

JAVA AWT BASED- NETWORK CONNECTION MANAGEMENT SYSTEM - SQL CONNECTIVITY USING JDBC

A

Report

*Submitted in partial fulfilment of the
Requirements for the award of the Degree of*

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

By

A.RUCHITA <1602-18-737-064>

Under the guidance of B.Leelavathy



**Department of Information Technology
Vasavi College of Engineering (Autonomous)
(Affiliated to Osmania University)
Ibrahimbagh, Hyderabad-31**

2019

BONAFIDE CERTIFICATE

This is to certify that the project report titled “**VCE NETWORK CONNECTION MANAGEMENT SYSTEM**” project work of Miss.Ruchita Ananthula bearing Roll.no:1602-18-737-064 who carried out this project under my supervision in the IV Semester for the academic year 2019-2020.

Signature

External examine

Signature

Internal examine

ABSTRACT:-

The Network Connection Management System is a Web-based system that provides a single-stop for registration and maintenance of the network connection database of the campus network. The database contains both data (wired and wireless) and voice network connections in the Main Campus and the Student Residences. Network management system (NMS) is important both in ensuring the correct operation of network devices and in maintaining the services that run on them. This project has total of 11 tables .It describes how the network is being connected in our college across the various blocks. When you enter the data it is stored in the data base and is displayed as of when it is needed.

AIM:

To create a **Java GUI based NETWORK CONNECTION MANAGEMENT SYSTEM** which takes the values like: computer ID, computer name , manufacturer , type , count, routers speed , username , website , block name , hod , server ipaddress , operating system name , version etc from the user. These values are to be updated in the database using **JDBC connectivity**.

INTRODUCTION

Requirements:

List of tables:

- Internet
- Computers
- Routers
- Block
- Server
- operating_system
- contains
- has
- are_having
- provides_network_to
- connected_to

List of attributes with their domain types:-

Internet:

Mac address : mac_address-varchar2(20)

HTML: html -varchar2(30)

Service provider: serv_provider- varchar2(20));

Computers:

Id of the computer: cid- varchar2(20)

Type of the computer: type - varchar2(20)

Count of computers: count -number

Manufacturer name: manufacturer - varchar2(20)

Type of model: model-varchar2(20)

Routers:

Website name :website -varchar2(50)

Speed of the router:speed- varchar2(10)

Model :model- varchar2 (20)

Username: username -varchar2(20)

Company :company- varchar2 (20)

Block:

Name of the block:bname-varchar2(20)

Name of the HOD:hod -varchar2(20)

Branch of the block:branch varchar2(20)

Server:

Ipaddress : ipaddress varchar2(20)

Operating_system:

Name of the operating system : osname- varchar2(20)

Version: version -varchar2(20)

Vendor: vendor-varchar2(20)

ARCHITECTURE AND TECHNOLOGY :

Software used:

Java Eclipse, Oracle 11g Database, Java SE version 7, SQL*Plus.

Java AWT:

Java AWT (Abstract Window Toolkit) is an API to develop GUI or window-based applications in java.

Java AWT components are platform-dependent i.e. components are displayed according to the view of operating system. AWT is heavyweight i.e. its components are using the resources of OS.

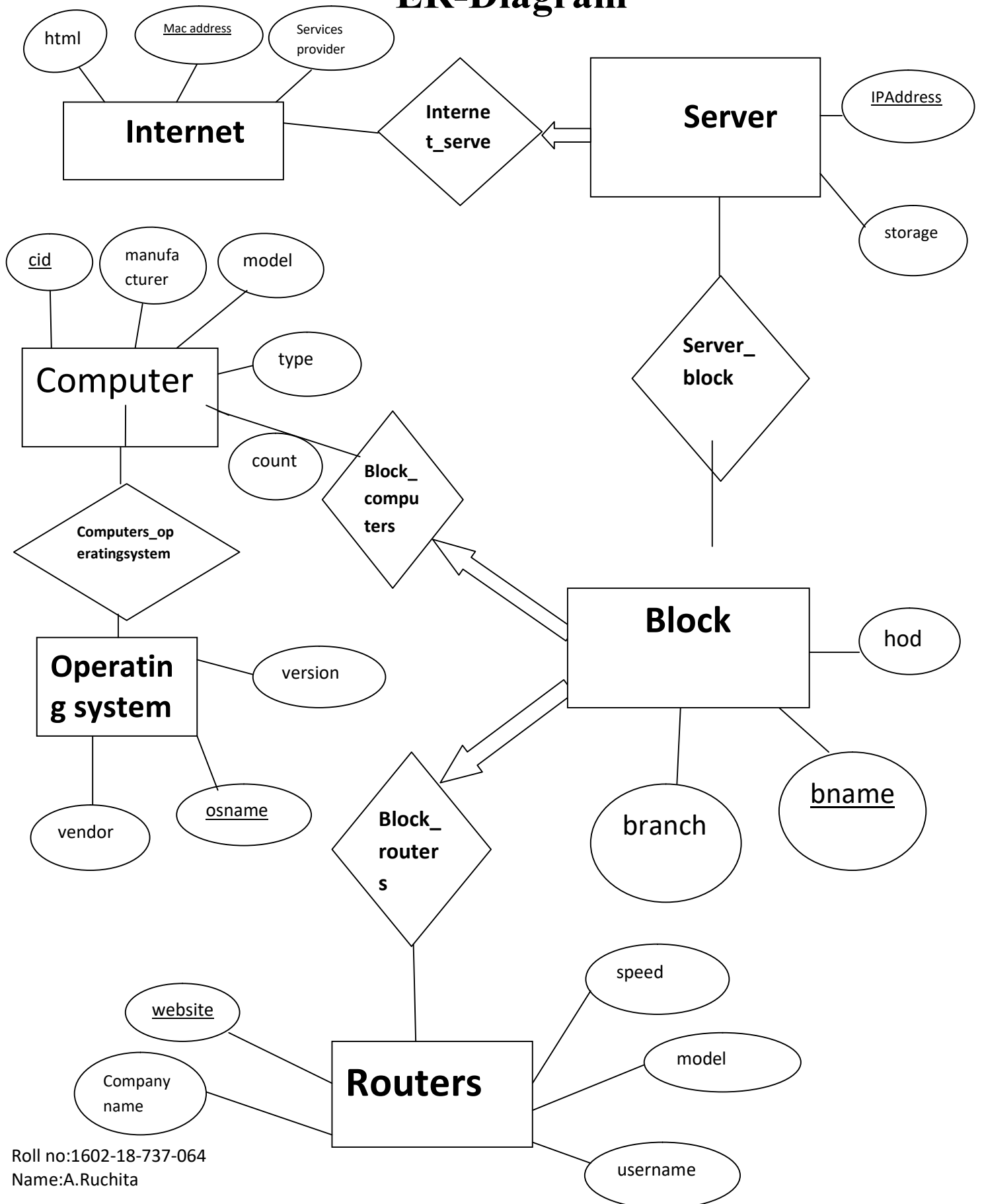
The java.awt package provides classes for AWT API such as TextField, Label, TextArea, RadioButton, CheckBox, Choice, List etc.

SQL:

Structure Query Language(SQL) is a database query language used for storing and managing data in Relational DBMS. SQL was the first commercial language introduced for E.F Codd's **Relational** model of database. Today almost all RDBMS (MySQL, Oracle, Infomix, Sybase, MS Access) use **SQL** as the standard database query language. SQL is used to perform all types of data operations in RDBMS.

DESIGN

ER-Diagram



DDL COMMANDS

create table **internet**(

mac_address varchar2(20) primary key,html varchar2(30),serv_provider varchar2(20));

create table **computers**(

cid varchar2(20) primary key, type varchar2(20), count number, manufacturer varchar2(20),model varchar2(20)) ;

create table **routers** (

website varchar2(50) primary key ,speed varchar2(10),model varchar2 (20), username varchar2(20),company varchar2 (20)) ;

create table **Block**(

bname varchar2(20)primary key ,hod varchar2(20),branch varchar2(20));

create table **server** (ipaddress varchar2(20)primary key);

create table **operating_system**(

osname varchar2(20) , version varchar2(20), vendor varchar2(20), primary key(osname));

create table **Block_computer**(

cid varchar2(20), bname varchar2(20),foreign key(cid) references computers(cid), foreign key (bname) references block(bname),primary key(cid,bname)) ;

create table **Computers_operatingsystem**(

cid varchar2(20), osname varchar2(20),foreign key (cid) references computers, foreign key (osname) reference operating_system) ;

create table **Block_routers**(

website varchar2(50), bname varchar2(20), foreign key(website) references routers, foreign key(bname) references block);

create table **Server_block**(

bname varchar2(20),ipaddress varchar(20),foreign key(ipaddress) references server, foreign key(bname) references block);

create table **Internet_server**(
 mac varchar2(20),ipadd varchar2(20),foreign key(mac) references internet, foreign
 key(ipadd) references server);

```
SQL> desc internet;
Name                                         Null?    Type
-----
HTML                                         VCHAR2(30)
SERV_PROVIDER                               VCHAR2(20)
MAC_ADDRESS                                NOT NULL VCHAR2(20)

SQL> desc server;
Name                                         Null?    Type
-----
IPADDRESS                                NOT NULL VCHAR2(20)

SQL> desc block;
Name                                         Null?    Type
-----
BNAME                                     NOT NULL VCHAR2(20)
HOD                                       VCHAR2(20)
BRANCH                                   VCHAR2(20)

SQL> desc routers
Name                                         Null?    Type
-----
SPEED                                     VCHAR2(10)
MODEL                                    VCHAR2(20)
USERNAME                                 VCHAR2(20)
COMPANY                                 VCHAR2(20)
WEBSITE                                NOT NULL VCHAR2(50)

SQL> desc computers
Name                                         Null?    Type
-----
TYPE                                     VCHAR2(20)
COUNT                                  NUMBER
MANUFACTURER                             VCHAR2(20)
MODEL                                    VCHAR2(20)
CID                                     NOT NULL VCHAR2(20)
```

```
ORA-04043: object operating_system does not exist

SQL> desc operating_system
Name                                         Null?    Type
-----
OSNAME                                    NOT NULL VCHAR2(20)
VERSION                                 VCHAR2(20)
VENDOR                                 VCHAR2(20)

SQL> desc connected_to
Name                                         Null?    Type
-----
MAC                                       VCHAR2(20)
IPADD                                    VCHAR2(20)

SQL> desc provides_network_to
Name                                         Null?    Type
-----
BNAME                                    VCHAR2(20)
IPADDRESS                               VCHAR2(20)

SQL> desc contains
Name                                         Null?    Type
-----
CID                                    NOT NULL VCHAR2(20)
BNAME                                   NOT NULL VCHAR2(20)

SQL> desc has
Name                                         Null?    Type
-----
CID                                    VCHAR2(20)
OSNAME                                 VCHAR2(20)

SQL> desc are_having
Name                                         Null?    Type
-----
WEBSITE                                VCHAR2(50)
BNAME                                 VCHAR2(20)
```

Java-SQL Connectivity using JDBC:

Java Database Connectivity (JDBC) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation. It provides methods to query and update data in a database and is oriented towards relational databases.

The connection to the database can be performed using Java programming (JDBC API) as:

```
private void connToDb() {
    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        connection =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1522:xe","rachana","vasavi");
        statement = connection.createStatement();

    } catch (SQLException connectException) {
        System.out.println(connectException.getMessage());
        System.out.println(connectException.getSQLState());
        System.out.println(connectException.getErrorCode());
        System.exit(1);
    }
    catch (Exception e)
    {
        System.err.println("Unable to find and load driver");
        System.exit(1);
    }
}
```

Thus, the connection from Java to Oracle database is performed and therefore, can be used for updating tables in the database directly.

Table Created in SQL for above mentioned purpose is as:

create table **computers**

```
(  
cid varchar2(20) primary key ,  
type varchar2(20) ,  
count number ,  
manufacturer varchar2(20) ,  
model varchar2(20)  
);
```

Program to insert computers:

```
package java_ass2;  
  
import java.awt.*;  
  
import java.awt.event.*;  
  
import java.sql.*;  
  
public class InsertComputers extends Frame  
{  
  
    Button insertComputersButton;  
  
    TextField cidText, typeText, countText, manufacturerText;  
  
    TextArea errorText;  
  
    Connection connection;  
  
    Statement statement;  
  
    public InsertComputers()  
    {  
  
        try  
        {  
  
            Class.forName ("oracle.jdbc.driver.OracleDriver");  
  
        }  
  
    }  
  
}
```

DBMS MINIPROJECT

TITLE:VCE Network Connection Management System

```
        catch (Exception e)
        {
            System.err.println("Unable to find and load driver");
            System.exit(1);
        }
        connectToDB ();
    }

    public void connectToDB()
    {
        try
        {
            connection=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","ruchi","04032001");
            statement = connection.createStatement();

        }
        catch (SQLException connectException)
        {
            System.out.println(connectException.getMessage());
            System.out.println(connectException.getSQLState());
            System.out.println(connectException.getErrorCode());
            System.exit(1);
        }
    }

    public void buildGUI()
    {
        //Handle Insert Account Button
        insertComputersButton = new Button("Insert Computers");
        insertComputersButton.addActionListener(new ActionListener()
        {
```

```
        public void actionPerformed(ActionEvent e)
        {
            try
            {
                String query= "INSERT INTO Computers VALUES(" + cidText.getText() + ", " +
"" + typeText.getText() + ", " + countText.getText() + ", " + manufacturerText.getText() + ")";

                int i = statement.executeUpdate(query);

                errorText.append("\nInserted " + i + " rows successfully");
            }
            catch (SQLException insertException)
            {
                displaySQLErrors(insertException);
            }
        }
    });

    cidText = new TextField(15);
    typeText = new TextField(15);
    countText = new TextField(15);
    manufacturerText = new TextField(15);


    errorText = new TextArea(10, 40);
    errorText.setEditable(false);


    Panel first = new Panel();
    first.setLayout(new GridLayout(4, 2));
    first.add(new Label("Computer ID:"));
    first.add(cidText);
    first.add(new Label("Type:"));
    first.add(typeText);
    first.add(new Label("Count:"));
    first.add(countText);
    first.add(new Label("Manufacturer:"));
    first.add(manufacturerText);
```

DBMS MINIPROJECT

TITLE:VCE Network Connection Management System

```
        first.add(countText);

        first.add(new Label("Manufacturer:"));

        first.add(manufacturerText);

        first.setBounds(125,90,200,100);


        Panel second = new Panel(new GridLayout(4, 1));

        second.add(insertComputersButton);

        second.setBounds(125,220,150,100);


        Panel third = new Panel();

        third.add(errorText);

        third.setBounds(125,320,300,200);


        setLayout(null);


        add(first);

        add(second);

        add(third);


        setTitle("New Computers Creation");

        setSize(500, 600);

        setVisible(true);
    }

    private void displaySQLExceptions(SQLException e)
    {

        errorText.append("\nSQLException: " + e.getMessage() + "\n");

        errorText.append("SQLState:    " + e.getSQLState() + "\n");

        errorText.append("VendorError: " + e.getErrorCode() + "\n");

    }


    public static void main(String[] args)
```

Roll no:1602-18-737-064

Name:A.Ruchita

```
{  
  
    InsertComputers cc = new InsertComputers();  
    cc.addWindowListener(new WindowAdapter(){  
        public void windowClosing(WindowEvent e)  
        {  
            System.exit(0);  
        }  
    });  
    cc.buildGUI();  
}  
}
```

Program to update computers:

```
package java_ass2;  
  
import java.awt.*;  
import java.awt.event.*;  
import java.sql.*;  
  
public class ViewComputers extends Frame  
{  
  
    Button updateComputersButton;  
    List ComputersList;  
    TextField cidText, typeText, countText, manufacturerText;  
    TextArea errorText;  
    Connection connection;  
    Statement statement;  
    ResultSet rs;  
  
    public ViewComputers()
```

```
{
    try
    {
        Class.forName("oracle.jdbc.driver.OracleDriver");
    }
    catch (Exception e)
    {
        System.err.println("Unable to find and load driver");
        System.exit(1);
    }
    connectToDB();
}

public void connectToDB()
{
    try
    {
        connection =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","ruchi","04032001
");

        statement = connection.createStatement();
    }
    catch (SQLException connectException)
    {
        System.out.println(connectException.getMessage());
        System.out.println(connectException.getSQLState());
        System.out.println(connectException.getErrorCode());
    }
}
```



```
        System.exit(1);
    }
}

private void loadComputers()
{
    try
    {
        rs = statement.executeQuery("SELECT * FROM Computers");
        while (rs.next())
        {
            ComputersList.add(rs.getString("CID"));
        }
    }
    catch (SQLException e)
    {
        displaySQLErrors(e);
    }
}

public void buildGUI()
{
    ComputersList = new List(6);
    loadComputers();
    add(ComputersList);

    //When a list item is selected populate the text fields
```

```
ComputersList.addItemListener(new ItemListener()
{
    public void itemStateChanged(ItemEvent e)
    {
        try
        {
            rs = statement.executeQuery("SELECT * FROM
Computers");

            while (rs.next())
            {
                if
(rs.getString("CID").equals(ComputersList.getSelectedItem()))
                    break;
            }
            if (!rs.isAfterLast())
            {
                cidText.setText(rs.getString("CID"));
                typeText.setText(rs.getString("TYPE"));
                countText.setText(rs.getString("COUNT"));

                manufacturerText.setText(rs.getString("MANUFACTURER"));
            }
        }
        catch (SQLException selectException)
        {
            displaySQLErrors(selectException);
        }
    }
}
```

```
        }

    });

    //Handle Update Computers Button

    updateComputersButton = new Button("Update Computers");
    updateComputersButton.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent e)
        {
            try
            {
                Statement statement = connection.createStatement();
                int i = statement.executeUpdate("UPDATE Computers "
                + "SET COUNT=" + countText.getText()
                + " WHERE cid = " + ComputersList.getSelectedItem()
                + "");

                errorText.append("\nUpdated " + i + " rows
                successfully");

                ComputersList.removeAll();
                loadComputers();
            }
            catch (SQLException insertException)
            {
                displaySQLErrors(insertException);
            }
        }
    });
```

```
cidText = new TextField(15);
cidText.setEditable(false);
typeText = new TextField(15);
typeText.setEditable(false);
countText = new TextField(15);
manufacturerText = new TextField(15);
manufacturerText.setEditable(false);

errorText = new TextArea(10, 40);
errorText.setEditable(false);

Panel first = new Panel();
first.setLayout(new GridLayout(4, 2));
first.add(new Label("Computer ID:"));
first.add(cidText);
first.add(new Label("Type:"));
first.add(typeText);
first.add(new Label("Count:"));
first.add(countText);
first.add(new Label("Manufacturer:"));
first.add(manufacturerText);

Panel second = new Panel(new GridLayout(4, 1));
second.add(updateComputersButton);
```

```
        Panel third = new Panel();
        third.add(errorText);

        add(first);
        add(second);
        add(third);

        setTitle("Update Computers");
        setSize(500, 600);
        setLayout(new FlowLayout());
        setVisible(true);
    }
    private void displaySQLExceptions(SQLException e)
    {
        errorText.append("\nSQLException: " + e.getMessage() + "\n");
        errorText.append("SQLState:    " + e.getSQLState() + "\n");
        errorText.append("VendorError: " + e.getErrorCode() + "\n");
    }
    public static void main(String[] args)
    {
        ViewComputers upb = new ViewComputers();

        upb.addWindowListener(new WindowAdapter(){
            public void windowClosing(WindowEvent e)
```

```
        {  
            System.exit(0);  
        }  
    });  
  
    upb.buildGUI();  
}  
}
```

Program to delete computers:

```
package java_ass2;  
  
import java.awt.*;  
import java.awt.event.*;  
import java.sql.*;  
  
public class DeleteComputers extends Frame  
{  
    Button DeleteComputersButton;  
    List ComputersIDList;  
    TextField cidText, typeText, countText, manufacturerText;  
    TextArea errorText;  
    Connection connection;  
    Statement statement;  
    ResultSet rs;  
  
    public DeleteComputers()
```

```
{

    try

    {

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

    }

    catch (Exception e)

    {

        System.err.println("Unable to find and load driver");

        System.exit(1);

    }

    connectToDB();

}

public void connectToDB()

{

    try

    {

        connection =

DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","ruchi","04032001

");

        statement = connection.createStatement();

    }

    catch (SQLException connectException)

    {

        System.out.println(connectException.getMessage());

    }

}
```

```
        System.out.println(connectException.getSQLState());
        System.out.println(connectException.getErrorCode());
        System.exit(1);
    }
}

private void loadComputers()
{
    try
    {
        rs = statement.executeQuery("SELECT * FROM computers");
        while (rs.next())
        {
            ComputersIDList.add(rs.getString("CID"));
        }
    }
    catch (SQLException e)
    {
        displaySQLErrors(e);
    }
}

public void buildGUI()
{
    ComputersIDList = new List(10);
```



```
loadComputers();  
add(ComputersIDList);  
  
//When a list item is selected populate the text fields  
ComputersIDList.addItemListener(new ItemListener()  
{  
    public void itemStateChanged(ItemEvent e)  
    {  
        try  
        {  
            rs = statement.executeQuery("SELECT * FROM  
computers");  
            while (rs.next())  
            {  
                if  
(rs.getString("CID").equals(ComputersIDList.getSelectedItem()))  
                    break;  
            }  
            if (!rs.isAfterLast())  
            {  
                cidText.setText(rs.getString("CID"));  
                typeText.setText(rs.getString("TYPE"));  
                countText.setText(rs.getString("COUNT"));  
  
                manufacturerText.setText(rs.getString("MANUFACTURER"));  
            }  
        }  
    }  
});
```

```
        catch (SQLException selectException)
        {
            displaySQLErrors(selectException);
        }
    }
});

//Handle Delete Computers Button
DeleteComputersButton = new Button("Delete Computers");
DeleteComputersButton.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            Statement statement = connection.createStatement();
            int i = statement.executeUpdate("DELETE FROM
computers WHERE CID = '" + ComputersIDList.getSelectedItem()+"'");
            errorText.append("\nDeleted " + i + " rows
successfully");

            cidText.setText(null);
            typeText.setText(null);
            countText.setText(null);
            manufacturerText.setText(null);
            ComputersIDList.removeAll();
            loadComputers();
        }
        catch (SQLException ex)
        {
            displaySQLErrors(ex);
        }
    }
});
```

```
        }  
        catch (SQLException deleteException)  
        {  
            displaySQLErrors(deleteException);  
        }  
    }  
});
```

```
cidText = new TextField(15);  
typeText = new TextField(15);  
countText = new TextField(15);  
manufacturerText = new TextField(15);
```

```
errorText = new TextArea(10, 40);  
errorText.setEditable(false);
```

```
Panel first = new Panel();  
first.setLayout(new GridLayout(4, 2));  
first.add(new Label("Computers ID:"));  
first.add(cidText);  
first.add(new Label("Type:"));  
first.add(typeText);  
first.add(new Label("Count:"));  
first.add(countText);  
first.add(new Label("Manufacturer:"));
```

```
        first.add(manufacturerText);

        Panel second = new Panel(new GridLayout(4, 1));
        second.add(DeleteComputersButton);

        Panel third = new Panel();
        third.add(errorText);

        add(first);
        add(second);
        add(third);

        setTitle("Remove computers");
        setSize(450, 600);
        setLayout(new FlowLayout());
        setVisible(true);

    }

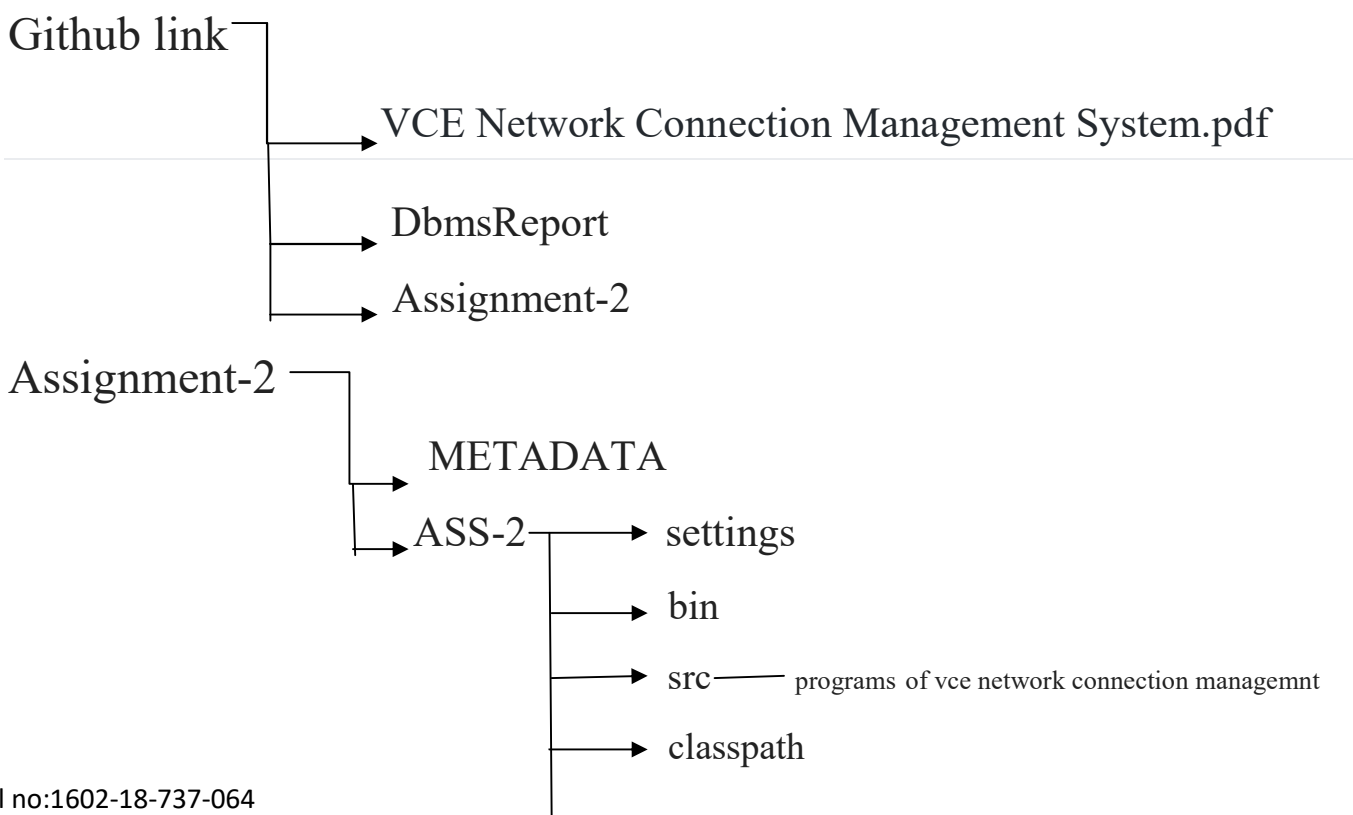
    private void displaySQLExceptions(SQLException e)
    {
        errorText.append("\nSQLException: " + e.getMessage() + "\n");
        errorText.append("SQLState:    " + e.getSQLState() + "\n");
        errorText.append("VendorError: " + e.getErrorCode() + "\n");
    }

    public static void main(String[] args)
```

```
{  
  
    DeleteComputers dels = new DeleteComputers();  
    dels.addWindowListener(new WindowAdapter(){  
        public void windowClosing(WindowEvent e)  
        {  
            System.exit(0);  
        }  
    });  
    dels.buildGUI();  
}  
}
```

GITHUB Link:

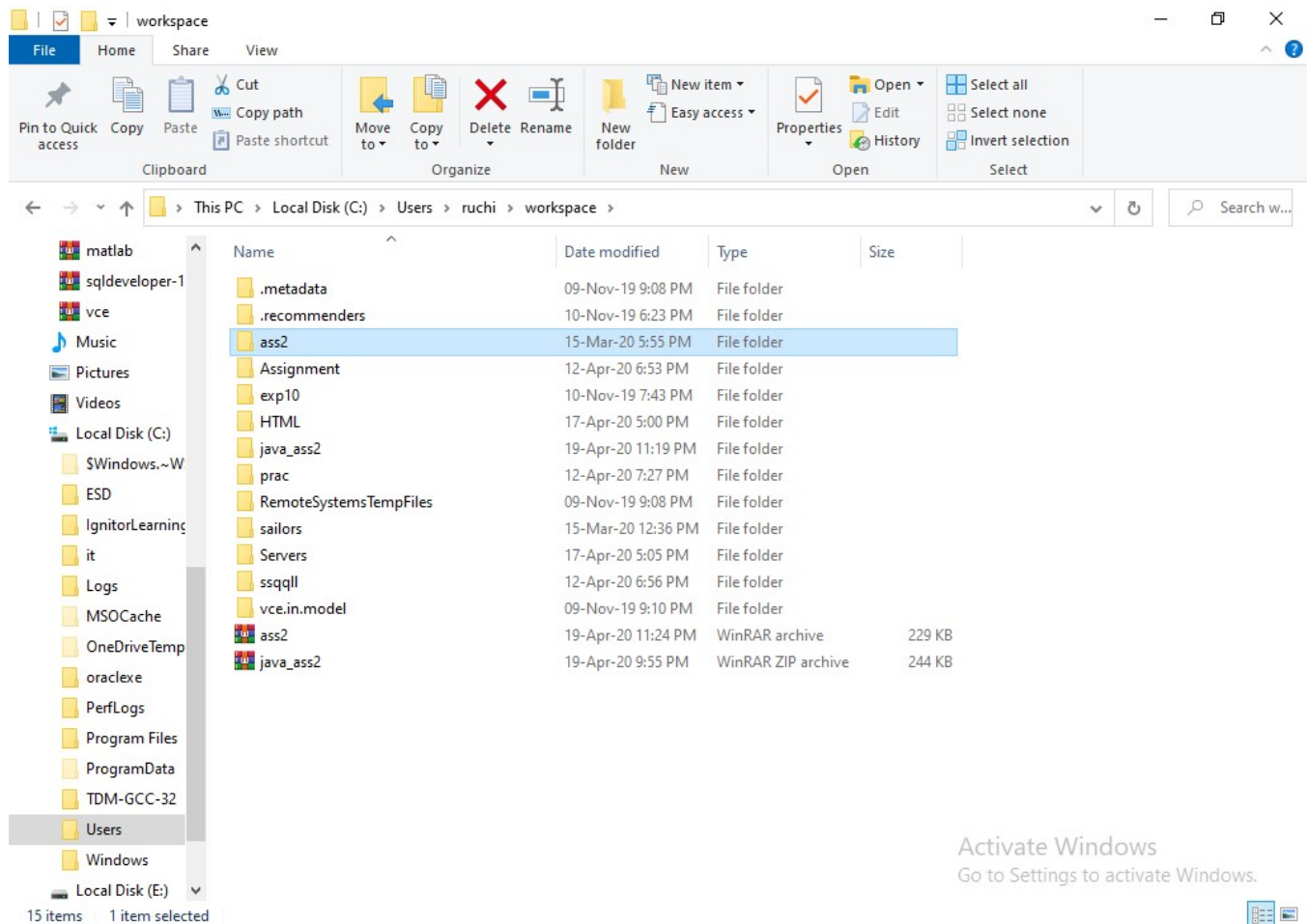
<https://github.com/ruchita0403-dot/VCE-Network-Connection-Management-System>



DBMS MINIPROJECT

TITLE:VCE Network Connection Management System

—————> project

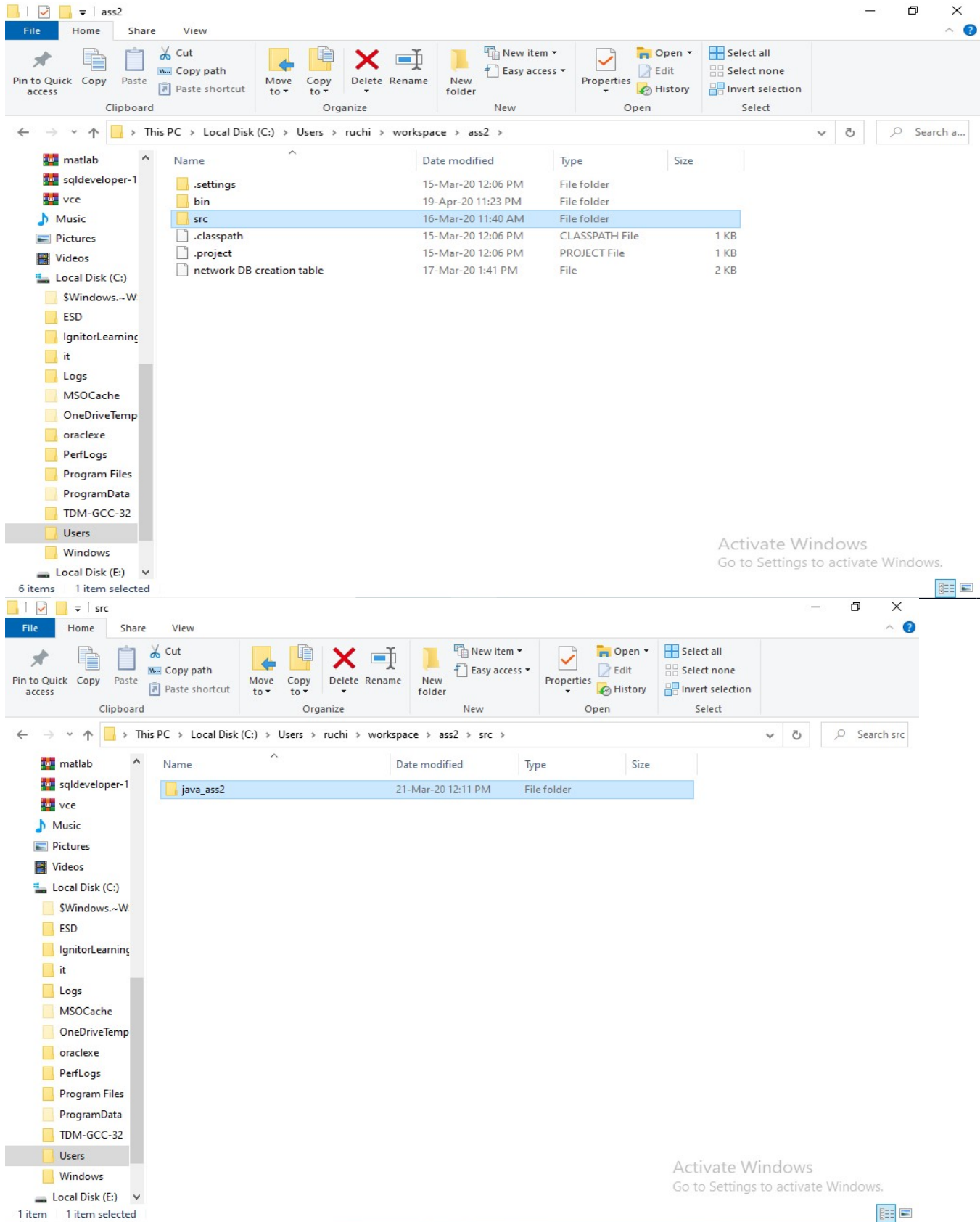


Roll no:1602-18-737-064

Name:A.Ruchita

DBMS MINIPROJECT

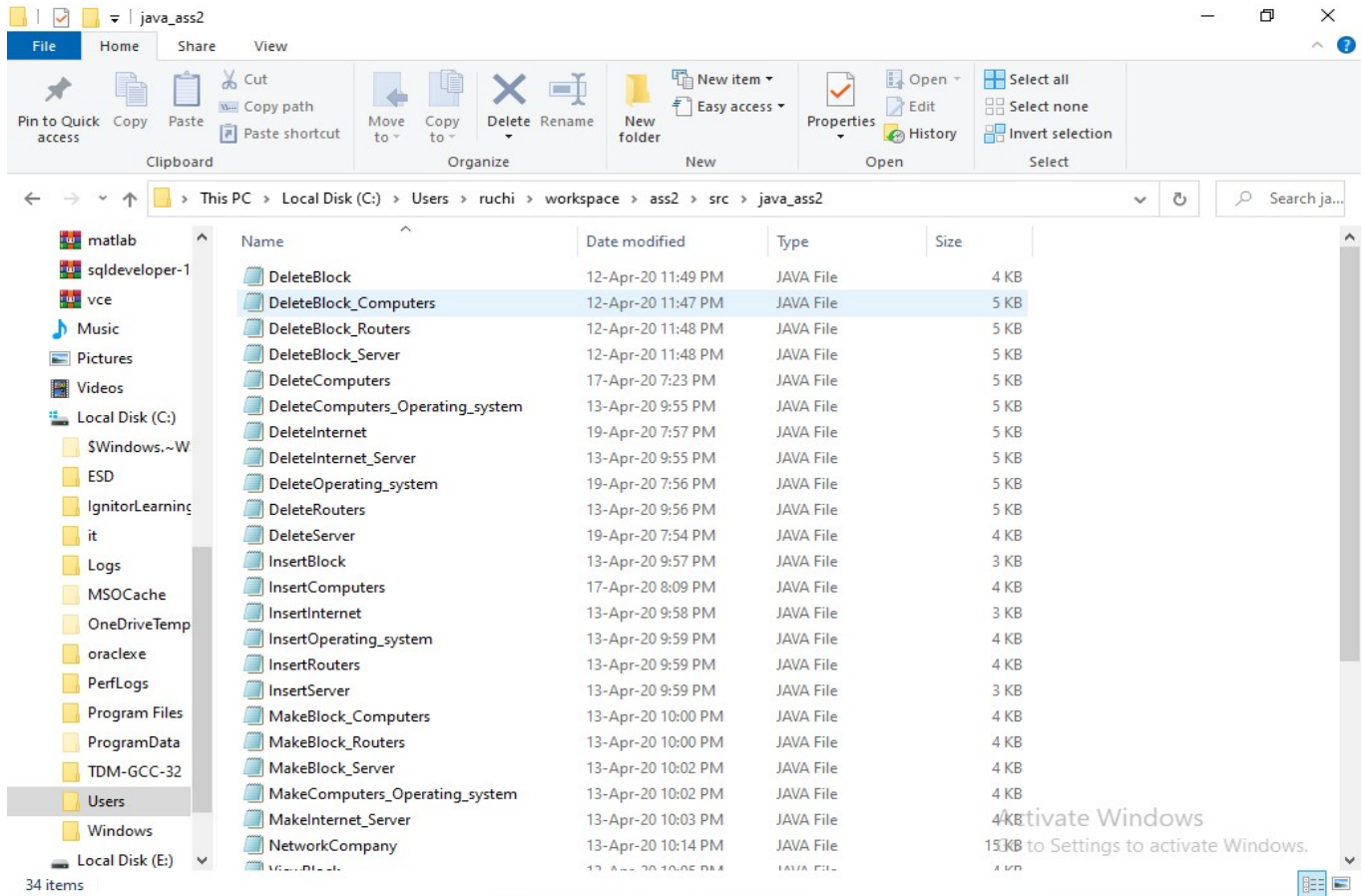
TITLE:VCE Network Connection Management System



Roll no:1602-18-737-064

Name:A.Ruchita

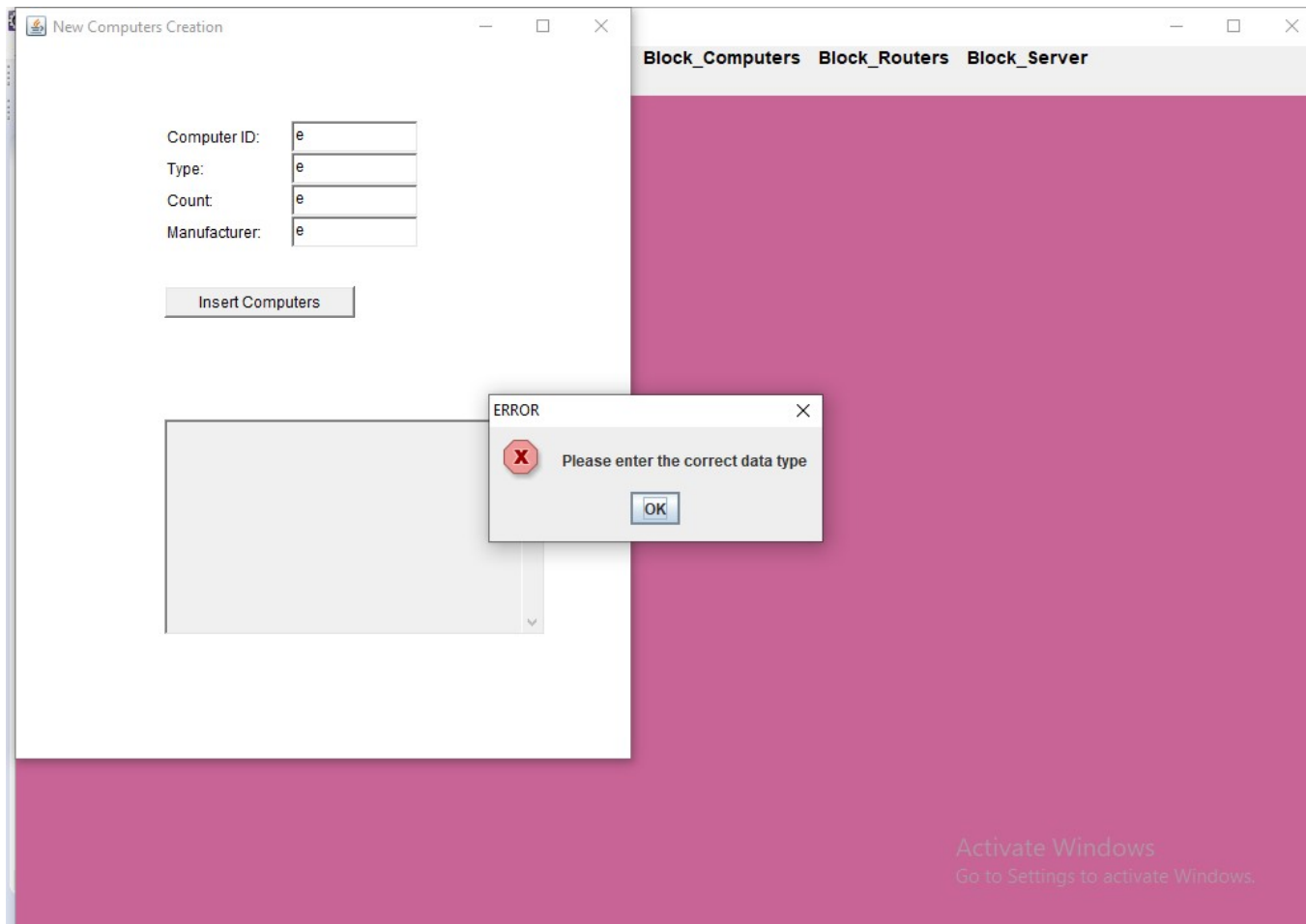
DBMS MINIPROJECT
TITLE:VCE Network Connection Management System



Roll no:1602-18-737-064

Name:A.Ruchita

TESTING



DML COMMANDS

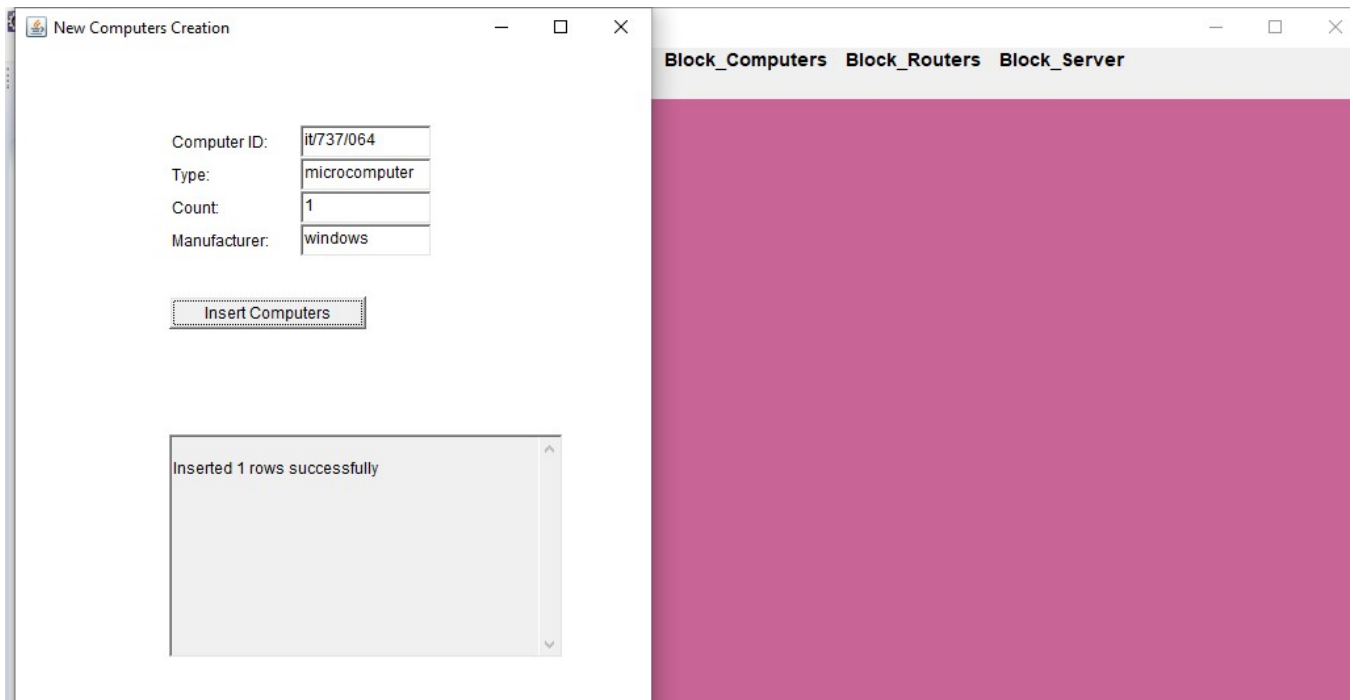
1. Insert into internet values('&html','&serv_provider','&mac_address');
2. Insert into computers values ('&type ' , ' & count ' , ' & manufacturer ' , ' & mode ' , ' &cid');
3. Insert into server values('&ipaddress');
4. Insert into block values('&bname','&hod','&branch');
5. Insert into server values('&ipaddress');
6. Insert into routers values('&website','&speed','&model','& username','&company');
7. Insert into operating_system values('&osname','& version','& vendor');
8. Insert into connected_to values('&mac','& ipadd');
9. Insert into provides_network_to('&bname','& ipaddress');
10. Insert into contains values ('&cid','&bname');
11. Insert into has values ('&cid','& osname');
12. Insert into are_having values ('&website','& bname');

OUTPUT SCREENSHOTS:

Java GUI Screenshot:

Computers

1) For inserting computers



2) for deleting computers

The screenshot shows two windows. The 'Remove computers' dialog box on the left has a list of computer IDs: abc/123-456, abc/123-789, abc/456-789, abc/456-111, and abc/345-998. It includes input fields for 'Computers ID:', 'Type:', 'Count:', and 'Manufacturer:', and a 'Delete Computers' button. A message box at the bottom states 'Deleted 1 rows successfully'. The 'Block_Computers' table window on the right is currently empty.

3) for updating computers

The screenshot shows two windows. The 'Update Computers' dialog box on the left has a list of computer IDs: abc/123-456, abc/123-789, abc/456-789, abc/456-111, abc/345-998, and it/737/064. It includes input fields for 'Computer ID:', 'Type:', 'Count:', and 'Manufacturer:', and an 'Update Computers' button. A message box at the bottom states 'Updated 1 rows successfully'. The 'Block_Computers' table window on the right is currently empty.

Block

1)insert block

New Block Creation

Block name: pinky

HOD: rosy

Branch: eee

Insert Block

Inserted 1 rows successfully

Block_Computers

Block_Routers

Block_Server

2) update block

Update Block

RAMANUJAN
CVRAMAN
JCBOSE
PENDAKANTI
VISHWESHWARAYYA
pinky

Block Name: pinky

Branch: mech

HOD : rosy

Updated 1 rows successfully

Block_Computers

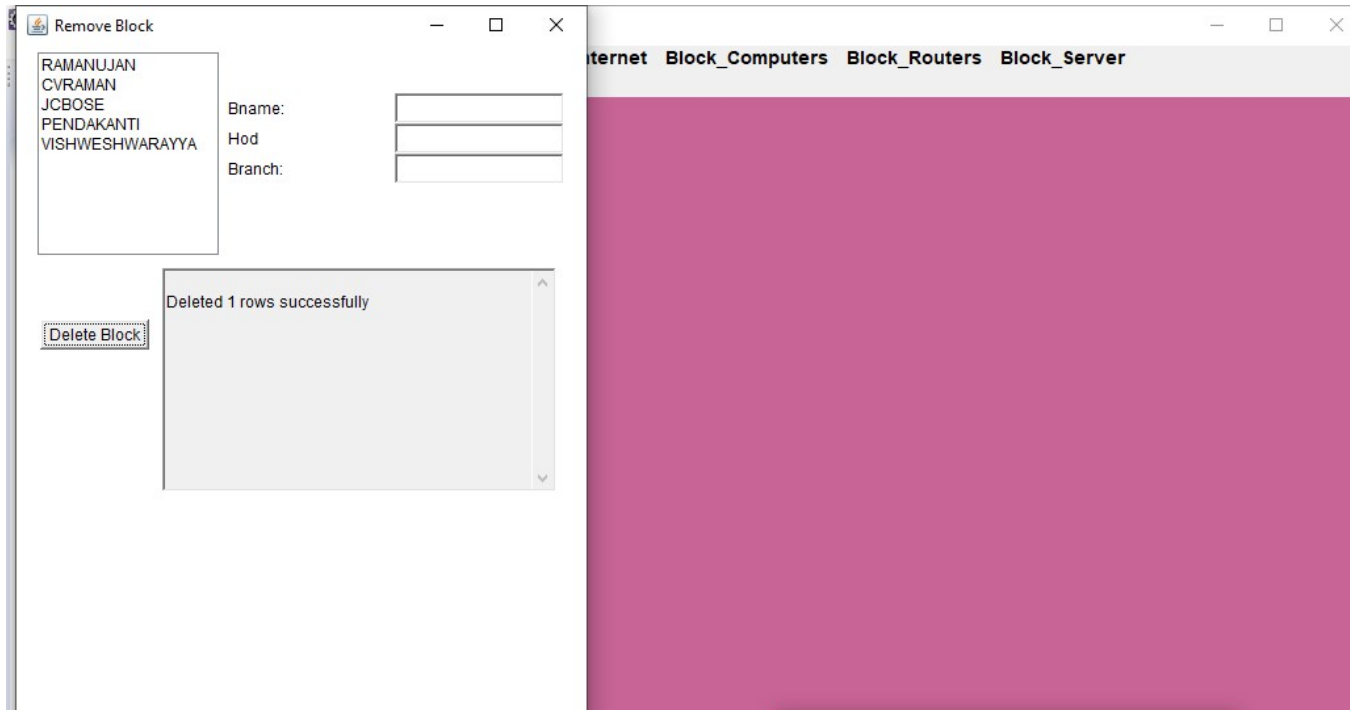
Block_Routers

Block_Server

3) delete block

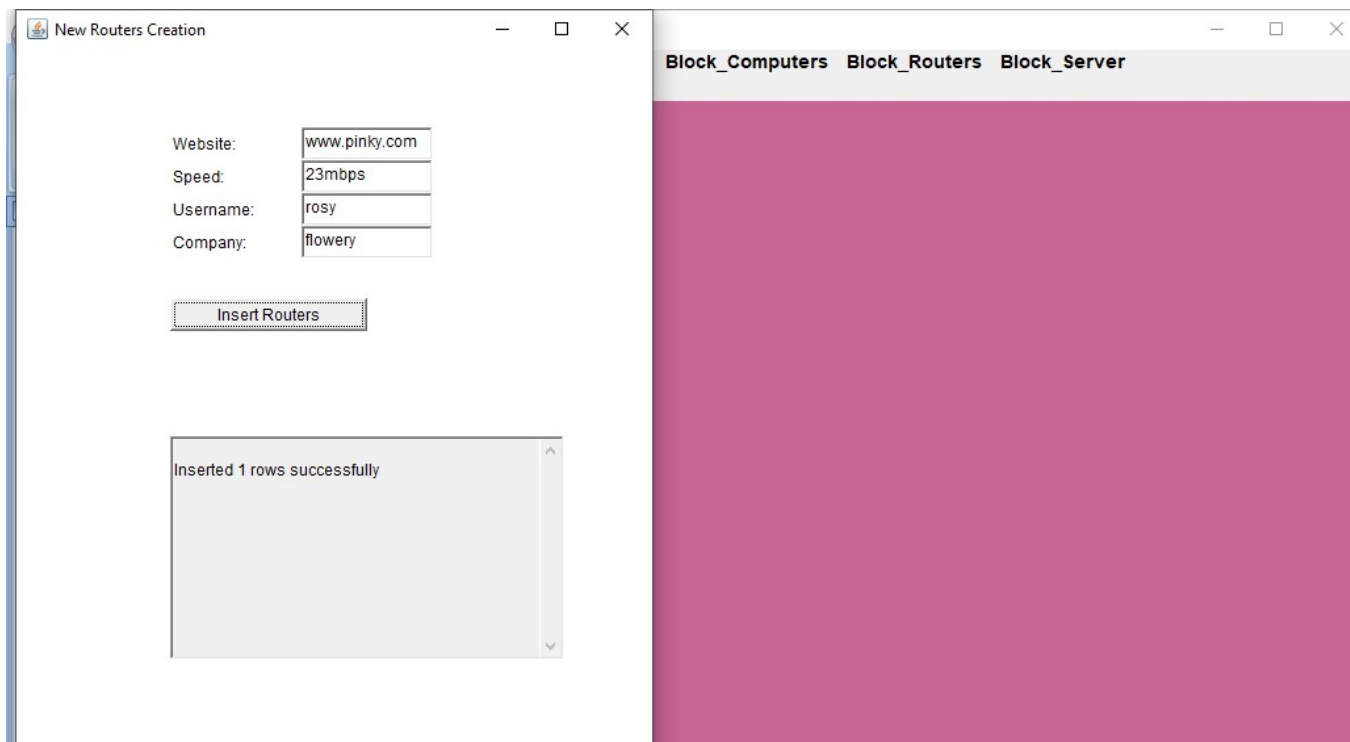
DBMS MINIPROJECT

TITLE:VCE Network Connection Management System



Routers

1)insert routers



2)update routers

The screenshot shows two windows from a network management application. On the left is the 'Update Routers' window, which contains a list of website URLs on the left, a form with fields for Website, Speed, Username, and Company on the right, and a confirmation message 'Updated 1 rows successfully' at the bottom. On the right is a sidebar menu with tabs labeled 'Block_Computers', 'Block_Routers', and 'Block_Server'. The 'Block_Routers' tab is currently selected, and the area below it is a solid pink color.

Update Routers

www.actcorp.in
www.hathway.in
www.bsnl.in
www.jio.in
www.xfinity.com
www

Website: www.pinky.com
Speed: 26mbps
Username: rosy
Company: flowery

Update Routers

Updated 1 rows successfully

Block_Computers **Block_Routers** **Block_Server**

3)delete routers

The screenshot shows two windows from the same network management application. On the left is the 'Remove Routers' window, which contains a list of website URLs on the left, a form with fields for Website, Speed, Username, and Company on the right, and a confirmation message 'Deleted 1 rows successfully' at the bottom. On the right is a sidebar menu with tabs labeled 'Internet', 'Block_Computers', 'Block_Routers', and 'Block_Server'. The 'Block_Routers' tab is currently selected, and the area below it is a solid pink color.

Remove Routers

www.actcorp.in
www.hathway.in
www.bsnl.in
www.jio.in
www.xfinity.com
www
www.abc.com

Website:
Speed:
Username:
Company:

Delete Routers

Deleted 1 rows successfully

Internet **Block_Computers** **Block_Routers** **Block_Server**

Servers

1)insert server

New Server Creation

IpaddressText:

123.321.4.7

Insert Server

Inserted 1 rows successfully

Block_Computers

Block_Routers

Block_Server

2)update server

Update Server

123.321.4.5
123.466.789
135.123.6.45
135.133.7.22
135.133.7.43
135.135.4.22

Ipaddress:

123.321.4.5

Update Server

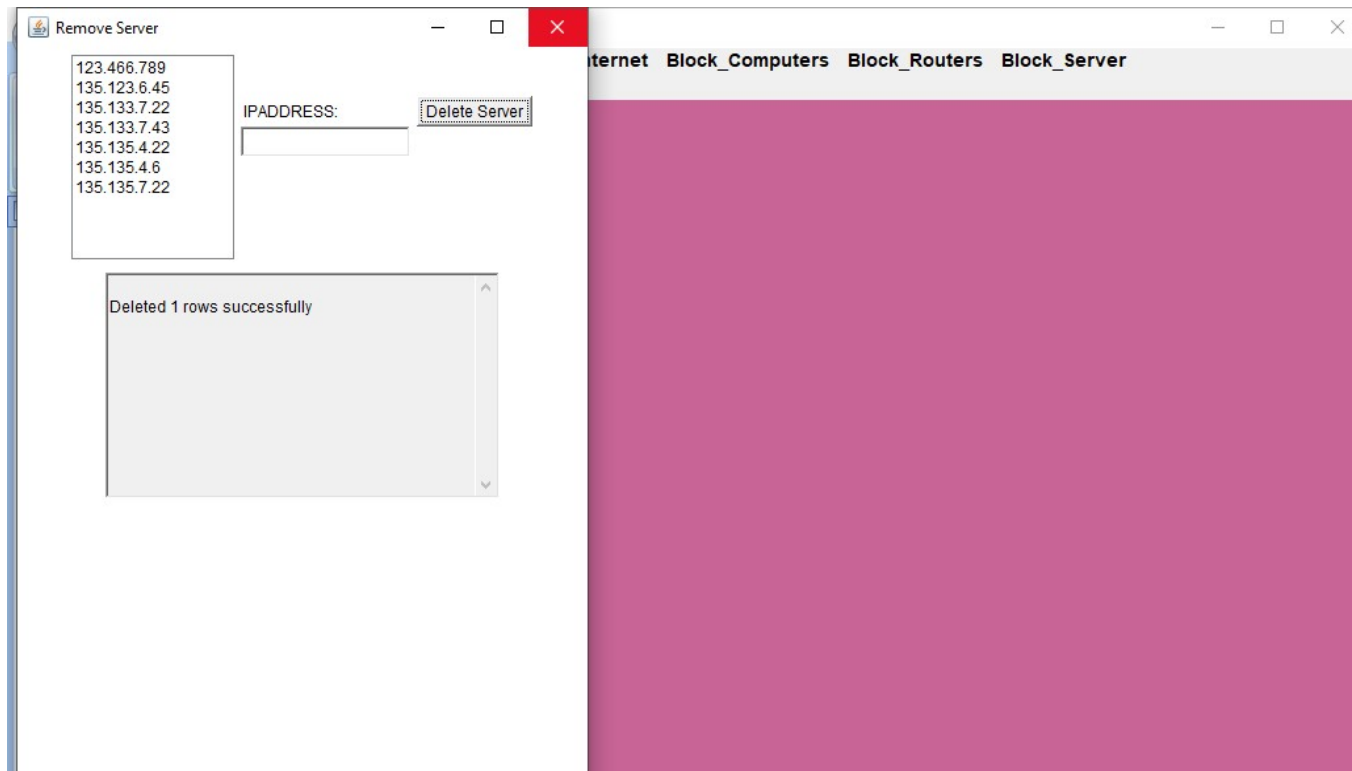
Updated 1 rows successfully

Block_Computers

Block_Routers

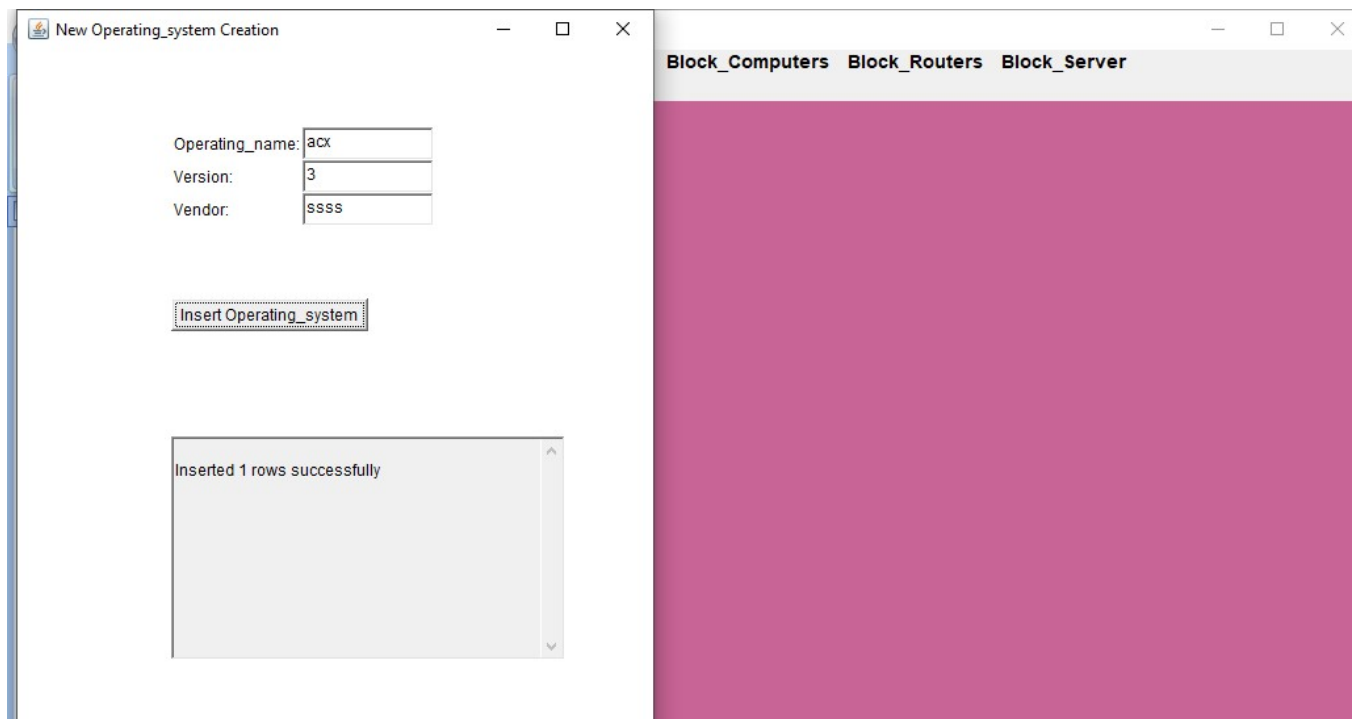
Block_Server

3)delete server



Operating System

1)insert operating system



2)update operating system

The screenshot shows two windows from a web application. On the left is a window titled 'Update Operating_system'. It features a list box on the left containing 'windows', 'linux', 'unix', 'ubuntu', 'andriod', and 'acx'. To the right of this list are three input fields: 'Operating_system_name:' with the value 'acx', 'Version:' with the value '4', and 'Vendor:' with the value 'SSSS'. Below these fields is a button labeled 'Update Operating_system'. At the bottom of the window, a message box states 'Updated 1 rows successfully'. On the right is a sidebar menu with a header bar containing 'Block_Computers', 'Block_Routers', and 'Block_Server'. The main content area of the sidebar is a solid pink color.

3)delete operating system

The screenshot shows two windows from a web application. On the left is a window titled 'Remove Operating_system'. It features a list box on the left containing 'windows', 'linux', 'unix', 'ubuntu', and 'andriod'. To the right of this list are three input fields: 'Operating sys name:', 'Version:', and 'Vendor:'. Below these fields is a button labeled 'Delete Operating_system'. At the bottom of the window, a message box states 'Deleted 1 rows successfully'. On the right is a sidebar menu with a header bar containing 'Internet', 'Block_Computers', 'Block_Routers', and 'Block_Server'. The main content area of the sidebar is a solid pink color.

Internet

1)insert internet

New Internet Creation

Mac_address:

-A4-5R-78-99-87

Html:

www.abc.com

Serv_provider:

aassdd

Insert Internet

Inserted 1 rows successfully

Block_ComputersBlock_RoutersBlock_Server

2)update internet

Update Internet

14-CC-20-2-B8-E5
565656
12-A4-5R-78-99-87
13-09-97-Y6-55-98
45-09-0G-22-45-89
46-09-I9-25-35-49
42-49-I6-65-H5-22

MAC:

12-A4-5R-78-99-87

Html:

www.abc.com

Serv_provider:

aasuu

Update Internet

Updated 1 rows successfully

Block_ComputersBlock_RoutersBlock_Server

3) delete internet

The screenshot shows two windows from a network management application. On the left is a 'Remove Internet' dialog box. It contains a list of five IP addresses: 14-CC-20-2-B8-E5, 565656, 13-09-97-Y6-55-98, 45-09-0G-22-45-89, and 46-09-I9-25-35-49. To the right of this list are three input fields labeled 'Mac_Address:', 'Html:', and 'Service_provider:'. Below the list is a 'Delete Internet' button. A message box at the bottom of the dialog says 'Deleted 1 rows successfully'. On the right is the main application window, which has a tabbed interface with 'Internet', 'Block_Computers', 'Block_Routers', and 'Block_Server'. The 'Internet' tab is currently selected, and the area below the tabs is a solid magenta color.

Block_computers

1)insert Block_computers

The screenshot shows two windows from the same network management application. On the left is a 'Make Block_ComputersButton' dialog box. It has two dropdown menus: 'Computer ID:' with the value 'abc/123-789' and 'Block Name:' with the value 'JCBOSE'. To the right of the 'Block Name' dropdown is a button labeled 'Block_Computers'. Below these fields is a message box that says 'Inserted 1 rows successfully'. On the right is the main application window, which has the same tabbed interface as before. The 'Block_Computers' tab is now selected, and the area below the tabs is a solid magenta color.

2)update Block_computers

The screenshot shows two windows from a network management application. On the left is a dialog box titled 'Update Block_Computers'. It features a list box containing the names: RAMANUJAN, CVRAMAN, JCBOSE, VISHWESHWARAYYA, and PENDAKANTI. To the right of the list are two input fields: 'Computer ID:' with the value 'abc/123-456' and 'Block Name:' with the value 'JCBOSE'. Below these fields is a button labeled 'Update Block_Computers'. At the bottom of the dialog is a message box that says 'Updated 1 rows successfully'. On the right is the main application window, which has a tabbed interface with three tabs: 'Block_Computers', 'Block_Routers', and 'Block_Server'. The 'Block_Computers' tab is currently selected, and the area below the tabs is a solid magenta color.

3)delete Block_computers

The screenshot shows two windows from the same network management application. On the left is a dialog box titled 'Remove Block_Computers'. It features a list box containing the following entries: 'abc/123-456 RAMANUJAN', 'abc/123-789 CVRAMAN', 'abc/345-998 VISHWESHWARAYYA', and 'abc/456-111 PENDAKANTI'. To the right of the list are two empty input fields: 'Computer ID:' and 'Block Name:'. Below these fields is a button labeled 'Delete Block_Computers'. At the bottom of the dialog is a message box that says 'Deleted 1 rows successfully'. On the right is the main application window, which has a tabbed interface with four tabs: 'Internet', 'Block_Computers', 'Block_Routers', and 'Block_Server'. The 'Block_Computers' tab is currently selected, and the area below the tabs is a solid magenta color.

Block_routers

1)insert block_routers

The screenshot shows two windows from the VCE Network Connection Management System. The left window, titled 'Make Block_Routers', contains a form with 'Router website:' set to 'www.actcorp.in' and 'Block Name:' set to 'CVRAMAN'. A 'Block_Routers' button is visible. Below the form, a message box states 'Inserted 1 rows successfully'. The right window shows the 'Block_Routers' tab selected in the application's main interface, which is currently empty.

2)update block_routers

The screenshot shows two windows from the VCE Network Connection Management System. The left window, titled 'Update Block_Routers', features a list of names: RAMANUJAN, CVRAMAN, JCBOSE, PENDAKANTI, VISHWESHWARAYYA, and CVRAMAN. To the right, 'Block Name:' is set to 'CVRAMAN' and 'Website:' is set to 'www.actcorp.in'. An 'Update Block_Routers' button is present. A message box below the list states 'Updated 2 rows successfully'. The right window shows the 'Block_Routers' tab selected in the application's main interface, which is currently empty.

3)delete block_routers

The screenshot displays two windows from the VCE Network Connection Management System. The left window, titled 'Update Block_Routers', contains a list of names (RAMANUJAN, CVRAMAN, JCBOSE, PENDAKANTI, VISHWESHWARAYYA, CVRAMAN) and a 'Block Name' field with the value 'CVRAMAN'. Below these fields is a message box stating 'Updated 2 rows successfully'. The right window shows a table with three columns: 'Block_Computers', 'Block_Routers', and 'Block_Server'. The 'Block_Server' column is highlighted in pink.

Block_server

1)insert block_server

The screenshot displays two windows from the VCE Network Connection Management System. The left window, titled 'Make Block_Server', contains a 'Block Name' field with the value 'RAMANUJAN' and an 'IPAddress' field with the value '123.466.789'. Below these fields is a message box stating 'Inserted 1 rows successfully'. The right window shows a table with three columns: 'Block_Computers', 'Block_Routers', and 'Block_Server'. The 'Block_Server' column is highlighted in pink.

2)update block_server

The screenshot shows two windows from the VCE Network Connection Management System. On the left is a dialog box titled 'Update Block_Server'. It contains a list of names: RAMANUJAN, VISHWESHWARAYYA, CVRAMAN, JCBOSE, and RAMANUJAN. To the right of the list are input fields for 'Block Name:' (containing 'Update Block_Server'), 'IPaddress:' (containing '135.133.7.22'), and a 'CVRAMAN' field. Below these fields is a message box that says 'Updated 1 rows successfully'. On the right is the main application window, which has tabs for 'Block_Computers', 'Block_Routers', and 'Block_Server'. The 'Block_Server' tab is selected, showing a large pink area.

3)delete block_server

The screenshot shows two windows from the VCE Network Connection Management System. On the left is a dialog box titled 'Remove Block_Server'. It contains a list of names and IP addresses: RAMANUJAN 135.135.4.6, VISHWESHWARAYYA 123.466.789, CVRAMAN 135.133.7.22, JCBOSE 135.135.4.22, null null, and null null. To the right of the list are input fields for 'Block Name' and 'IPAddress'. Below these fields is a message box that says 'Deleted 1 rows successfully'. On the right is the main application window, which has tabs for 'Internet', 'Block_Computers', 'Block_Routers', and 'Block_Server'. The 'Block_Server' tab is selected, showing a large pink area.

Computer_Operating_system

1)insert computer_operating_system

The screenshot shows two windows. The left window, titled 'Make Computers_Operating_system', contains a form with 'Computer ID' set to 'abc/123-456' and 'Operating_system' set to 'linux'. A button labeled 'Computers_Operating_system' is visible. Below the form, a message box states 'Inserted 1 rows successfully'. The right window is a sidebar with tabs 'Block_Computers', 'Block_Routers', and 'Block_Server', with 'Block_Computers' selected. The main area of the right window is a solid pink color.

2)update computer_operating_system

The screenshot shows two windows. The left window, titled 'Update Boat', contains a list of computer IDs: 'abc/123-456', 'abc/123-789', 'abc/456-789', 'abc/456-111', 'abc/345-998', and 'abc/123-456'. To the right of this list are input fields for 'Computer ID' (containing 'abc/456-789') and 'Operating_system_Name' (containing 'windows'). A button labeled 'Update Computers_Operating_system' is present. Below the form, a message box states 'Updated 1 rows successfully'. The right window is identical to the one in the previous screenshot, with the 'Block_Computers' tab selected and a pink main area.

3)delete computer_operating_system

The screenshot shows two windows. The left window, titled 'Remove Computers_Operating_system', contains a list box with two entries: 'abc/123-789' and 'abc/456-789'. To the right of the list box are two input fields labeled 'Computer ID:' and 'Operating Name:'. Below these fields is a button labeled 'Delete Computers_Operating_system'. At the bottom of the window is a message box that says 'Deleted 1 rows successfully'. The right window shows a table with the following columns: 'Internet', 'Block_Computers', 'Block_Routers', and 'Block_Server'. The 'Block_Computers' column is highlighted in pink.

Internet_server

1)insert internet_server

The screenshot shows two windows. The left window, titled 'Make Internet_Server', contains two input fields: 'Internet:' with a dropdown menu showing '13-09-97-Y6-55-98' and 'Server:' with a dropdown menu showing '135.123.6.45'. To the right of these fields is a button labeled 'Internet_Server'. Below these fields is a message box that says 'Inserted 1 rows successfully'. The right window shows a table with the following columns: 'Block_Computers', 'Block_Routers', and 'Block_Server'. The 'Block_Computers' column is highlighted in pink.

2)update internet_server

The screenshot shows a web application interface. On the left is a sidebar menu with four items: **Block_Computers**, **Block_Routers**, **Block_Server**, and **internet**. The **internet** item is highlighted. The main content area is divided into two panels. The left panel, titled 'Update Internet_Server', contains a list of MAC addresses: 14-CC-20-2-B8-E5, 13-09-97-Y6-55-98, 45-09-0G-22-45-89, 46-09-19-25-35-49, 42-49-16-65-H5-22, and 13-09-97-Y6-55-98. To the right of this list are input fields for 'MAC:' (containing '46-09-19-25-35-49') and 'Ipaddress:' (containing '135.133.7.43'). Below these fields is a button labeled 'Update Internet_Server'. At the bottom of the panel is a message box that says 'Updated 1 rows successfully'. The right panel is a large, empty area with a light blue background.

3)delete internet_server

The screenshot shows the same web application interface. The sidebar menu is identical, with **internet** highlighted. The main content area is divided into two panels. The left panel, titled 'Remove Internet_Server', contains a list of MAC addresses: 13-09-97-Y6-55-98, 45-09-0G-22-45-89, 46-09-19-25-35-49, 42-49-16-65-H5-22, and 13-09-97-Y6-55-98. To the right of this list are input fields for 'MAC:' and 'IPAddress:'. Below these fields is a button labeled 'Delete Internet_Server'. At the bottom of the panel is a message box that says 'Deleted 1 rows successfully'. The right panel is a large, empty area with a light blue background.

DISCUSSION & FUTURE WORK :

The application done till now is to store all the information related to the network connection of our college . Furthermore, other programming languages can also be used along with database by connecting SQL with it. This application can be extended further more to store network connections of other colleges, organizations etc

CONCLUSION:

Thus, a Java AWT based network connection management system is created which is connected to the Oracle 11g database. Therefore, all the entries in the form are directly updated on the network table created in the database.

REFERENCES :

<https://www.oracle.com/technetwork/java/javase/documentation/index.html>

<https://nptel.ac.in/courses/106105175/>

<https://google.github.io/styleguide/javaguide.html>

<https://nptel.ac.in/courses/106105191/>