**JAVA SWING BASED**

**SPAM DETECTION SYSTEM-SQL CONNECTIVITY USING JDBC**

*A*

*Report*

*Submitted in partial fulfillment of the*

Requirements for the award of the Degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

By

**M. Nikhil Narasimha <1602-20-737-025>**

**Under the Guidance of**

**B. Leelavathy**

**Department of Information Technology**

**Vasavi College of Engineering (Autonomous)**

**(Affiliated to Osmania University)**

**Ibrahimbagh, Hyderabad-31**

****

**2021-2022**

**BONAFIDE CERTIFICATE**

This to Certify that the project report titled **“SPAM DETECTION SYSTEM”** project work of Mr. M. Nikhil Narasimha bearing Roll.no:1602-20-737-025 who carried out this project under my supervision in the IV semester for the academic year 2021-2021.

*Signature Signature*

*external examiner internal examiner*

# 

# Spam Detection

**By M Nikhil Narasimha**

**1602-20-737-025**

**ABSTRACT**

Spam is any kind of unwanted, unsolicited digital communication that gets sent out in bulk. Often spam is sent via email, but it can also be distributed via text messages, phone calls, or social media Spammers use many forms of communication to bulk-send their unwanted messages. Some of these are marketing messages peddling unsolicited goods. Other types of spam messages can spread malware, trick you into divulging personal information, or scare you into thinking you need to pay to get out of trouble.

Email spam filters catch many of these types of messages, and phone carriers often warn you of a “spam risk” from unknown callers. Whether via email, text, phone, or social media, some spam messages do get through, and you want to be able to recognize them and avoid these threats.

The Project Spam Detection is an application developed using Java and SQL which will help detect spam easily or efficiently this will help users filter out spam emails.

**REQUIREMENT ANALYSIS:**

**LIST OF TABLES:**

* User
* Login
* Inbox
* Message
* Spam
* Blocked Contacts

**LIST OF ATTRIBUTES WITH DOMAIN TYPES:**

**User\_Login:**

Email varchar2(25)

Username varchar2(25)

Date date

Phone Number number(10)

**Login:**

Email VARCHAR2(25),Password VARCHAR2(25)

**Inbox:**

I\_ID Number(10),

From varchar2(25),

To varchar2(25)

**Message:**

ID Number(10);

Content varchar2(60)

Subject varchar2(20)

**Spam:**

ID number(10)

Is\_Spam Number(3)

Type Varchar2(10)

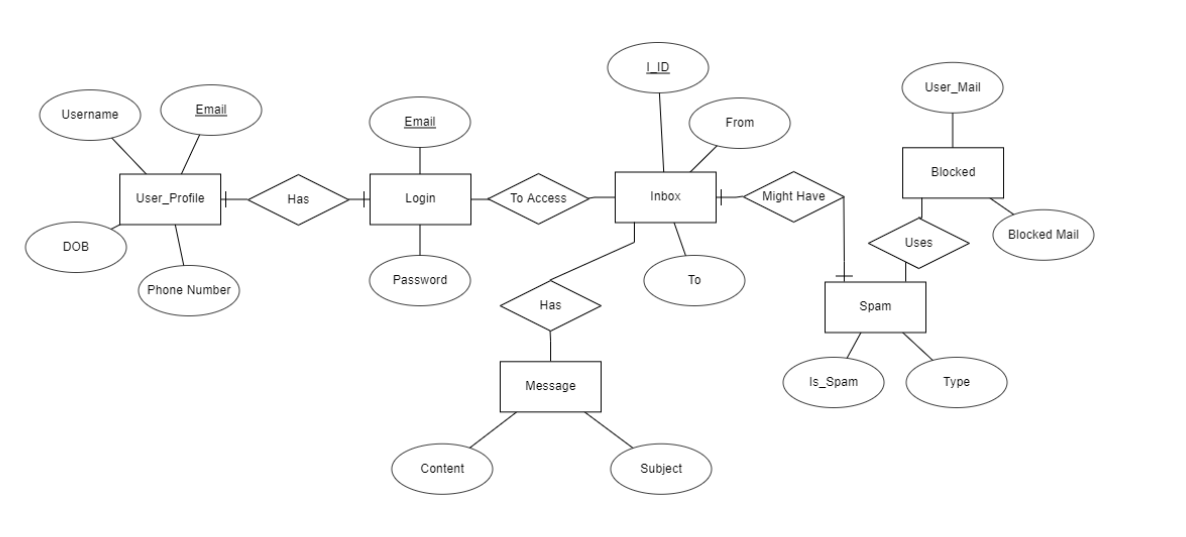
**Blocked Contacts:**

User Mail Varchar2(25)

Blocked Mail VarChar2(25)

**DESIGN ANALYSIS:**

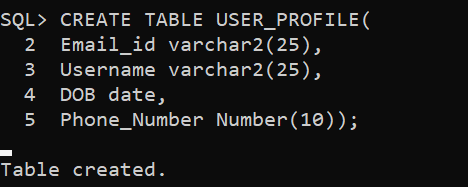
**ER DIAGRAM:**

****

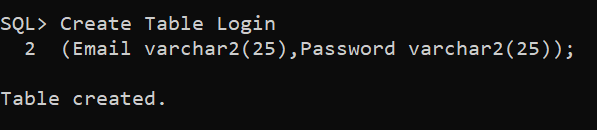
**RELATIONAL MODEL:**

**DDL OPERATIONS:**

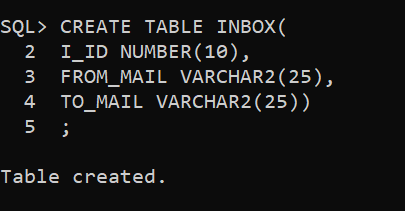
**User Profile**

****

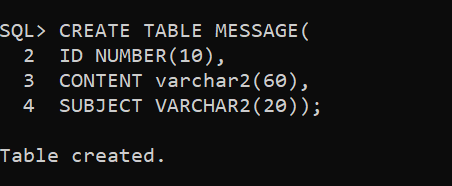
**Login**

****

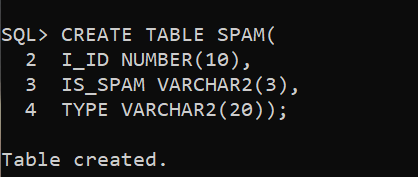
**Inbox**



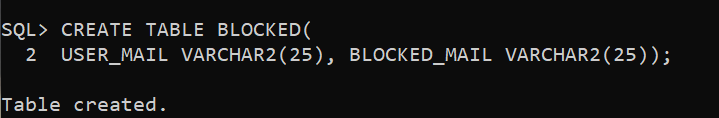
**MESSAGE**

****

**SPAM**

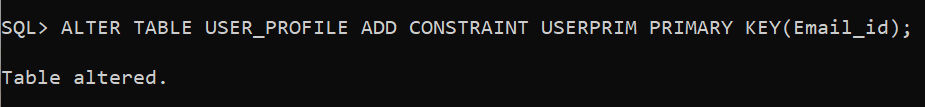
****

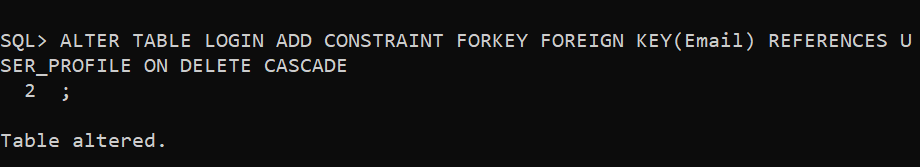
**Blocked**

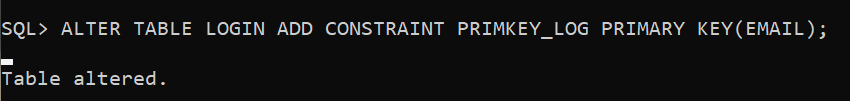
****

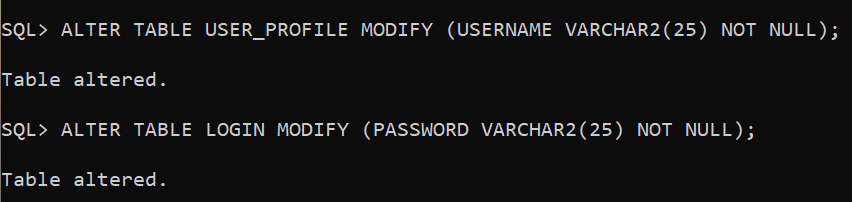
**KEY CONSTRAINTS:**

**User Profile and Login**

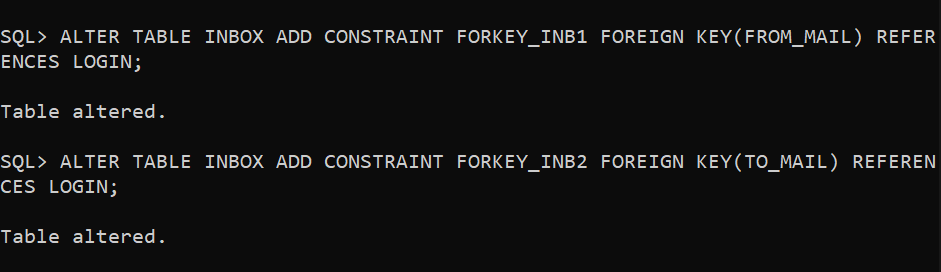
****

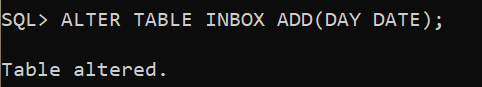
****

****

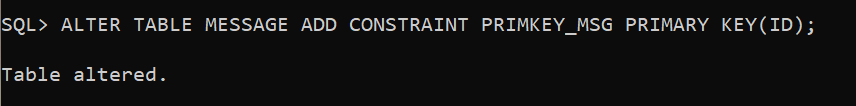
****

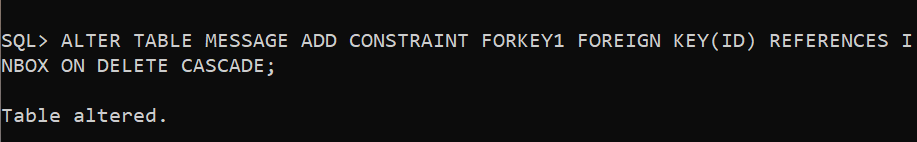
**Inbox**

****

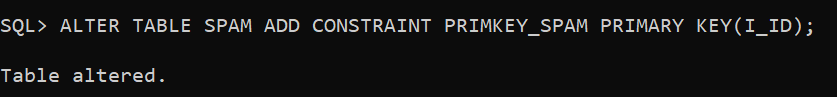
****

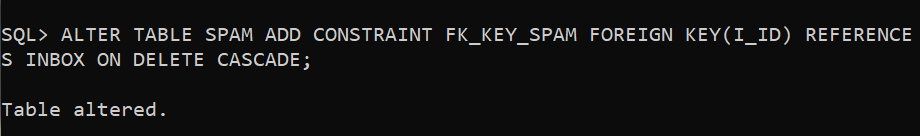
**Message**

****

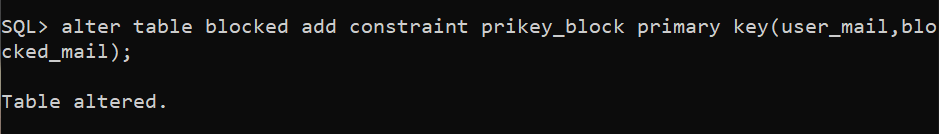
****

**Spam**





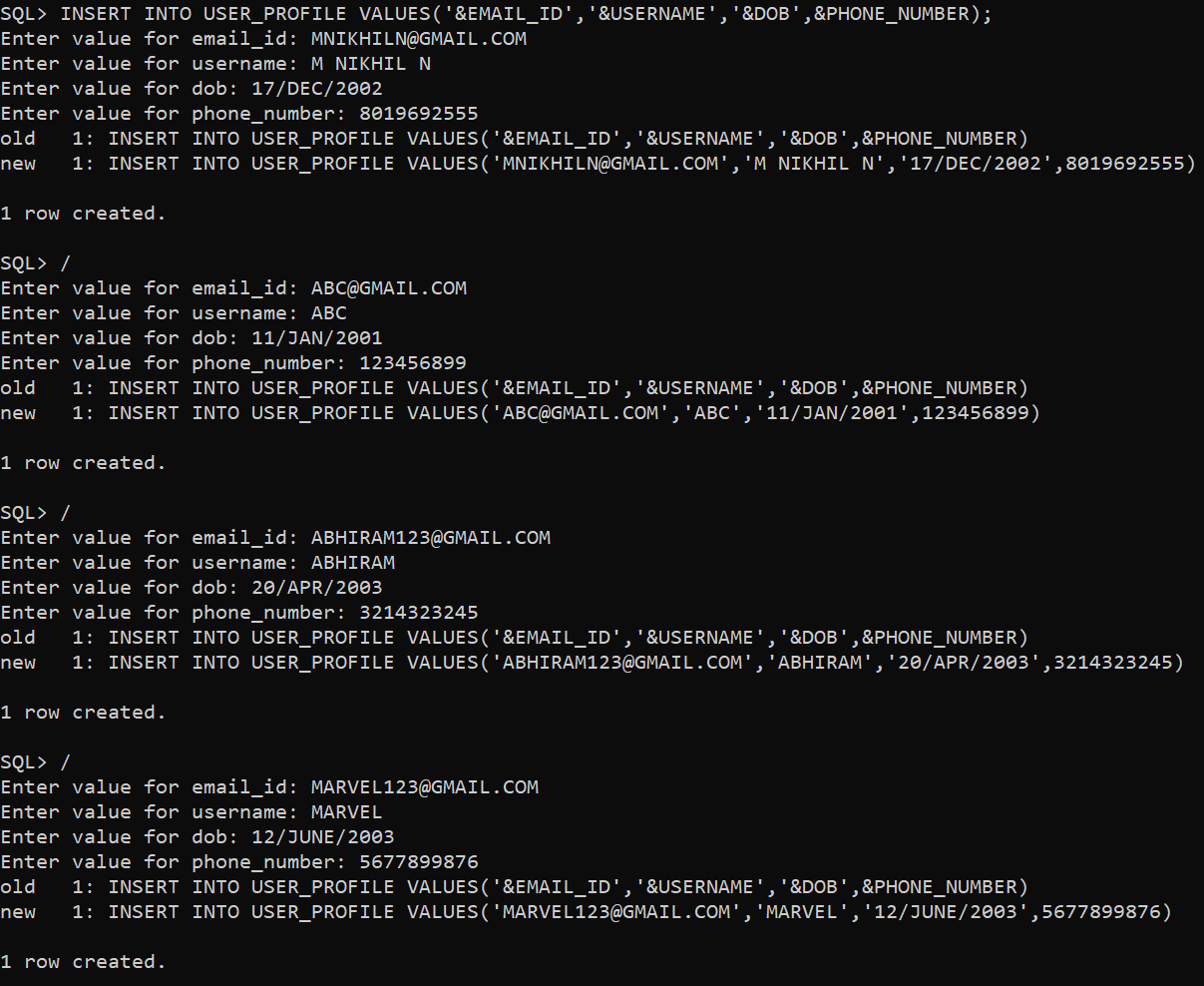
**Blocked**

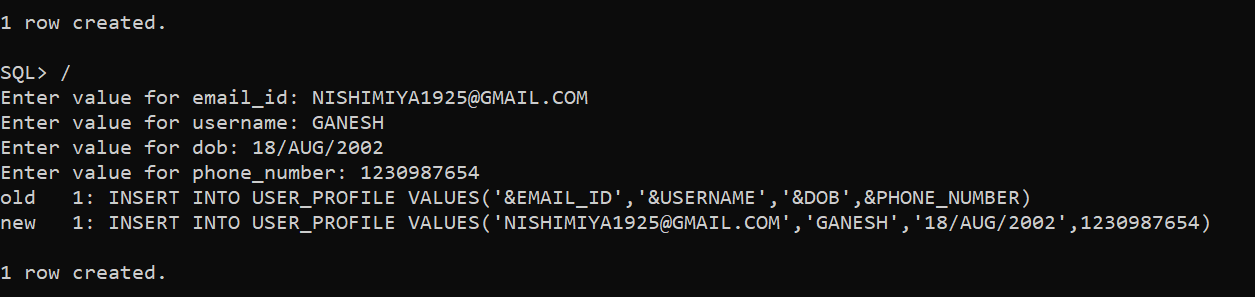


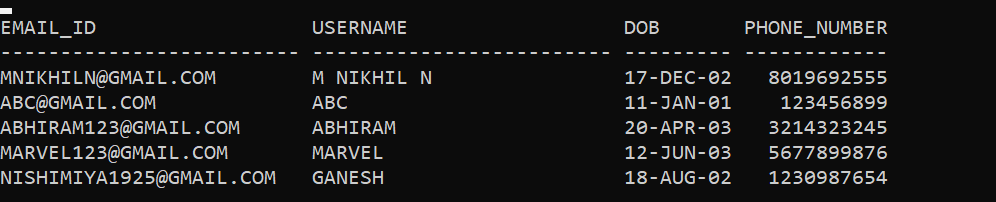
**MODIFICATION OF DB:**

**DML OPERATIONS:**

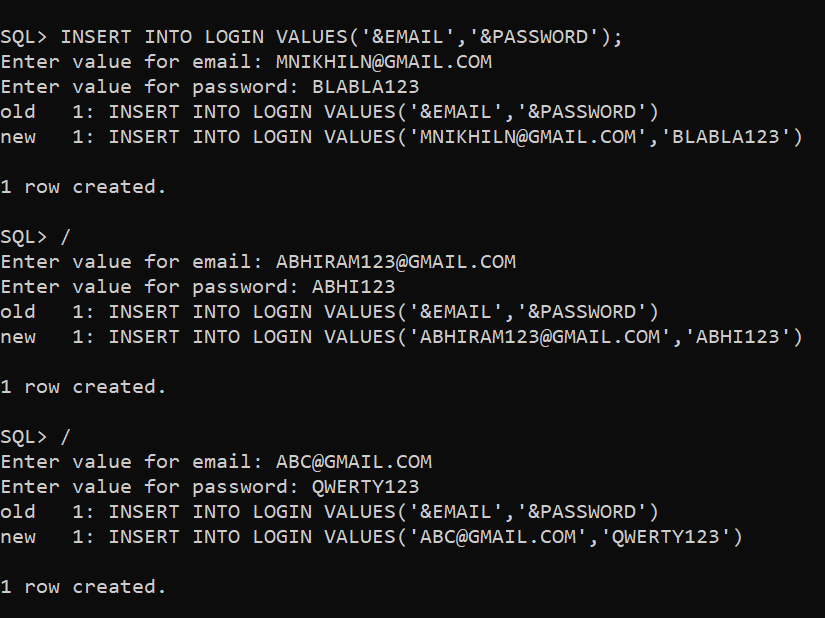
**User\_Profile**

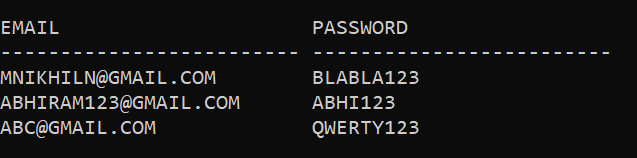
****

****

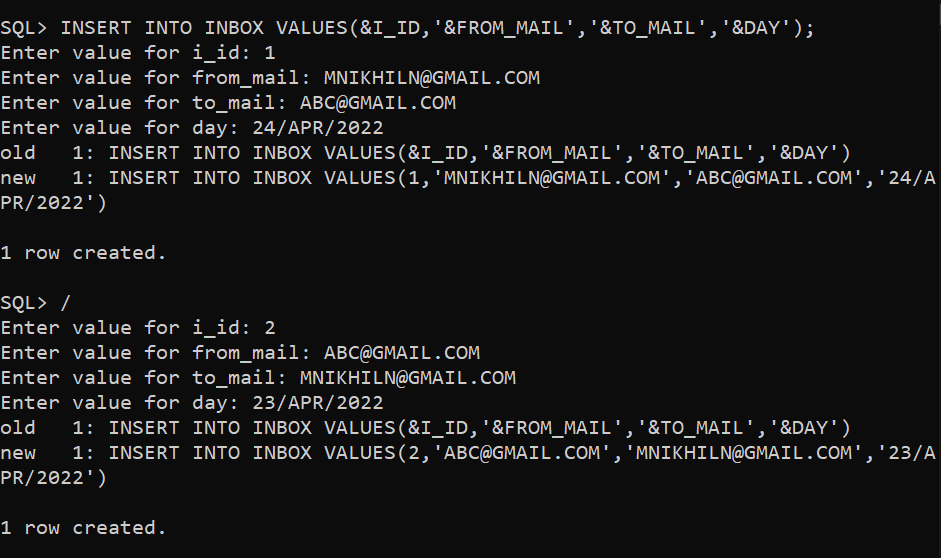
****

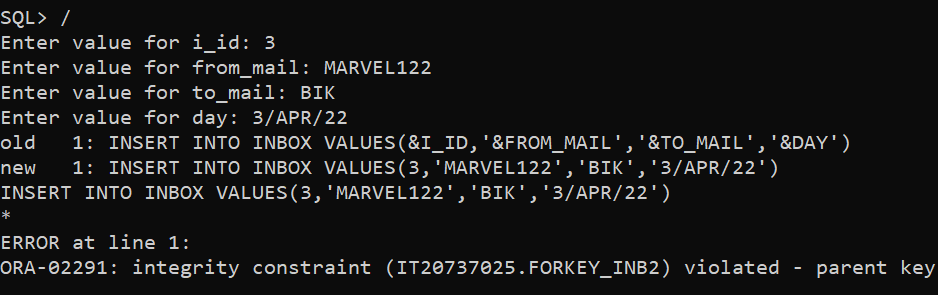
**Login**

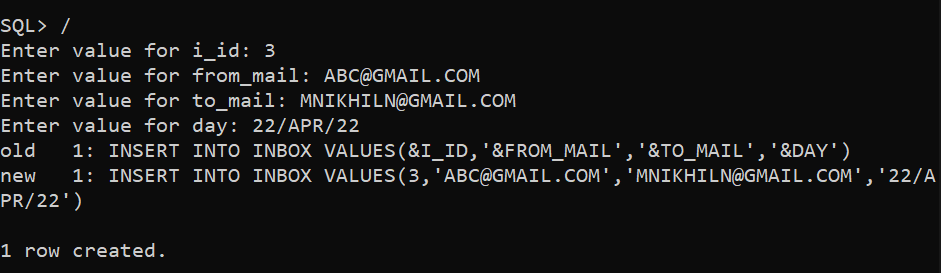
****

****

**Inbox**

****

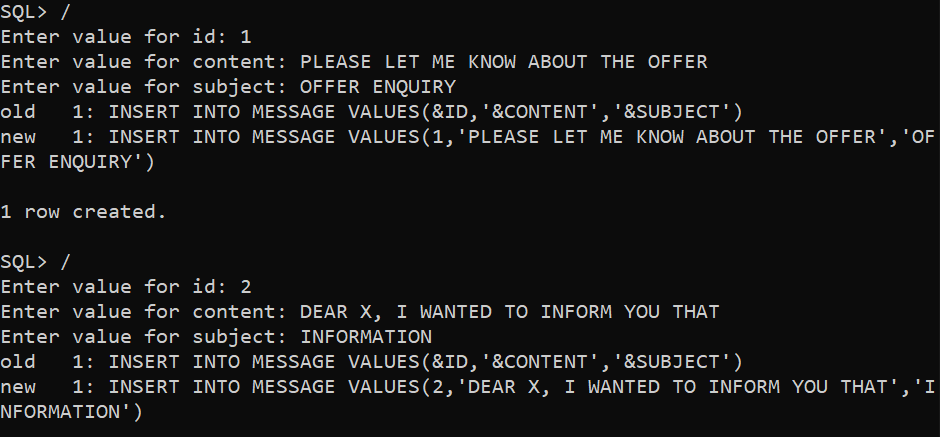
****

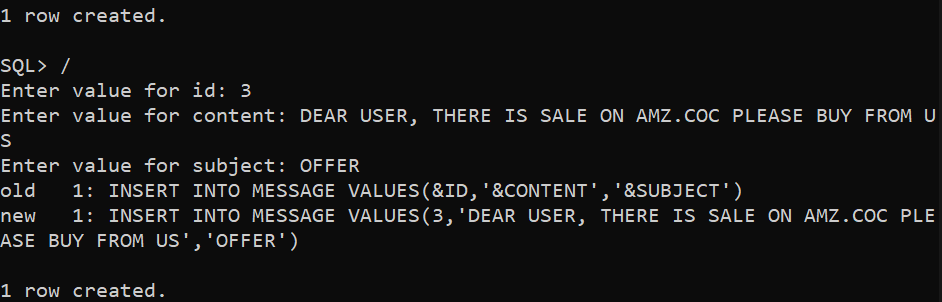
****



**Message**

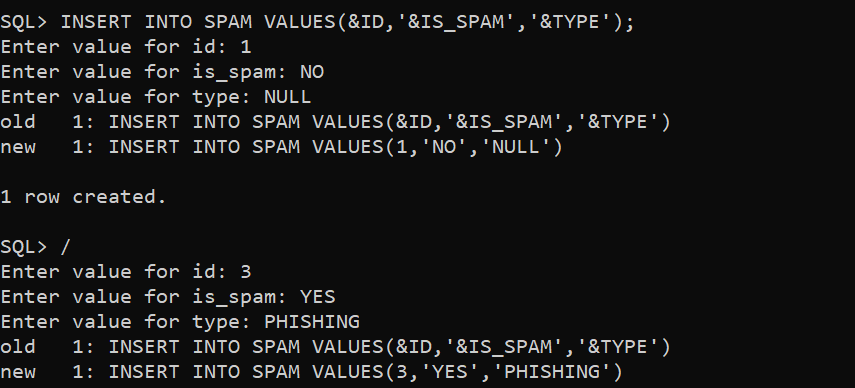
****

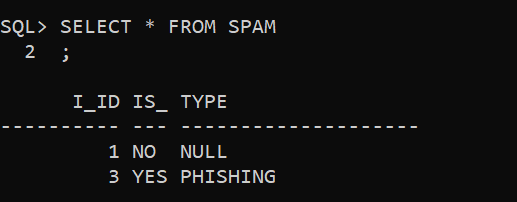
****

****

****

**SPAM**

****

****

**BLOCKED**

Insert Into Blocked Values (‘MNIKHILN@GMAIL.COM’,’ABC123@GMAIL.COM’)

# Implementation

MyClassSwing.java

import java.awt.\*;

import java.time.LocalDateTime; // Import the LocalDateTime class

import java.time.format.DateTimeFormatter; // Import the DateTimeFormatter class

import javax.imageio.ImageIO;

import javax.swing.\*;

import javax.swing.event.ListSelectionEvent;

import javax.swing.event.ListSelectionListener;

import javax.swing.table.TableColumnModel;

import java.awt.event.\*;

import java.sql.\*;

class FirstPg extends JFrame implements ActionListener, ItemListener {

    JLabel spammax;

    JLabel lblTitle, lbllogin, lblpass, lblfrom, lblto, lblmsg, lblsub, lblis\_spam, lbltype;

    JLabel lblid, lblfrom2, lblto2, lblmsg2, lblsub2, lblis\_spam2, lbltype2, lbltable;

    JLabel lblid3;

    JTextField tfid, tffrom2, tfto2, tfmsg2, tfsub2, tftype2;

    JScrollPane sp\_all, sp\_block, sp\_delete;

    JTextField tflogin, tffrom, tfto, tfmsg, tfsub, tftype;

    JTextArea tfdisplay;

    JComboBox l\_is, cb\_display;

    JPasswordField tfpass;

    JComboBox l\_is2;

    JPanel pnl1, pnl\_new, pnl\_update, pnl\_display;

    panelDelete pnl\_delete;

    paneluser pnl\_user;

    JButton btupdate, btlogin, btSubmit, btdelete, btview, bt;

    JButton button;

    JButton btback;

    JButton btnew,btcreate;

    Font f1, f2, f3, f4, f5;

    MenuBar mbar;

    JTable tbdisplay, tbdisplay\_block, tbdelete;

    Color r1;

    MenuItem item1, item2, item3, item4, item5;

    // JDBC

    Connection con;

    public FirstPg() {

        jdbcInitial();

        initializeComponents();

        registerListeners();

        addComponenetsToFrame();

        setVisible(true);

        setSize(335, 335);

        setTitle("Spam Detection");

        setLayout(null);

        pnl1.setBounds(82, 90, 150, 70);

        lblTitle.setBounds(84, 20, 165, 70);

        lblTitle.setFont(f1);

        btlogin.setBounds(125, 175, 75, 25);

        btnew.setBounds(120, 210, 85, 25);

        btlogin.setFont(f2);

        btnew.setFont(f2);

        // panel1();

        setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        r1 = new Color(142, 183, 230);

        ImageIcon imgf;

        pnl\_user = new paneluser();

        add(pnl\_user);

        pnl\_user.setVisible(false);

        pnl\_user.setBounds(82, 82, 150, 118);

        pnl\_user.setBackground(r1);

        btcreate = new JButton("Create");

        btcreate.setBounds(120, 210, 85, 25);

        add(btcreate);

        btcreate.setVisible(false);

        btcreate.addActionListener(this);

        btback = new JButton("<-");

        add(btback);

        btback.setVisible(false);

        getContentPane().setBackground(r1);

        try

        {

        Image imgfav = ImageIO.read(getClass().getResource("icon2.png"));

        imgf = new ImageIcon(imgfav);

        setIconImage(imgf.getImage());

    }

        catch(Exception e)

        {

            JOptionPane.showMessageDialog(this, e.getMessage(), "ERROR", JOptionPane.ERROR\_MESSAGE);

        }

        pnl1.setBackground(r1);

    }

    public void jdbcInitial() {

        try {

            Class.forName("oracle.jdbc.driver.OracleDriver");

        } catch (Exception e) {

            JOptionPane.showMessageDialog(this, "Network Error/ Driver error", "ERROR", JOptionPane.ERROR\_MESSAGE);

        }

    }

    public void display\_blocked() {

        try {

            int c = 0;

            con = DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms", "it20737025", "vasavi");

            Statement stmt = con.createStatement();

            ResultSet rs = stmt.executeQuery("select \* from BLOCKED");

            while (rs.next())

                c++;

            String[][] data = new String[c][2];

            int i = 0;

            ResultSet rs1 = stmt.executeQuery("select \* from BLOCKED");

            while (rs1.next()) {

                data[i][0] = rs1.getString(1);

                data[i][1] = rs1.getString(2);

                i++;

            }

            String[] colname = { "EMAIL", "NAME" };

            tbdisplay\_block = new JTable(data, colname);

            tbdisplay\_block.setBounds(10, 30, 275, 175);

            sp\_block = new JScrollPane();

            sp\_block.setViewportView(tbdisplay\_block);

            pnl\_display.add(sp\_block);

            sp\_block.setBounds(10, 30, 275, 175);

            con.close();

        } catch (Exception e) {

            JOptionPane.showMessageDialog(this, e.getMessage(), "ERROR", JOptionPane.ERROR\_MESSAGE);

        }

    }

    private void registerListeners() {

        btlogin.addActionListener(this);

        btnew.addActionListener(this);

    }

    public void showtable(int opt) {

        try {

            int c = 0;

            con = DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms", "it20737025", "vasavi");

            Statement stmt = con.createStatement();

            Statement stmt2 = con.createStatement();

            Statement stmt3 = con.createStatement();

            ResultSet rs1 = stmt.executeQuery(

                    "SELECT Inbox.I\_ID, Inbox.From\_Mail,Inbox.To\_Mail,Message.Content,Message.Subject FROM Inbox INNER JOIN Message ON Inbox.I\_ID=Message.ID");

                ResultSet rs2 = stmt2.executeQuery(

                        "SELECT  SPAM.IS\_SPAM,SPAM.TYPE FROM Inbox INNER JOIN SPAM ON Inbox.I\_ID=SPAM.I\_ID");

                        ResultSet rs3 = stmt3.executeQuery(

                            "SELECT S.TYPE,COUNT(S.TYPE) FROM SPAM S GROUP BY TYPE ORDER BY COUNT(S.TYPE) DESC");

            if (opt == 2)

            {

                this.remove(spammax);

            }

            if(rs3.next())

            {

                spammax = new JLabel("");

                spammax.setText("The frequent spam type: "+rs3.getString(1)+"\n (Count:"+rs3.getString(2)+")");

                add(spammax);

                spammax.setBounds(500, 275, 300, 20);

                String s1 = "Highly reported : "+rs3.getString(1);

                spammax.setToolTipText(s1);

                spammax.setForeground(Color.RED);

            }

            String colnames[] = { "ID", "From", "To", "Date", "Content", "Subject","IS\_SPAM","TYPE" };

            while (rs1.next())

                c++;

            String rows[][] = new String[c][9];

            ResultSet rs = stmt.executeQuery(

                    "SELECT Inbox.I\_ID, Inbox.From\_Mail,Inbox.To\_Mail,Inbox.Day,Message.Content,Message.Subject FROM Inbox INNER JOIN Message ON Inbox.I\_ID=Message.ID");

            int i = 0;

            while (rs.next()) {

                rows[i][0] = rs.getString(1);

                rows[i][1] = rs.getString(2);

                rows[i][2] = rs.getString(3);

                rows[i][3] = rs.getString(4);

                rows[i][4] = rs.getString(5);

                rows[i][5] = rs.getString(6);

                i++;

            }

            i=0;

            while (rs2.next())

            {

                rows[i][6]=rs2.getString(1);

                rows[i][7]=rs2.getString(2);

                i++;

            }

            if (opt == 2) {

                this.remove(sp\_delete);

            }

            tbdelete = new JTable(rows, colnames);

            tbdelete.setRowHeight(40);

            TableColumnModel columnModel = tbdelete.getColumnModel();

            columnModel.getColumn(0).setPreferredWidth(40);

            columnModel.getColumn(1).setPreferredWidth(100);

            columnModel.getColumn(2).setPreferredWidth(100);

            columnModel.getColumn(3).setPreferredWidth(40);

            columnModel.getColumn(4).setPreferredWidth(200);

            columnModel.getColumn(5).setPreferredWidth(200);

            ListSelectionModel model = tbdelete.getSelectionModel();

            model.addListSelectionListener((ListSelectionListener) new ListSelectionListener() {

                public void valueChanged(ListSelectionEvent e) {

                    int sel = model.getMinSelectionIndex();

                    if (pnl\_delete.isVisible()) {

                        pnl\_delete.tfid3.setText("" + tbdelete.getValueAt(sel, 0));

                    } else if (pnl\_update.isVisible()) {

                        tfid.setText("" + tbdelete.getValueAt(sel, 0));

                        tffrom2.setText("" + tbdelete.getValueAt(sel, 1));

                        tfto2.setText("" + tbdelete.getValueAt(sel, 2));

                        tfmsg2.setText("" + tbdelete.getValueAt(sel, 4));

                        tfsub2.setText("" + tbdelete.getValueAt(sel, 5));

                        tftype2.setText(""+ tbdelete.getValueAt(sel, 7));

                        String val = ""+tbdelete.getValueAt(sel,6);

                        if(val.equals("Yes"))

                        {

                            l\_is.setSelectedIndex(0);

                        }

                        else

                        {

                            l\_is.setSelectedIndex(1);

                        }

                    }

                }

            });

            tbdelete.setBounds(10, 30, 275, 175);

            sp\_delete = new JScrollPane(tbdelete);

            sp\_delete.setBounds(375, 100, 570, 150);

            tbdelete.setToolTipText("Select a row to edit/delete");

            add(sp\_delete);

            con.close();

        } catch (Exception e) {

            e.printStackTrace();

        }

    }

    private void panel1() {

        // VIEW INSERT

        mbar = new MenuBar();

        setMenuBar(mbar);

        Menu file = new Menu("Options");

        file.add(item1 = new MenuItem("New data"));

        file.add(item2 = new MenuItem("Update data"));

        file.add(item3 = new MenuItem("Delete"));

        mbar.add(file); // Adding Menu to MenuBar

        item1.addActionListener(this);

        item2.addActionListener(this);

        item3.addActionListener(this);

        String bool[] = { "Yes", "No" };

        lblTitle.setText("Inbox");

        lblTitle.setBounds(160, 20, 170, 70);

        lblfrom = new JLabel("From :");

        lblto = new JLabel("To :");

        lblmsg = new JLabel("Message :");

        lblsub = new JLabel("Subject :");

        lblis\_spam = new JLabel("Spam ?");

        lbltype = new JLabel("Type :");

        tftype = new JTextField(10);

        l\_is = new JComboBox(bool);

        tffrom = new JTextField(10);

        tfto = new JTextField(10);

        tfmsg = new JTextField(10);

        tfsub = new JTextField(10);

        l\_is.setSelectedIndex(0);

        l\_is.addItemListener(this);

        btSubmit = new JButton("Submit");

        btSubmit.addActionListener(this);

        add(btSubmit);

        btSubmit.setBounds(155, 275, 75, 25);

        pnl1.setVisible(false);

        pnl\_new = new JPanel();

        pnl\_new.setLayout(new GridLayout(6, 2));

        pnl\_new.add(lblfrom);

        pnl\_new.add(tffrom);

        pnl\_new.add(lblto);

        pnl\_new.add(tfto);

        pnl\_new.add(lblmsg);

        pnl\_new.add(tfmsg);

        pnl\_new.add(lblsub);

        pnl\_new.add(tfsub);

        pnl\_new.add(lblis\_spam);

        pnl\_new.add(l\_is);

        pnl\_new.setVisible(true);

        add(pnl\_new);

        pnl\_new.setBounds(82, 90, 250, 170);

        btlogin.setVisible(false);

        setSize(1000, 425);

        lbltable = new JLabel("Table View");

        showtable(1);

        add(lbltable);

        lbltable.setBounds(600, 15, 100, 100);

        lbltable.setFont(f1);

        pnl\_new.add(lbltype);

        pnl\_new.add(tftype);

    }

    private void panel2() {

        // VIEW UPDATE

        pnl\_update = new JPanel(new GridLayout(7, 2));

        pnl\_update.setVisible(false);

        String bool2[] = { "Yes", "No" };

        lblid = new JLabel("ID");

        lblfrom2 = new JLabel("From :");

        lblto2 = new JLabel("To :");

        lblmsg2 = new JLabel("Message :");

        lblsub2 = new JLabel("Subject :");

        lblis\_spam2 = new JLabel("Spam ?");

        lbltype2 = new JLabel("Type :");

        tfid = new JTextField(10);

        tfid.setText("Suggested to update using this");

        tftype2 = new JTextField(10);

        l\_is2 = new JComboBox(bool2);

        tffrom2 = new JTextField(10);

        tfto2 = new JTextField(10);

        tfmsg2 = new JTextField(10);

        tfsub2 = new JTextField(10);

        l\_is2.setSelectedIndex(0);

        l\_is2.addItemListener(this);

        btupdate = new JButton("Update");

        add(btupdate);

        btupdate.setBounds(155, 275, 75, 25);

        btupdate.addActionListener(this);

        pnl\_update.add(lblid);

        pnl\_update.add(tfid);

        pnl\_update.add(lblfrom2);

        pnl\_update.add(tffrom2);

        pnl\_update.add(lblto2);

        pnl\_update.add(tfto2);

        pnl\_update.add(lblmsg2);

        pnl\_update.add(tfmsg2);

        pnl\_update.add(lblsub2);

        pnl\_update.add(tfsub2);

        pnl\_update.add(lblis\_spam2);

        pnl\_update.add(l\_is2);

        pnl\_update.add(lbltype2);

        pnl\_update.add(tftype2);

        tffrom2.setEditable(false);

        tfto2.setEditable(false);

        pnl\_update.setBounds(82, 90, 265, 170);

        add(pnl\_update);

    }

    private void panel3() {

        pnl\_delete = new panelDelete();

        btdelete = new JButton("Delete");

        add(btdelete);

        add(pnl\_delete);

        btdelete.setBounds(155, 250, 100, 30);

        btdelete.addActionListener(this);

    }

    public void panel\_display() {

        String opt\_display[] = { "USER\_PROFILE", "BLOCKED", "INBOX" };

        pnl\_display = new JPanel(null);

        pnl\_display.setVisible(false);

        cb\_display = new JComboBox(opt\_display);

        cb\_display.setBounds(60, 6, 120, 20);

        cb\_display.addItemListener(this);

        add(pnl\_display);

        pnl\_display.setBounds(50, 100, 300, 250);

        pnl\_display.add(cb\_display);

    }

    private void initializeComponents() {

        btlogin = new JButton("Login");

        f1 = new Font(Font.SERIF, Font.BOLD, 20);

        f2 = new Font(Font.SERIF, Font.BOLD, 10);

        f3 = new Font(Font.SERIF, Font.BOLD, 30);

        f4 = new Font(Font.SERIF, Font.PLAIN, 15);

        lblTitle = new JLabel("Spam Detection");

        btnew = new JButton("New User");

btnew.setToolTipText("Click to create a new account");

pnl1 = new JPanel();

        lbllogin = new JLabel("Username:");

        lblpass = new JLabel("Password:");

        tflogin = new JTextField(10);

        tfpass = new JPasswordField(10);

        }

    private void addComponenetsToFrame() {

        add(lblTitle);

        pnl1.setLayout(new GridLayout(2, 2, 2, 10));

        pnl1.add(lbllogin);

        pnl1.add(tflogin);

        pnl1.add(lblpass);

        pnl1.add(tfpass);

        add(pnl1);

        add(btlogin);

        add(btnew);

    }

    public void jdbcdelete()

    {

        try {

            int todel = Integer.parseInt(pnl\_delete.tfid3.getText());

            con = DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms", "it20737025", "vasavi");

            Statement stmt = con.createStatement();

            int rs1 = stmt.executeUpdate("delete from inbox where I\_ID=" + todel);

            con.close();

            if(rs1==0)

            {

                throw new Exception("Delete Unsuccessfull/No data");

            }

            JOptionPane.showMessageDialog(this, "Successfully deleted", "Delete", JOptionPane.INFORMATION\_MESSAGE);

        }

        catch(NumberFormatException e)

        {

            JOptionPane.showMessageDialog(this,"Enter a number only", "ERROR", JOptionPane.ERROR\_MESSAGE);

        }

        catch (Exception e) {

            JOptionPane.showMessageDialog(this,e.getMessage(), "ERROR", JOptionPane.ERROR\_MESSAGE);

        }

    }

    public void jdbcinsert()

    {

        try {

            int c = 0;

            Connection con = DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms", "it20737025",

                    "vasavi");

            Statement stmtcount = con.createStatement();

            ResultSet rs = stmtcount.executeQuery("select max(s.i\_id) from inbox s");

            int id=0;

            while (rs.next()) {

                id = rs.getInt(1);

            }

            c = id + 1;

             LocalDateTime myDateObj = LocalDateTime.now();

             DateTimeFormatter myFormatObj = DateTimeFormatter.ofPattern("dd-MM-yy");

            Statement stmt = con.createStatement();

            String dat = myDateObj.format(myFormatObj);

            int s = stmt.executeUpdate("insert into inbox values ("+c+",'"+tffrom.getText()+"','"+tfto.getText()+"','"+dat+"')");

            if(s==0)throw new Exception("Insert failed");

            s =stmt.executeUpdate("insert into message values ("+c+",'"+tfmsg.getText()+"','"+tfsub.getText()+"')");

            if(s==0)throw new Exception("Insert failed");

            String val = (String) l\_is.getSelectedItem();

            s = stmt.executeUpdate("insert into spam values ("+c+",'"+val+"','"+tftype.getText()+"')");

            if(s==0)throw new Exception("Insert failed");

            JOptionPane.showMessageDialog(this, "Successfully inserted", "Insert", JOptionPane.INFORMATION\_MESSAGE);

            con.close();

        }

        catch (SQLIntegrityConstraintViolationException e1) {

            JOptionPane.showMessageDialog(this, "Enter a valid email id", "ERROR", JOptionPane.ERROR\_MESSAGE);

        }

        catch (Exception e1) {

            JOptionPane.showMessageDialog(this, e1.getMessage(), "ERROR", JOptionPane.ERROR\_MESSAGE);

        }

    }

    public void actionPerformed(ActionEvent e) {

        if(e.getSource()==button)

        {

            dispose();

            new FirstPg();

        }

        if(e.getActionCommand().equals("Create"))

        {

            createuser();

        }

        if(e.getActionCommand().equals("<-"))

        {

            pnl1.setVisible(true);

            btlogin.setVisible(true);

            pnl\_user.setVisible(false);

            btcreate.setVisible(false);

            btback.setVisible(false);

            btnew.setVisible(true);

        }

        if(e.getActionCommand().equals("New User"))

        {

            pnl1.setVisible(false);

            btlogin.setVisible(false);

            pnl\_user.setVisible(true);

            btcreate.setVisible(true);

            btnew.setVisible(false);

            btback.setBounds(20, 20, 50, 25);

            btback.addActionListener(this);

            btback.setVisible(true);

        }

        if (e.getActionCommand().equals("Submit")) {

            jdbcinsert();

            showtable(2);

        }

        if (e.getActionCommand().equals("Display")) {

            pnl\_delete.setVisible(false);

            pnl\_new.setVisible(false);

            pnl\_update.setVisible(false);

            pnl\_display.setVisible(true);

            btSubmit.setVisible(false);

            btdelete.setVisible(false);

            btupdate.setVisible(false);

            lblTitle.setText("Display");

            lblTitle.setFont(f3);

            btnew.setVisible(false);

        }

        if (e.getActionCommand().equals("Update")) {

            jdbcupdate();

        }

        if (e.getActionCommand().equals("Update data")) {

            pnl\_delete.setVisible(false);

            pnl\_new.setVisible(false);

            pnl\_update.setVisible(true);

            pnl\_display.setVisible(false);

            btSubmit.setVisible(false);

            btdelete.setVisible(false);

            btupdate.setVisible(true);

            btupdate.setToolTipText("Click to update data");

            lblTitle.setText("Update");

            lblTitle.setFont(f3);

            btnew.setVisible(false);

        }

        if (e.getSource() == btdelete) {

            jdbcdelete();

            showtable(2);

        } else if (e.getActionCommand().equals("Delete")) {

            pnl\_new.setVisible(false);

            pnl\_update.setVisible(false);

            pnl\_delete.setVisible(true);

            pnl\_display.setVisible(false);

            btSubmit.setVisible(false);

            btupdate.setVisible(false);

            btdelete.setVisible(true);

            btdelete.setToolTipText("Click to delete data");

            lblTitle.setText("Delete");

            lblTitle.setFont(f3);

            this.setSize(1000, 450);

            btnew.setVisible(false);

        }

        if (e.getActionCommand().equals("New data")) {

            pnl\_new.setVisible(true);

            pnl\_update.setVisible(false);

            pnl\_delete.setVisible(false);

            pnl\_display.setVisible(false);

            btSubmit.setVisible(true);

            btupdate.setVisible(false);

            btdelete.setVisible(false);

            btSubmit.setToolTipText("Click to insert data");

            lblTitle.setText("Insert");

            lblTitle.setFont(f3);

        }

        if (e.getActionCommand().equals("Login")) {

            String s1 = tflogin.getText();

            String s2 = tfpass.getText();

            button = new JButton();

        try {

        Image img = ImageIO.read(getClass().getResource("icons8-login-rounded-30.png"));

        button.setIcon(new ImageIcon(img));

        } catch (Exception ex) {

            JOptionPane.showMessageDialog(this, ex.getMessage(), "ERROR", JOptionPane.ERROR\_MESSAGE);

        }

            add(button);

            button.setBounds(20, 300, 60, 40);

            button.addActionListener(this);

              try{

            con=DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it20737025","vasavi");

            Statement stmt=con.createStatement();

            ResultSet rs=stmt.executeQuery("select \* from login where email='"+s1+"'");

              if(rs.next())

              {

              if(s2.equals(rs.getString(2)))

              {

                panel1(); panel2(); panel3(); panel\_display();

                btdelete.setVisible(false); btupdate.setVisible(false);

                pnl\_new.setBackground(r1); pnl\_update.setBackground(r1);

                pnl\_delete.setBackground(r1);

                btnew.setVisible(false);

            }

            else

            {

                throw new Exception("Enter valid credentials");

            }

            }

            else

            {

                throw new Exception("Email not found");

            }

              con.close();

            }

            catch(Exception e1)

            {

                JOptionPane.showMessageDialog(this, e1.getMessage(), "ERROR", JOptionPane.ERROR\_MESSAGE);

            }

        }

    }

    private void createuser() {

        String s1=pnl\_user.getTfemail().getText();

        String s2=pnl\_user.getTfname().getText();

        String s3=pnl\_user.getTfdob().getText();

        String s4=pnl\_user.getTfphn().getText();

        String s5=pnl\_user.getTfpass().getText();

        try {

            int c = 0;

            Connection con = DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms", "it20737025",

                    "vasavi");

            Statement stmt = con.createStatement();

            stmt.executeUpdate("insert into user\_profile values ('"+s1+"','"+s2+"','"+s3+"','"+s4+"')");

            stmt.executeUpdate("insert into login values ('"+s1+"','"+s5+"')");

            JOptionPane.showMessageDialog(this, "Successfully inserted", "Insert", JOptionPane.INFORMATION\_MESSAGE);

            con.close();

        } catch (Exception e1) {

            JOptionPane.showMessageDialog(this, e1.getMessage(), "ERROR", JOptionPane.ERROR\_MESSAGE);

        }

    }

       private void jdbcupdate() {

        try {

            Connection con = DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms", "it20737025",

                    "vasavi");

            int val1 = Integer.parseInt(tfid.getText());

            Statement stmt = con.createStatement();

            int done=0;

            if (tfmsg2.getText() != "") {

                String pt = "Update Message set content='" + tfmsg2.getText() + "' where id=" + val1;

                int insert = stmt

                        .executeUpdate("Update Message set content='" + tfmsg2.getText() + "' where id=" + val1);

                done = insert;

                if(done==0) throw new Exception("Update Failed\nMight be due to ID D.N.E");

            }

            if (tfsub2.getText() != "") {

                int insert1 = stmt.executeUpdate("Update Message set subject='" + tfsub2.getText() + "' where id=" + val1);

                done = insert1;

                if(done==0) throw new Exception("Update Failed");

            }

            if(tftype.getText()!="")

            {

                int insert2 = stmt.executeUpdate("Update Spam set Type='" + tftype2.getText() + "' where i\_id=" + val1);

                done = insert2;

                if(done==0) throw new Exception("Update Failed");

            }

                String cbox = (String) l\_is2.getSelectedItem();

                System.out.println(cbox);

                if(cbox.equals("No"))

                {

                    int insert3 = stmt.executeUpdate("Update Spam set IS\_SPAM='" + cbox + "' where i\_id=" + val1);

                    int insert2 = stmt.executeUpdate("Update Spam set Type='" + tftype2.getText() + "' where i\_id=" + val1);

                    done = insert2+insert3;

                    if(done==0) throw new Exception("Update Failed");

                }

                else

                {

                int insert3 = stmt.executeUpdate("Update Spam set IS\_SPAM='" + cbox + "' where i\_id=" + val1);

                System.out.println("Update Spam set IS\_SPAM='" + cbox + "' where i\_id=" + val1);

                done = insert3;

                if(done==0) throw new Exception("Update Failed");

                }

            if(done==1)

            {

            JOptionPane.showMessageDialog(null,"Succesfully Updated","Update !!", JOptionPane.INFORMATION\_MESSAGE);

            }

            con.close();

        }

        catch(NumberFormatException e)

        {

            JOptionPane.showMessageDialog(this,"id : Enter a number only", "ERROR", JOptionPane.ERROR\_MESSAGE);

        }

        catch (Exception e1) {

            JOptionPane.showMessageDialog(null,e1.getMessage(),"alert", JOptionPane.ERROR\_MESSAGE);

        }

        showtable(2);

    }

    public void itemStateChanged(ItemEvent e) {

        if (cb\_display.getSelectedItem() == "BLOCKED") {

            tbdisplay.setVisible(false);

            sp\_all.setVisible(false);

            display\_blocked();

        }

        String s = e.paramString();

        if (l\_is.getSelectedItem() == "Yes") {

            lbltype.setVisible(true);

            tftype.setVisible(true);

        } else {

            tftype.setText("");

            lbltype.setVisible(false);

            tftype.setVisible(false);

        }

        if (l\_is2.getSelectedItem() == "Yes") {

            lbltype2.setVisible(true);

            tftype2.setVisible(true);

        } else {

            lbltype2.setVisible(false);

            tftype2.setVisible(false);

        }

    }

}

public class MyClassSwing {

    public static void main(String... args) {

        try

        {

        new FirstPg();

        }

        catch(Exception e)

        {

            System.out.println(e);

        }

    }

}

panelDelete.java

import java.awt.Font;

import javax.swing.JButton;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JTextField;

public class panelDelete extends JPanel{

    JButton btdelete;

    private JLabel lblid3;

    public JTextField tfid3;

    private Font f4;

    public panelDelete()

    {

        f4 = new Font(Font.SERIF, Font.PLAIN, 15);

        setVisible(false);

        btdelete = new JButton("Delete");

        lblid3 = new JLabel("ID :");

        tfid3 = new JTextField(15);

        tfid3.setText("suggested for deletion");

        add(lblid3);

        add(tfid3);

        lblid3.setBounds(2, 2, 10, 10);

        tfid3.setBounds(2, 2, 10, 10);

//      add(btdelete);

//      add(pnl\_delete);

        tfid3.setFont(f4);

        lblid3.setFont(f4);

        setBounds(20, 150, 300, 400);

        setAutoscrolls(true);

    }

    /\*\*

     \*

     \*/

    private static final long serialVersionUID = 1L;

}

paneluser.java

import java.awt.GridLayout;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JTextField;

public class paneluser extends JPanel {

    private JLabel lblemail;

    private JLabel lblname;

    private JLabel lbldob;

    private JLabel lblphn;

    private JTextField tfemail;

    private JTextField tfname;

    private JTextField tfphn;

    private JTextField tfdob;

    private JTextField tfpass;

    private JLabel lblpass;

    public paneluser()

    {

        this.setLayout(new GridLayout(5,5,2,2));

        lblemail = new JLabel("Email:");

        lblname = new JLabel("UserName:");

        lbldob = new JLabel("DOB:");

        lblphn = new JLabel("Phone:");

        lblpass = new JLabel("Pass:");

        tfemail = new JTextField(10);

        tfname = new JTextField(10);

        tfdob = new JTextField(10);

        tfphn = new JTextField(10);

        tfpass = new JTextField(10);

        addComponents();

    }

    private void addComponents() {

        add(lblemail);

        add(getTfemail());

        add(lblname);

        add(getTfname());

        add(lbldob);

        add(getTfdob());

        add(lblphn);

        add(getTfphn());

        add(lblpass);

        add(getTfpass());

    }

    public JTextField getTfemail() {

        return tfemail;

    }

    public JTextField getTfname() {

        return tfname;

    }

    public JTextField getTfdob() {

        return tfdob;

    }

    public JTextField getTfphn() {

        return tfphn;

    }

    public JTextField getTfpass() {

        return tfpass;

    }

    /\*\*

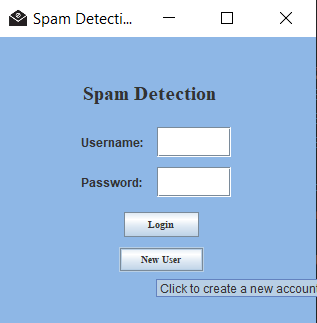
     \*

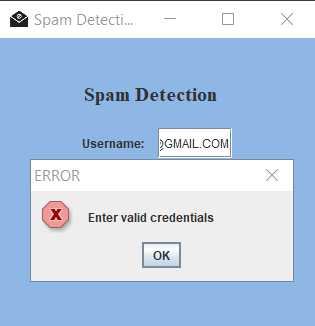
     \*/

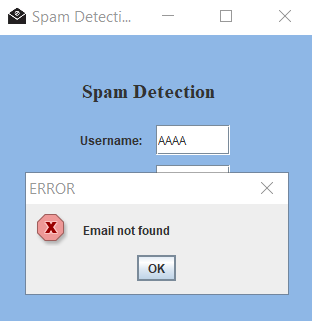
    private static final long serialVersionUID = 1L;

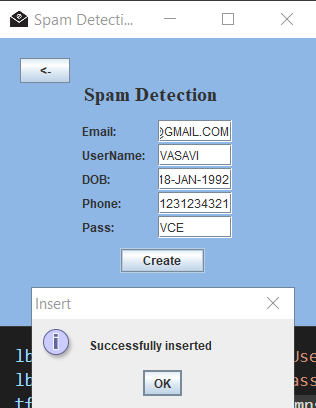
}

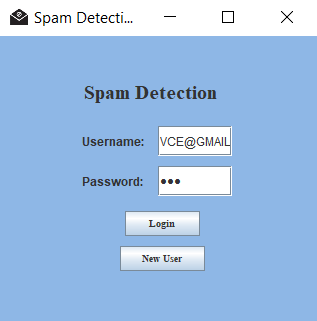
# Screenshots

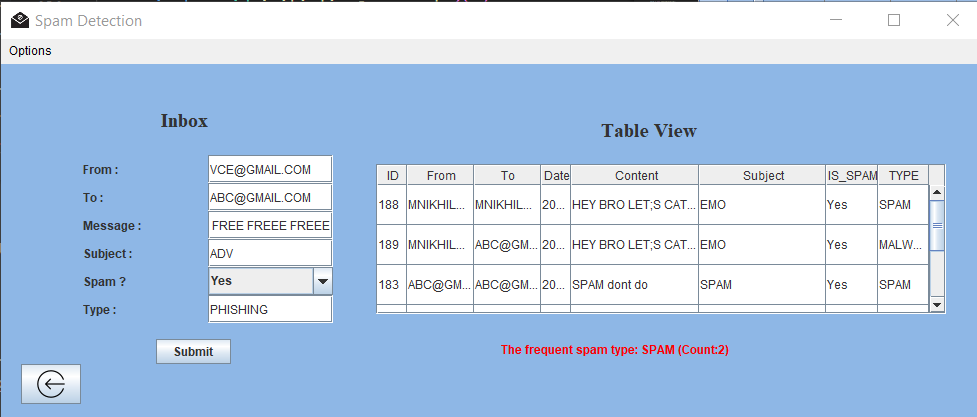


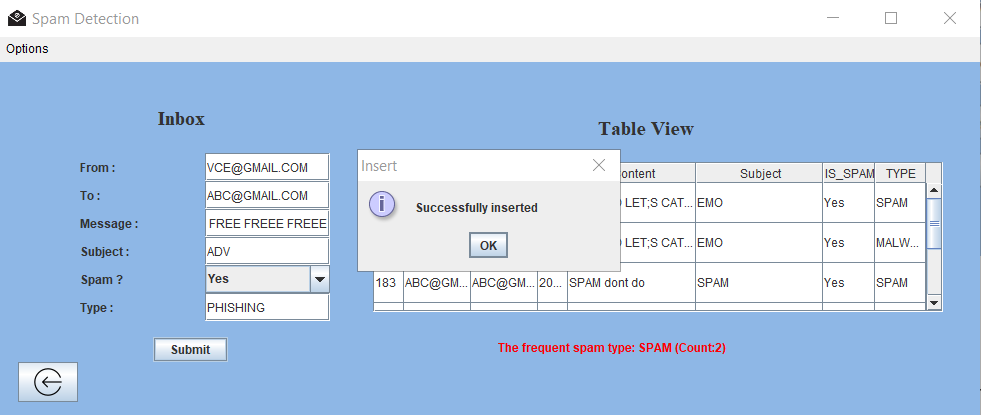


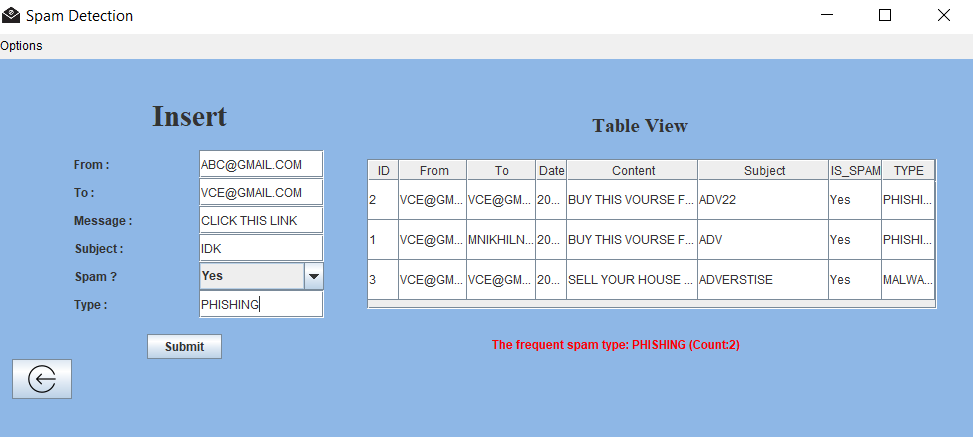


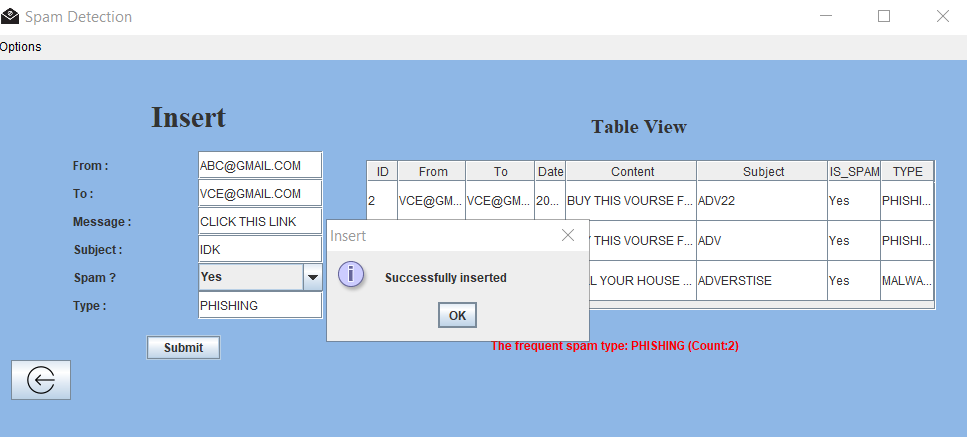


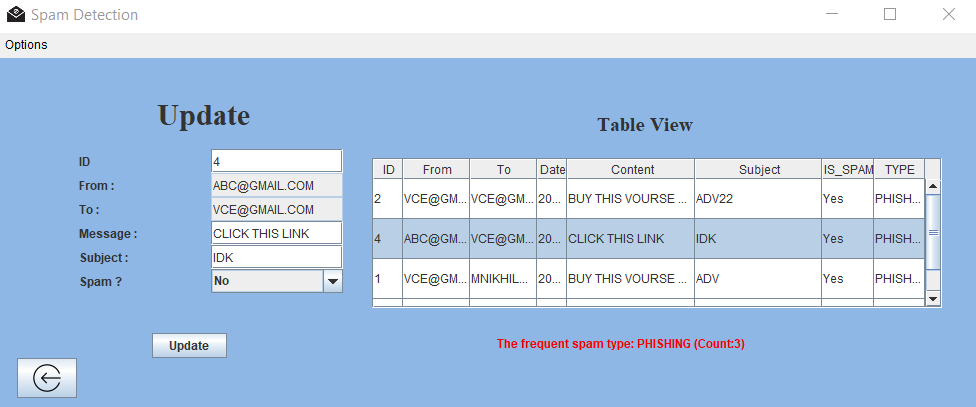




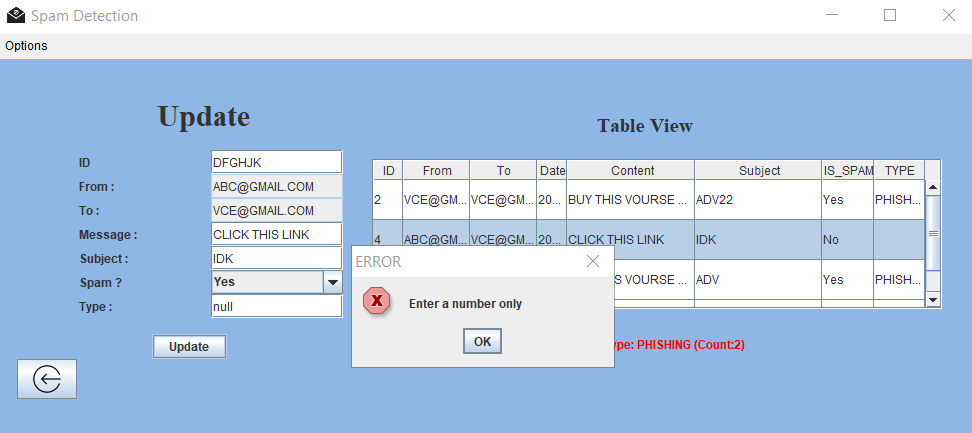






**

**

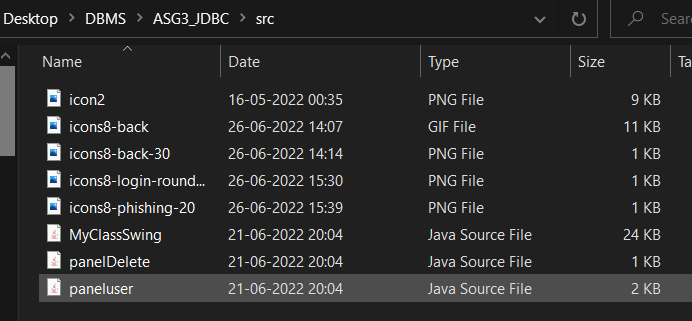


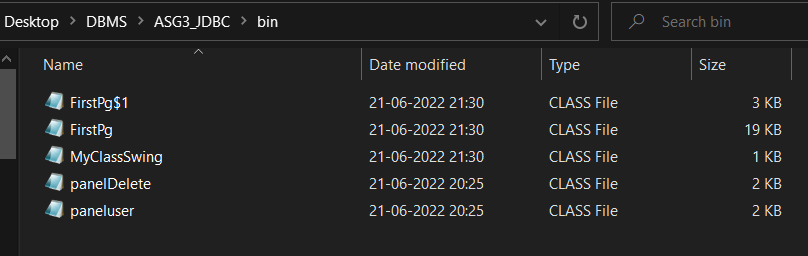


GIT HUB LINK

<https://github.com/Nikhil1357/Spam_Detection_DBMS>

Folder Structure





Thank You