();
        contentPane.setBackground(new Color(135, 206, 235));
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);
        table = new JTable();
        table.setFont(new Font("Rockwell", Font.PLAIN, 11));
        table.setBorder(new LineBorder(new Color(0, 0, 0)));
        table.setBounds(458, 145, 297, 96);
        contentPane.add(table);

        int d=0, e=0, f=0,g=0,h=0;
        String dd,ee,ff,gg,hh;

        try {

            PreparedStatement st;
            Connection con=My_CNX.getConnection();
             ResultSet rs;

        String sql="SELECT * FROM total";
        st=con.prepareStatement(sql);
         rs = st.executeQuery();
         while(rs.next()) {
              d = rs.getInt("T_Blankets");
                 e = rs.getInt("T_Meals");
                 f = rs.getInt("T_DryRations");
                 g = rs.getInt("T_Shoes");
                 h = rs.getInt("T_Clothes");

         }
         if(d>0) {
             dd="Available";
         }
         else
             dd="Not Available";
         if(e>0) {
             ee="Available";
         }
         else
             ee="Not Available";
         if(f>0) {
             ff="Available";
         }
         else
             ff="Not Available";
         if(g>0) {
             gg="Available";
```

```
              }
              else
                      gg="Not Available";
              if(h>0) {
                      hh="Available";
              }
              else
                      hh="Not Available";

              table.setModel(new DefaultTableModel(
                    new Object[][] {
                          {"Blankets",dd},
                          {"Meals", ee},
                          {"DryRations", ff},
                          {"Shoes", gg},
                          {"Clothes", hh},
                          //{null, null},
                    },
                    new String[] {
                          "Item Name", "Quantity"
                    }
              ) {
                    Class[] columnTypes = new Class[] {
                          String.class, String.class
                    };
                    public Class getColumnClass(int columnIndex) {
                          return columnTypes[columnIndex];
                    }
              });
             }
             catch (Exception ex) {
                    JOptionPane.showMessageDialog(null,ex);
             }
            Font headerFont = new Font("Tahoma", Font.PLAIN, 18);

            JComboBox itemBox = new JComboBox();
            itemBox.setBackground(new Color(255, 240, 245));

            itemBox.addActionListener(new ActionListener() {
             public void actionPerformed(ActionEvent e) {
                    value=itemBox.getSelectedItem().toString();
                    item++;
                    if(item>=3) {
                          itemBox.~~disable~~();

                    }
             }
            });
                    itemBox.setFont(new Font("Rockwell", Font.PLAIN, 20));
                    itemBox.setModel(new DefaultComboBoxModel(new String[]
{"Blankets", "Meals", "DryRations", "Shoes", "Clothes"}));
                    itemBox.setBounds(191, 145, 182, 41);
                    contentPane.add(itemBox);

                    itemQuantity = new JTextField();
                    itemQuantity.setFont(new Font("Rockwell", Font.PLAIN, 20));
                    itemQuantity.setBackground(new Color(255, 240, 245));
                    itemQuantity.addActionListener(new ActionListener() {
                          public void actionPerformed(ActionEvent e) {
                                 value2=itemQuantity.getText();
                                       V=Integer.parseInt(value2);
                                       count=count-V;
                                       System.out.println(count);

                          }
                    });
                    itemQuantity.setBounds(192, 260, 181, 41);
                    contentPane.add(itemQuantity);
                    itemQuantity.setColumns(1);



             JButton printBtn = new JButton("Confirm");
             printBtn.setBackground(new Color(255, 228, 225));
             printBtn.addActionListener(new ActionListener() {
                    public void actionPerformed(ActionEvent e) {
                          PreparedStatement st,ps,ps1,q1s,qds,st1;

                           Connection con=My_CNX.getConnection();
                           ResultSet r1,rq,rt;
                           String que;
                           int pt=17;
                           int r;
                           int d1=0,d2=0,d3=0,d4=0,d5=0;
                           if(V<=3 && count>=0){
                          try {
                                  query = "select 
T_Blankets,T_Meals,T_DryRations,T_Shoes,T_Clothes from total "; //for prev values
                               ps=con.prepareStatement(query);
                               rt=ps.executeQuery();
                               while(rt.next()){
                                   b=rt.getInt("T_Blankets");
                                   m=rt.getInt("T_Meals");
                                   p=rt.getInt("T_DryRations");
                                   s=rt.getInt("T_Shoes");
                                  c=rt.getInt("T_Clothes");

                               }


                                  query1 = "select * from donated_items where id = 
'" +id+ "' "; //for prev values
                               ps=con.prepareStatement(query1);
                               rq=ps.executeQuery();
                               while(rq.next()) {
                                  d1=rq.getInt("Blankets");
                                  d2=rq.getInt("Meals");
                                  d3=rq.getInt("DryRations");
                                  d4=rq.getInt("Shoes");
                                  d5=rq.getInt("Clothes");

                               }

                                  if(value=="Blankets") {
                                  q="UPDATE donated_items set Blankets= "+(V+d1);
                                  t_q="UPDATE total set T_Blankets= "+(b-V) ;

                              }
                              else if(value=="Meals") {
                                 q="UPDATE donated_items set Meals= "+(V+d2);
                                  t_q="UPDATE total set T_Meals= " +(m-V) ;
                              }
                              else if(value=="DryRations") {


                                 q="UPDATE donated_items set DryRations= "+(V+d3);
                                  t_q="UPDATE total set T_DryRations= " +(p-V) ;

                              }
                              else if(value=="Shoes") {
                                 q="UPDATE donated_items set Shoes= "+(V+d4);
                                  t_q="UPDATE total set T_Shoes= " +(s-V) ;

                              }
                              else if (value=="Clothes") {
                                 q="UPDATE donated_items set Clothes="+(V+d5);
                                  t_q="UPDATE total set T_Clothes= " +(c-V);
                              }
                              else {
                                 JOptionPane.showMessageDialog(null, "input column
doesnt exist");
                              }

                                        st=con.prepareStatement(q);
                                        st1=con.prepareStatement(t_q);
                                        r=st.executeUpdate();
                                        int r2=st1.executeUpdate();


                                         if(st.executeUpdate()!=0){

       JOptionPane.showMessageDialog(null, "You have received the donations");



                                              }else {

       JOptionPane.showMessageDialog(null, "Error; try again");
                                              }


                          }

                            catch (Exception ex) {

       JOptionPane.showMessageDialog(null,ex);
                                              }
                    }
                     else
                  {
                           JOptionPane.showMessageDialog(null, "You have reached
your limit for donations for today");
                  }
                    }


             });
             printBtn.setFont(new Font("Rockwell", Font.PLAIN, 20));
             printBtn.setBounds(191, 436, 177, 41);
             contentPane.add(printBtn);



             JTextArea recieptText = new JTextArea();
             recieptText.setFont(new Font("Rockwell", Font.PLAIN, 13));
             recieptText.setBounds(418, 306, 418, 177);
             contentPane.add(recieptText);

             JLabel itemLbl = new JLabel("Item\r\n");
             itemLbl.setFont(new Font("Tahoma", Font.PLAIN, 20));
             itemLbl.setBounds(53, 143, 128, 41);
             contentPane.add(itemLbl);

             JLabel quantityLbl = new JLabel("Quantity\r\n");
             quantityLbl.setFont(new Font("Tahoma", Font.PLAIN, 20));
             quantityLbl.setBounds(53, 256, 128, 41);
             contentPane.add(quantityLbl);



             JLabel lblNewLabel_1 = new JLabel("Only 3 items can be claimed per
day");
             lblNewLabel_1.setFont(new Font("Tahoma", Font.PLAIN, 22));
             lblNewLabel_1.setBounds(53, 522, 371, 27);
             contentPane.add(lblNewLabel_1);

             JButton addBtn = new JButton("Add to Cart");
             addBtn.setBackground(new Color(255, 228, 225));
             addBtn.addActionListener(new ActionListener() {
                    int i=0;
                    public void actionPerformed(ActionEvent e) {


                             if(itemQuantity.getText().isEmpty())
                             {
                                    JOptionPane.showMessageDialog(null, "empty
quantity");
                             }
                             else {
                                    i++;
                              if(i == 1)
                              {
                                 recieptText.setText(recieptText.getText()+"\t
CARE n SHARE \t\n"+"\t NUM\t PRODUCT\t QUANTITY
\n\t"+i+"\t"+itemBox.getSelectedItem().toString()+"\t"+V+"\n\t");
                              }
                              else{
                                   recieptText.setText("\t\n\t"+
recieptText.getText()+i+"\t"+itemBox.getSelectedItem().toString()+"\t"+V+"\n\t");
                              }
                              JOptionPane.showMessageDialog(null, "Kindly Press
OK to get Donation");
                          }
                    }
```

```
        });
        addBtn.setFont(new Font("Rockwell", Font.PLAIN, 20));
        addBtn.setBounds(186, 368, 187, 47);
        contentPane.add(addBtn);

        JLabel lblNewLabel_2 = new JLabel("Get Donations\r\n");
        lblNewLabel_2.setFont(new Font("Rockwell", Font.BOLD, 32));
        lblNewLabel_2.setBounds(328, 21, 238, 41);
        contentPane.add(lblNewLabel_2);

        JButton btnNewButton = new JButton("exit");
        btnNewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                Homepage hp = new Homepage();
                hp.setVisible(true);
                dispose();
            }
        });
        btnNewButton.addMouseMotionListener(new MouseMotionAdapter() {
            @Override
            public void mouseMoved(MouseEvent e) {

                btnNewButton.setToolTipText("sample text
xxxxxxxxxxxxxxxxxxxxxxxxxxxx");

            }
        });
        btnNewButton.setBounds(709, 38, 89, 23);
        contentPane.add(btnNewButton);

        JButton btnNewButton_1 = new JButton("Print Receipt");
        btnNewButton_1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

            try {

                boolean complete = recieptText.print();
                if(complete) {
                    JOptionPane.showMessageDialog(null, "done
printing", "information",
                    JOptionPane.INFORMATION_MESSAGE);

                }else {
                    JOptionPane.showMessageDialog(null,"Printing!",
"Printer", JOptionPane.ERROR_MESSAGE);
                }
            }catch(PrinterException ex){
                JOptionPane.showMessageDialog(null, ex);

            }

            }
        });
        btnNewButton_1.setFont(new Font("Rockwell", Font.PLAIN, 20));
        btnNewButton_1.setBounds(540, 494, 163, 56);
        contentPane.add(btnNewButton_1);
```

```
          }
}
```