

# **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.

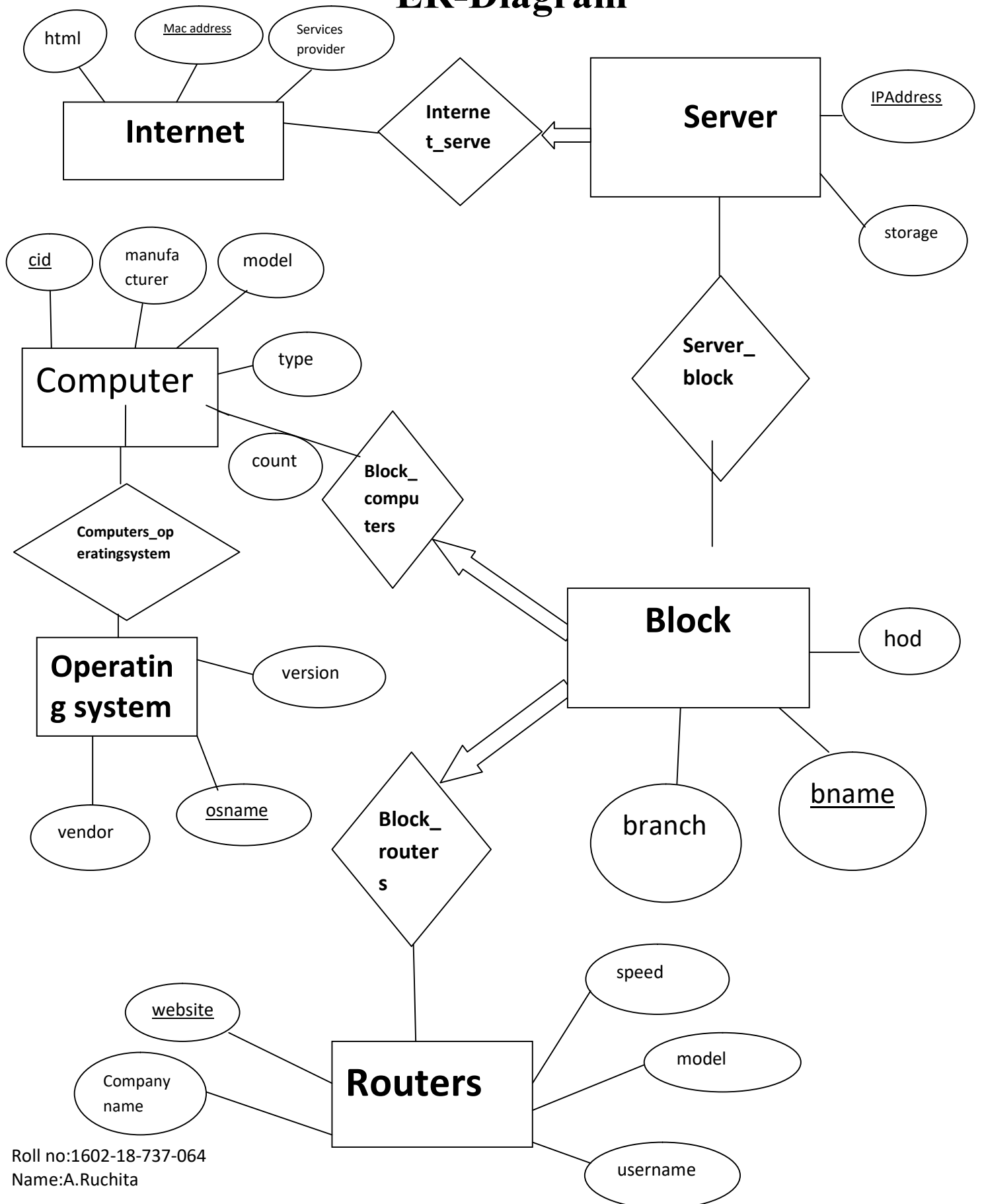
The java.awt package provides classes for AWT API such as TextField, Label, TextArea, RadioButton, CheckBox, Choice, List etc.

### **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                                         VARCHAR2(30)
SERV_PROVIDER                               VARCHAR2(20)
MAC_ADDRESS                                NOT NULL VARCHAR2(20)

SQL> desc server;
Name                                         Null?    Type
-----
IPADDRESS                                NOT NULL VARCHAR2(20)

SQL> desc block;
Name                                         Null?    Type
-----
BNAME                                     NOT NULL VARCHAR2(20)
HOD                                       VARCHAR2(20)
BRANCH                                   VARCHAR2(20)

SQL> desc routers
Name                                         Null?    Type
-----
SPEED                                     VARCHAR2(10)
MODEL                                    VARCHAR2(20)
USERNAME                                VARCHAR2(20)
COMPANY                                VARCHAR2(20)
WEBSITE                                NOT NULL VARCHAR2(50)

SQL> desc computers
Name                                         Null?    Type
-----
TYPE                                     VARCHAR2(20)
COUNT                                  NUMBER
MANUFACTURER                             VARCHAR2(20)
MODEL                                    VARCHAR2(20)
CID                                     NOT NULL VARCHAR2(20)
```

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

SQL> desc operating_system
Name                                         Null?    Type
-----
OSNAME                                   NOT NULL VARCHAR2(20)
VERSION                                VARCHAR2(20)
VENDOR                                VARCHAR2(20)

SQL> desc connected_to
Name                                         Null?    Type
-----
MAC                                     VARCHAR2(20)
IPADD                                  VARCHAR2(20)

SQL> desc provides_network_to
Name                                         Null?    Type
-----
BNAME                                   VARCHAR2(20)
IPADDRESS                              VARCHAR2(20)

SQL> desc contains
Name                                         Null?    Type
-----
CID                                    NOT NULL VARCHAR2(20)
BNAME                                   NOT NULL VARCHAR2(20)

SQL> desc has
Name                                         Null?    Type
-----
CID                                    VARCHAR2(20)
OSNAME                                VARCHAR2(20)

SQL> desc are_having
Name                                         Null?    Type
-----
WEBSITE                                VARCHAR2(50)
BNAME                                   VARCHAR2(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");
```

## 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);
        }
    }
}

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"));
                }
            }
            catch (SQLException e)
            {
                displaySQLErrors(e);
            }
        }
    });
}
```

```
        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);
        }
    }
}
```

## DBMS MINIPROJECT

TITLE:VCE Network Connection Management System

```
});  
  
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);
```

Roll no:1602-18-737-064

Name:A.Ruchita

## DBMS MINIPROJECT

TITLE:VCE Network Connection Management System

```
        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()
    {
```

## DBMS MINIPROJECT

TITLE:VCE Network Connection Management System

```
        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);
    }
}
```

Roll no:1602-18-737-064

Name:A.Ruchita

```
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)
            {

```

```
                displaySQLExceptions(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 deleteException)
            {
                displaySQLExceptions(deleteException);
            }
        }
    });
```



## DBMS MINIPROJECT

TITLE:VCE Network Connection Management System

```
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);
```

Roll no:1602-18-737-064

Name:A.Ruchita

```
        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();
    }
}
```

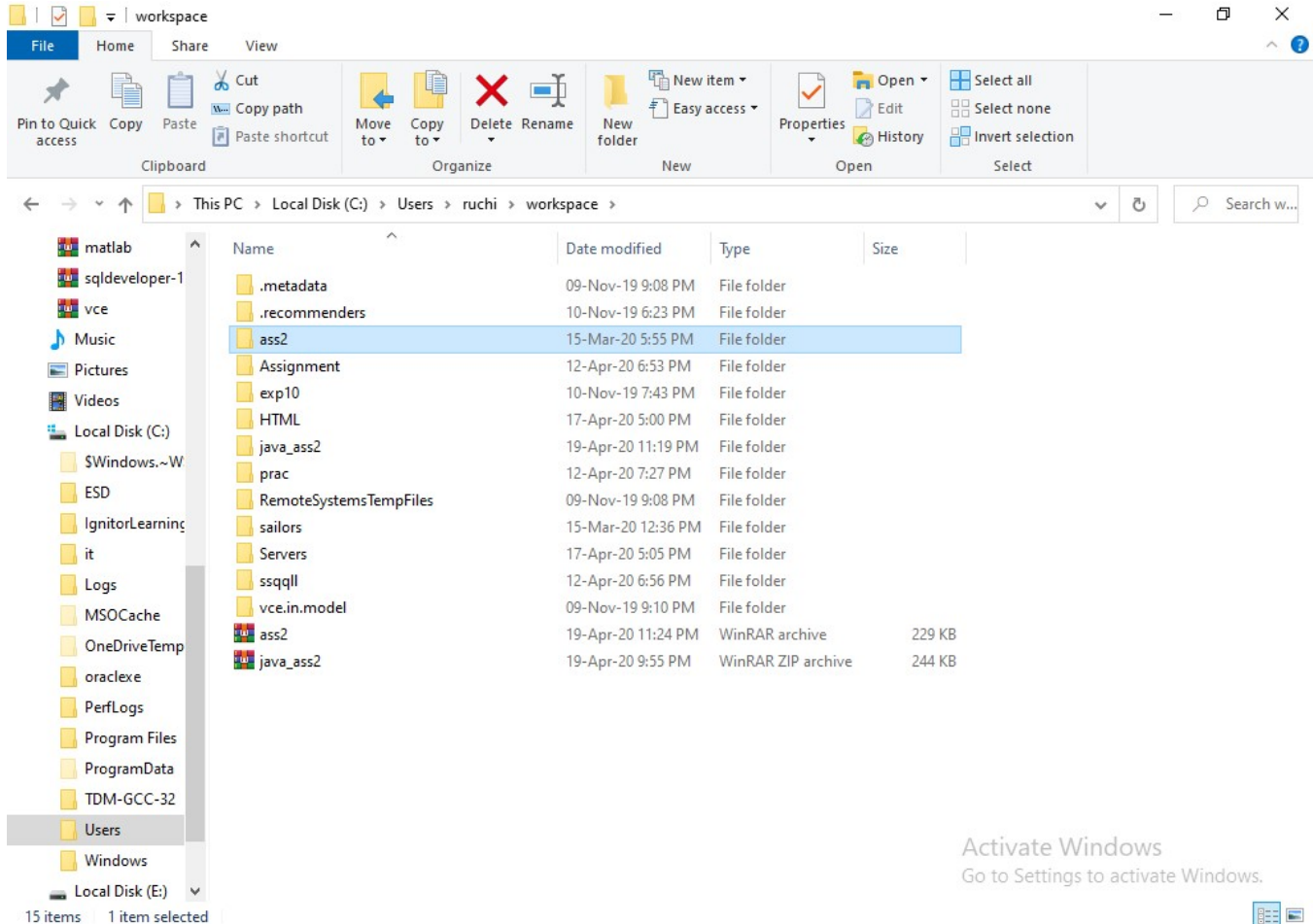
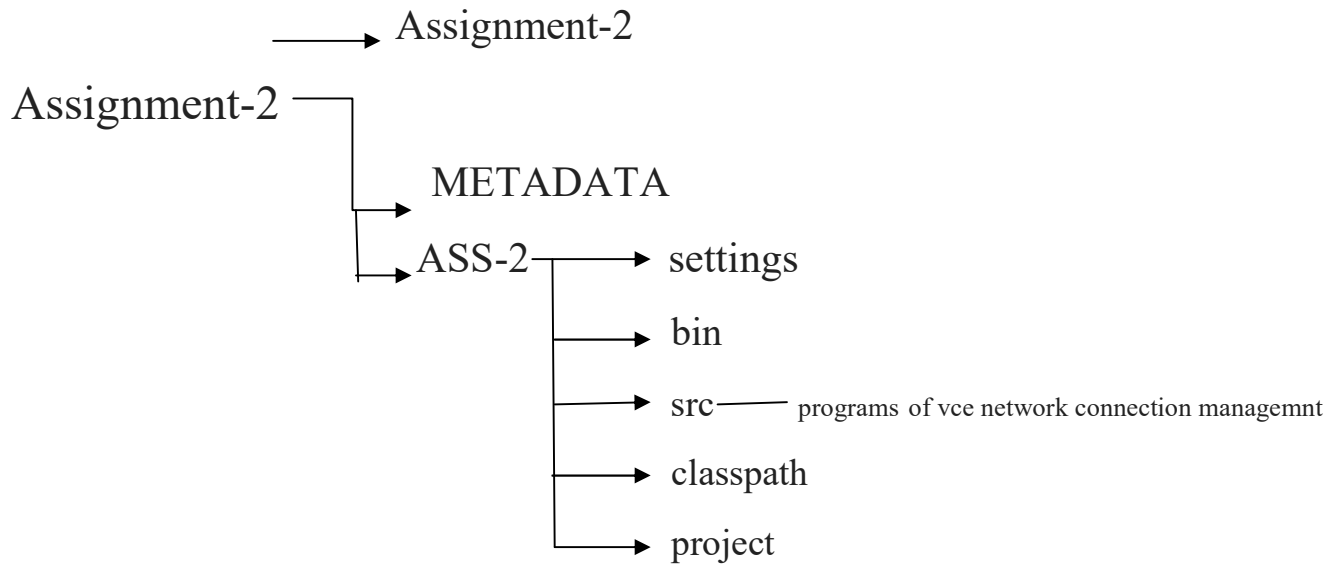
## GITHUB Link:

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

Github link

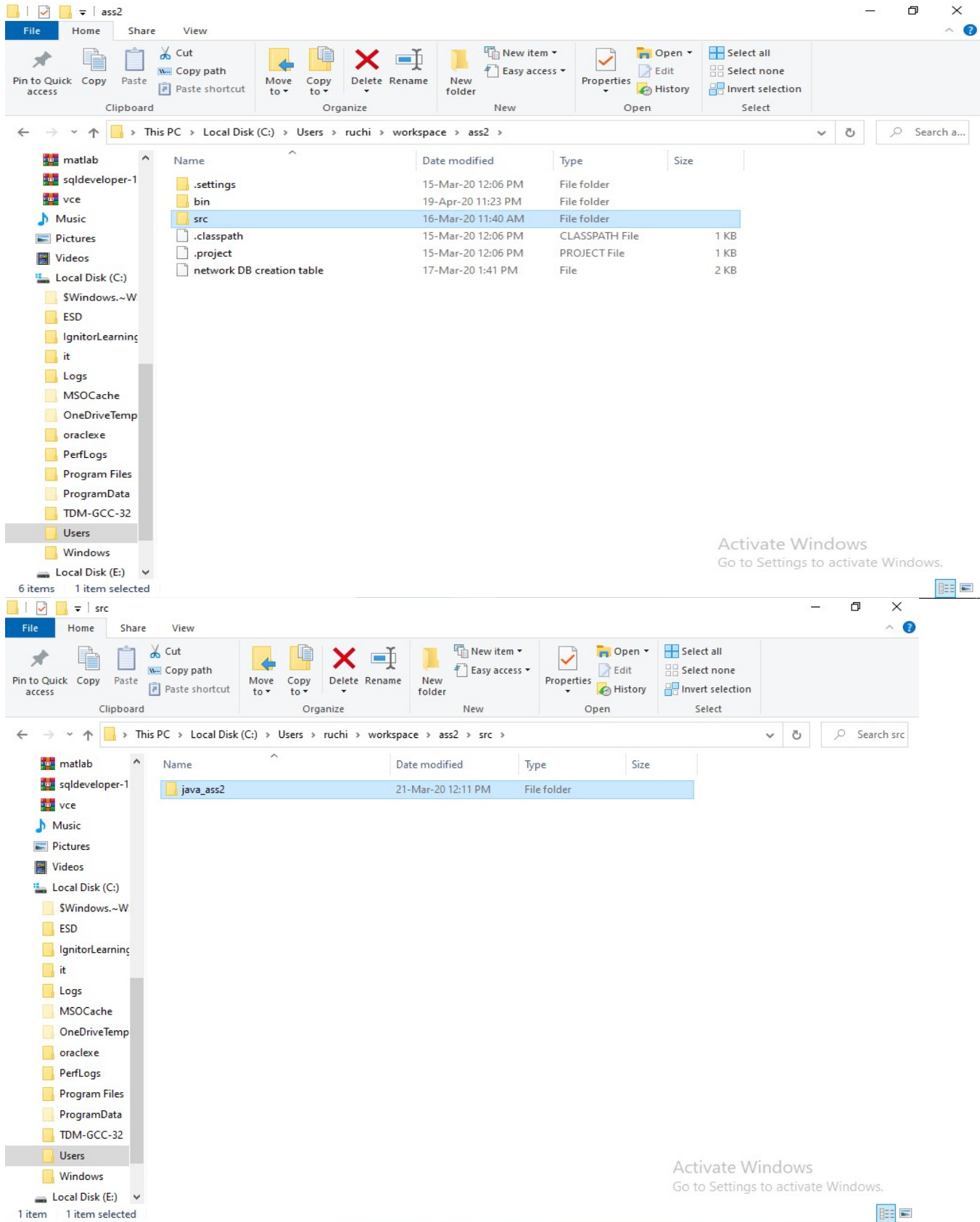
VCE Network Connection Management System.pdf

DbmsReport



## DBMS MINIPROJECT

### TITLE:VCE Network Connection Management System

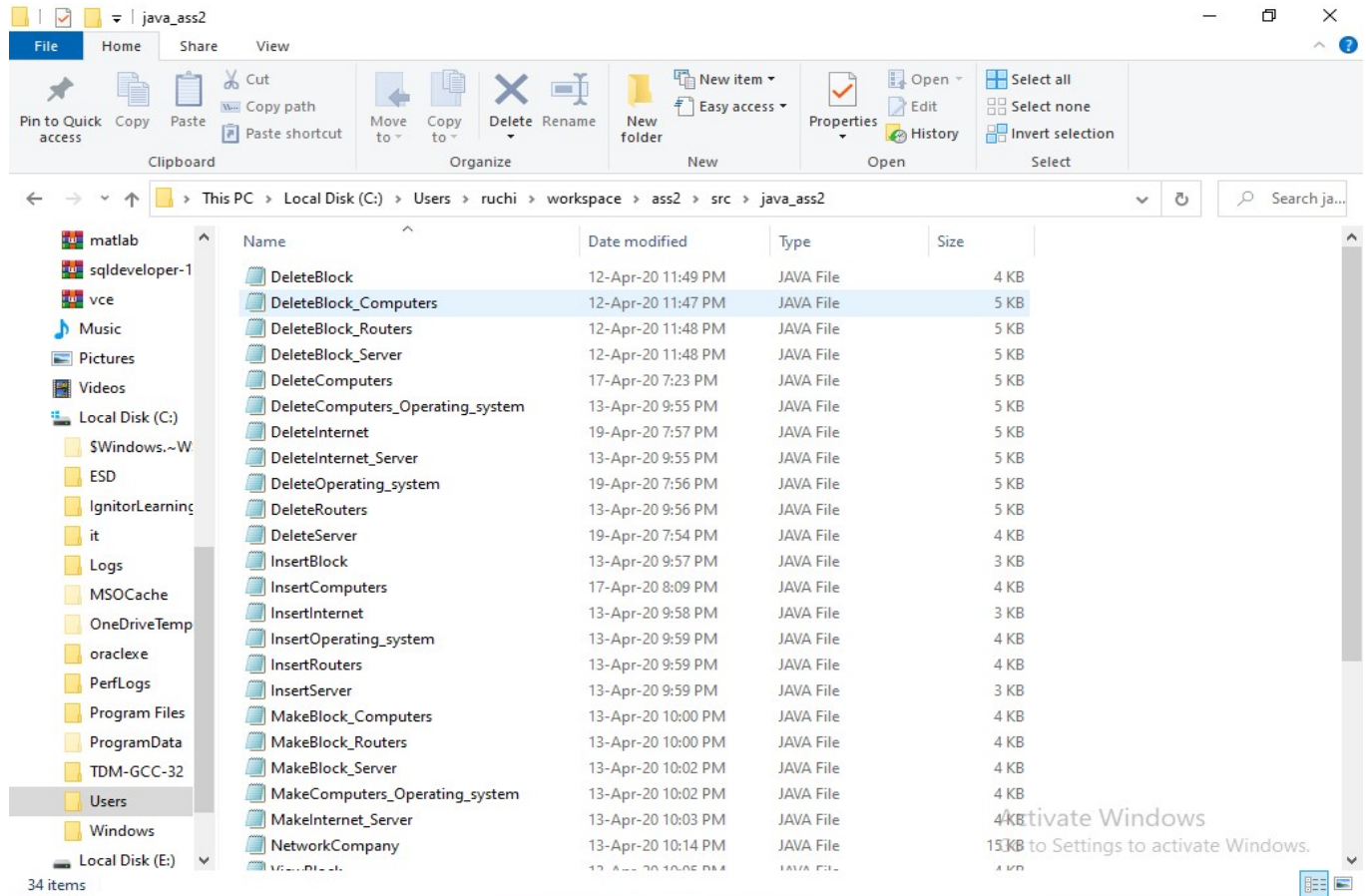


Roll no:1602-18-737-064

Name:A.Ruchita

# DBMS MINIPROJECT

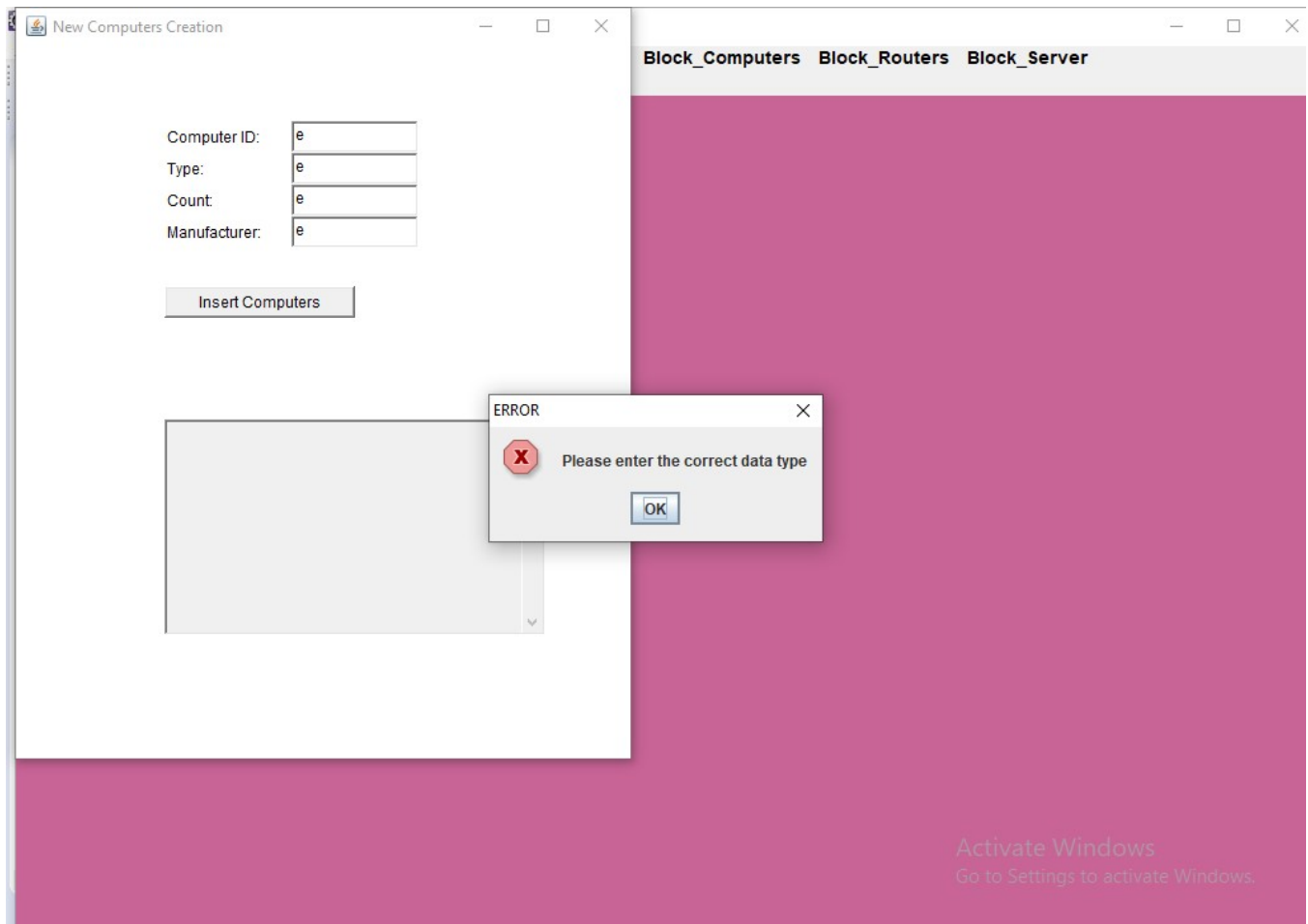
## TITLE:VCE Network Connection Management System



Roll no:1602-18-737-064

Name:A.Ruchita

## TESTING



## **DML COMMANDS**

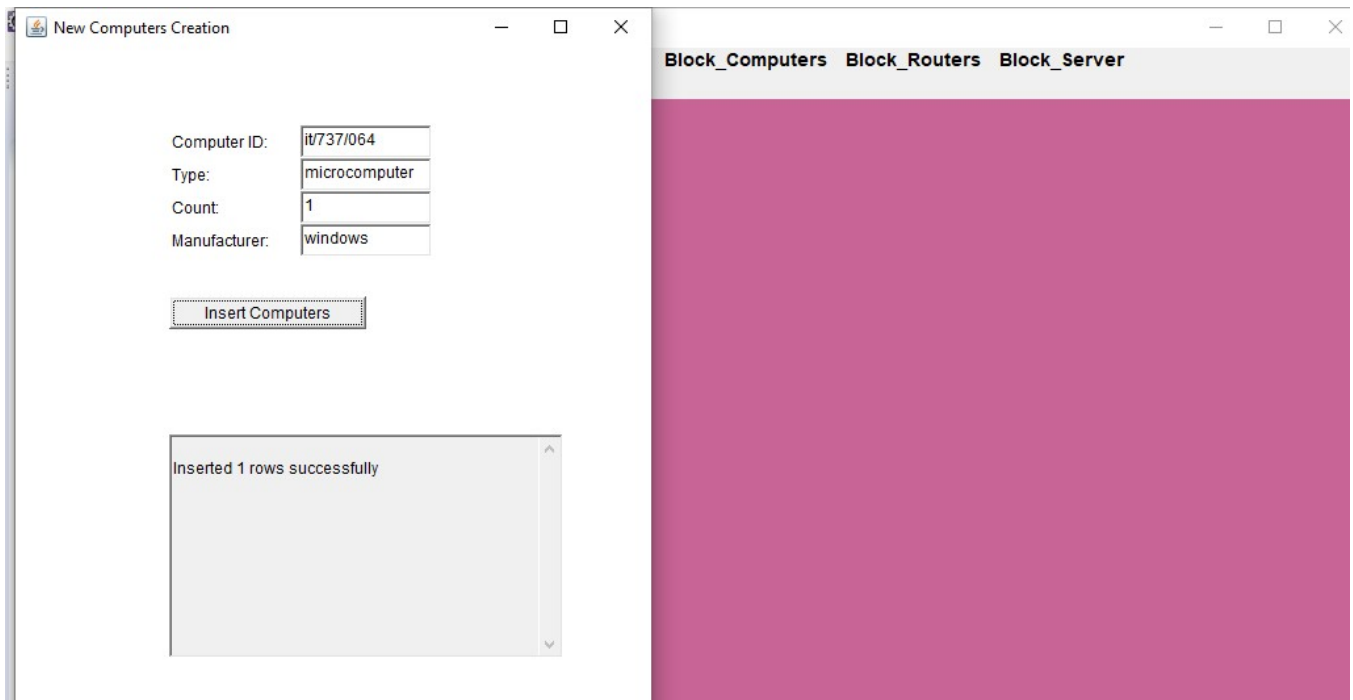
1. Insert into internet values('&html','&serv\_provider','&mac\_address');
2. Insert into computers values ( '&type ' , ' & count ' , ' & manufacturer ' , ' & mode ' , ' &cid');
3. Insert into server values('&ipaddress');
4. Insert into block values('&bname','&hod','&branch');
5. Insert into server values('&ipaddress');
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

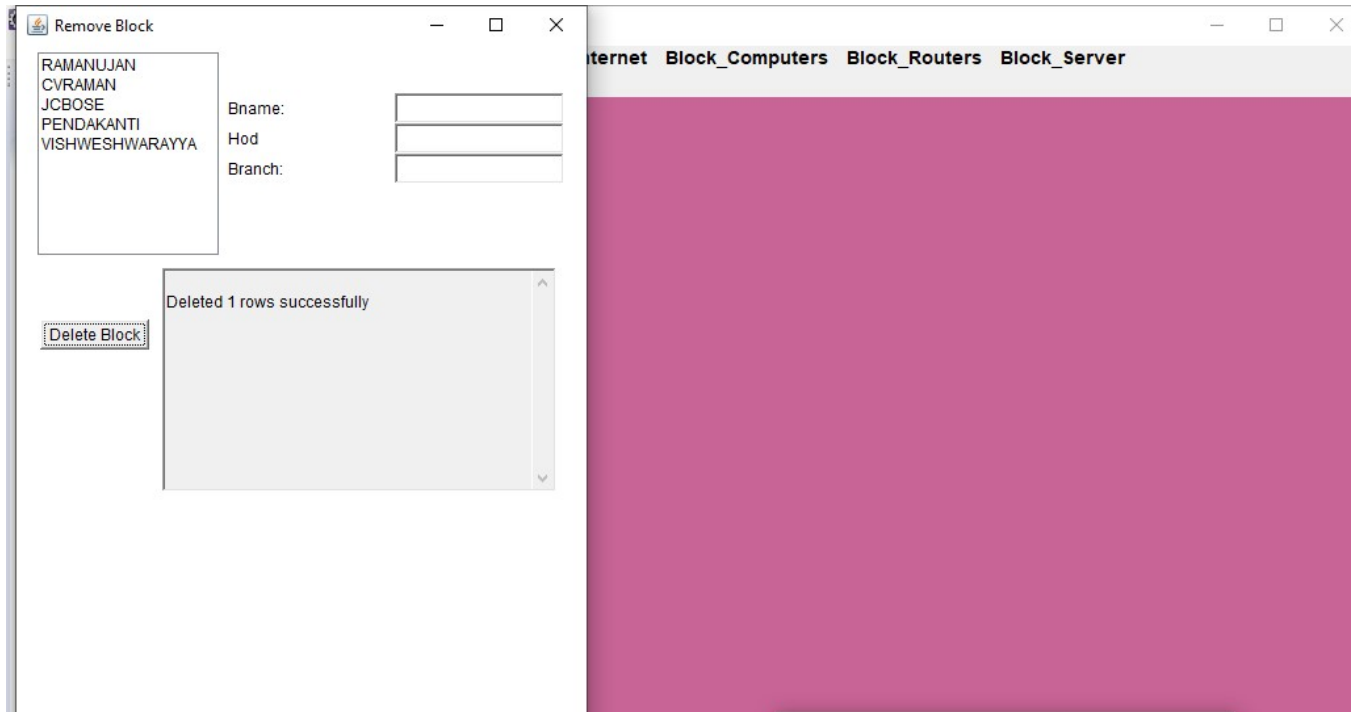
Updated 1 rows successfully

Block\_Computers

Block\_Routers

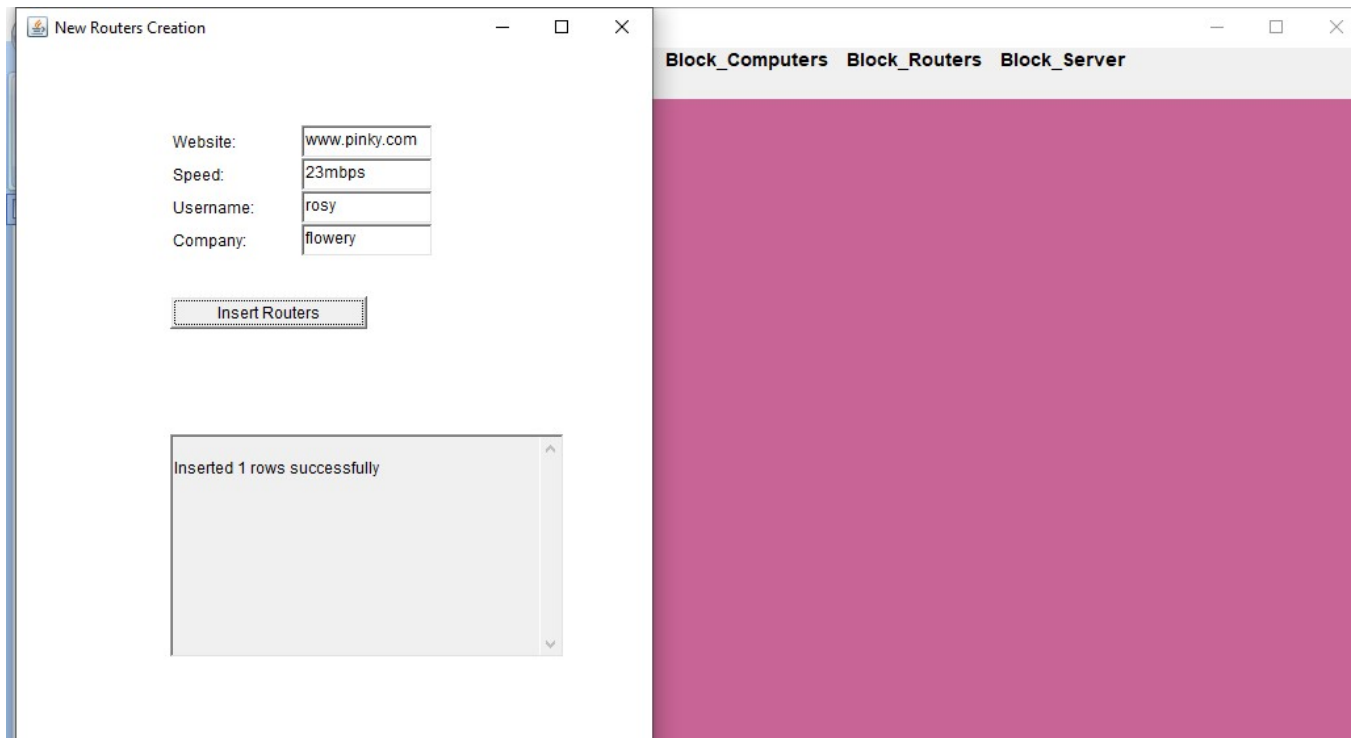
Block\_Server

### 3) delete block



## Routers

### 1)insert routers



## 2)update routers

Update Routers

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

Website:  
Speed:  
Username:  
Company:

www.pinky.com  
26mbps  
rosy  
flowery

Update Routers

Updated 1 rows successfully

Block\_Computers

Block\_Routers

Block\_Server

## 3)delete routers

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

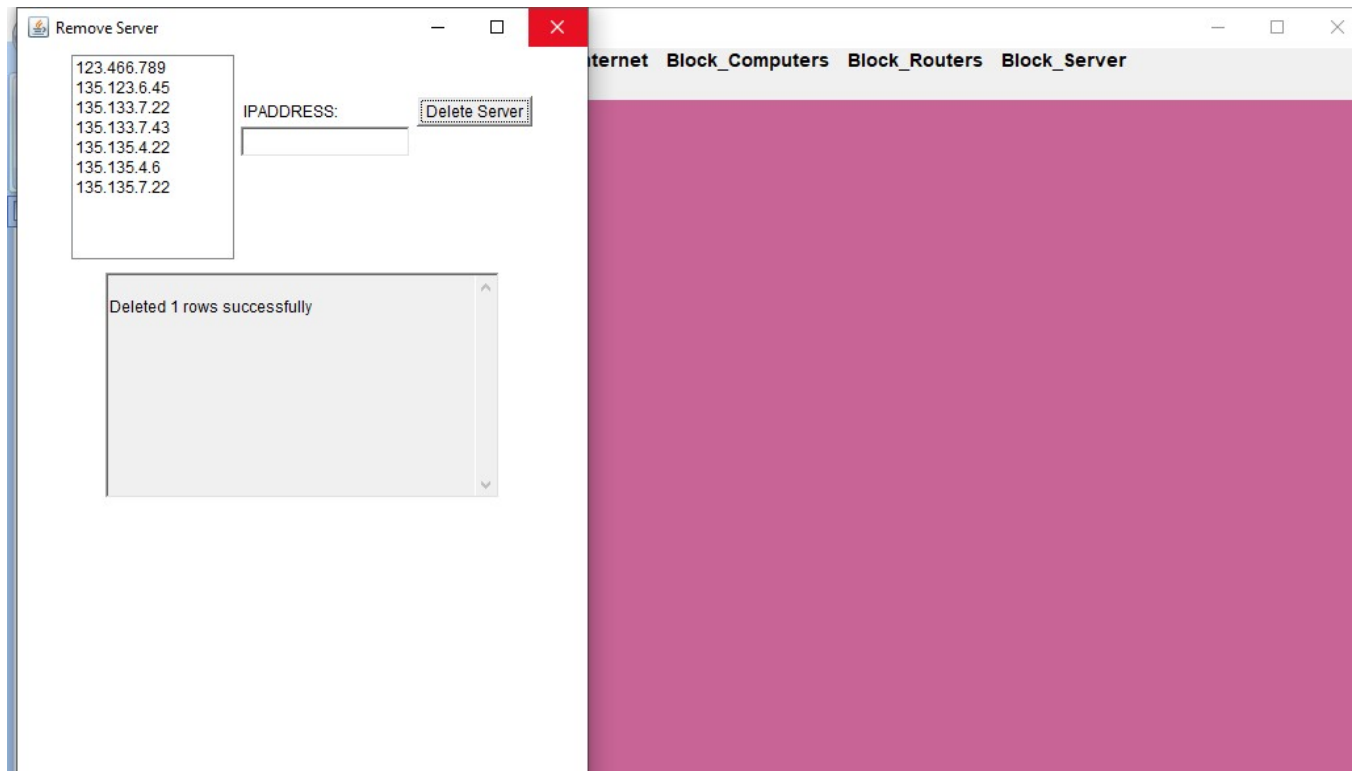
Updated 1 rows successfully

Block\_Computers

Block\_Routers

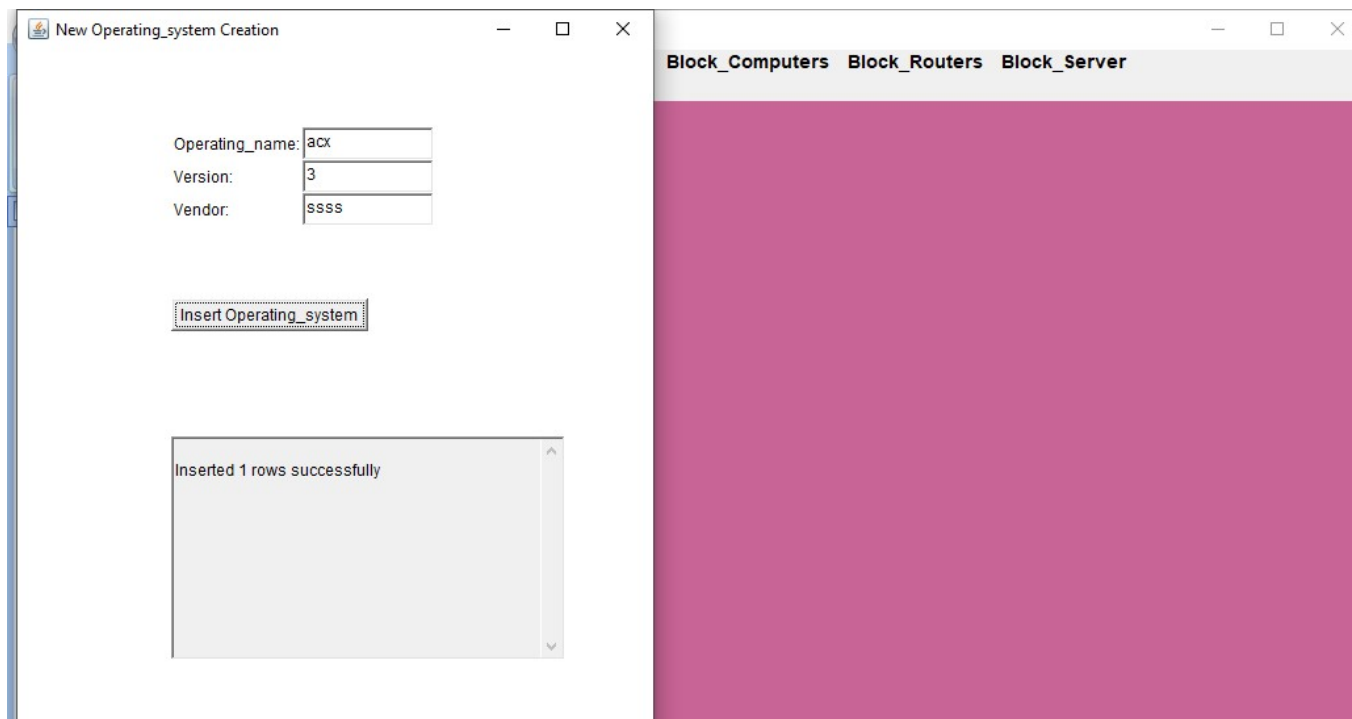
Block\_Server

### 3)delete server



## Operating System

### 1)insert operating system



## 2)update operating system

The screenshot shows two windows from a database application. The left window, titled 'Update Operating\_system', contains a list box with the following items: windows, linux, unix, ubuntu, andriod, and acx. To the right of the list box 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'. A message box at the bottom of the window states 'Updated 1 rows successfully'. The right window shows a table named 'Block\_Computers' with a pink background. The table has a header row with the following columns: Block\_Computers, Block\_Routers, and Block\_Server.

## 3)delete operating system

The screenshot shows two windows from a database application. The left window, titled 'Remove Operating\_system', contains a list box with the following items: windows, linux, unix, ubuntu, andriod, and acx. To the right of the list box are three input fields: 'Operating sys name:', 'Version:', and 'Vendor:'. Below these fields is a button labeled 'Delete Operating\_system'. A message box at the bottom of the window states 'Deleted 1 rows successfully'. The right window shows a table named 'Block\_Computers' with a pink background. The table has a header row with the following columns: Block\_Computers, Block\_Routers, and Block\_Server.

## 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 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

Update Block\_Computers

RAMANUJAN  
CVRAMAN  
JCBOSE  
VISHWESHWARAYYA  
PENDAKANTI

Computer ID:  
abc/123-456  
Block Name:  
JCBOSE

Update Block\_Computers

Updated 1 rows successfully

Block\_Computers Block\_Routers Block\_Server

## 3)delete Block\_computers

Remove Block\_Computers

abc/123-456 RAMANUJAN  
abc/123-789 CVRAMAN  
abc/345-998 VISHWESHWARAYYA  
abc/456-111 PENDAKANTI

Computer ID:  
Block Name:

Delete Block\_Computers

Deleted 1 rows successfully

Internet Block\_Computers Block\_Routers Block\_Server

## Block\_routers

### 1)insert block\_routers

The screenshot shows two windows from a database application. The left window, titled 'Make Block\_Routers', contains a form with two dropdown menus: 'Router website:' set to 'www.actcorp.in' and 'Block Name:' set to 'CVRAMAN'. A 'Block\_Routers' button is next to the second dropdown. Below the form is a text box displaying 'Inserted 1 rows successfully'. The right window is a sidebar menu with three tabs: 'Block\_Computers', 'Block\_Routers' (which is selected), and 'Block\_Server'. The selected tab shows a solid pink area, likely representing a data table.

### 2)update block\_routers

The screenshot shows two windows from a database application. The left window, titled 'Update Block\_Routers', contains a list of names on the left: RAMANUJAN, CVRAMAN, JCBOSE, PENDAKANTI, VISHWESHWARAYYA, and CVRAMAN. To the right of this list are three input fields: 'Block Name:' set to 'CVRAMAN', 'Website:' set to 'www.actcorp.in', and an 'Update Block\_Routers' button. Below the form is a text box displaying 'Updated 2 rows successfully'. The right window is a sidebar menu with three tabs: 'Block\_Computers', 'Block\_Routers' (which is selected), and 'Block\_Server'. The selected tab shows a solid pink area, likely representing a data table.

### 3)delete block\_routers

Update Block\_Routers

RAMANUJAN  
CVRAMAN  
JCBOSE  
PENDAKANTI  
VISHWESHWARAYYA  
CVRAMAN

Block Name:  
CVRAMAN

Website :  
www.actcorp.in

Update Block\_Routers

Updated 2 rows successfully

Block\_ComputersBlock\_RoutersBlock\_Server

## Block\_server

### 1)insert block\_server

Make Block\_Server

Block Name:  
RAMANUJAN

IPAddress:  
123.466.789

Block\_Server

Inserted 1 rows successfully

Block\_ComputersBlock\_RoutersBlock\_Server

## 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 a window titled "Make Computers\_Operating\_system". It contains two dropdown menus: "Computer ID:" with the value "abc/123-456" and "Operating\_system:" with the value "linux". A button labeled "Computers\_Operating\_system" is visible. Below the inputs, a message box states "Inserted 1 rows successfully". To the right, a larger window titled "Block\_Computers Block\_Routers Block\_Server" is partially visible, showing a solid pink area.

### 2)update computer\_operating\_system

The screenshot shows a window titled "Update Boat". It 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". A dropdown menu for "Computer ID:" is set to "abc/456-789". A text input field for "Operating\_system\_Name:" contains the value "windows". A button labeled "Update Computers\_Operating\_system" is visible. Below the inputs, a message box states "Updated 1 rows successfully". To the right, a larger window titled "Block\_Computers Block\_Routers Block\_Server" is partially visible, showing a solid pink area.

### 3)delete computer\_operating\_system

The screenshot shows two windows from the VCE Network Connection Management System. The left window, titled 'Remove Computers\_Operating\_system', contains a list box with two entries: 'abc/123-789' and 'abc/456-789'. Below the list box are two input fields labeled 'Computer ID:' and 'Operating Name:'. A button labeled 'Delete Computers\_Operating\_system' is positioned below the input fields. At the bottom of the window, a message box states 'Deleted 1 rows successfully'. The right window shows the 'Block\_Computers' tab, which is currently empty and has a pink background.

### Internet\_server

#### 1)insert internet\_server

The screenshot shows two windows from the VCE Network Connection Management System. 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'. A button labeled 'Internet\_Server' is positioned to the right of the 'Server:' field. Below the input fields, a message box states 'Inserted 1 rows successfully'. The right window shows the 'Block\_Computers' tab, which is currently empty and has a pink background.

## 2)update internet\_server

The screenshot shows two windows from a network management application. The main window is titled 'Update Internet\_Server' and contains a list of MAC addresses on the left: 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'). A button labeled 'Update Internet\_Server' is positioned to the right of the MAC input field. Below these fields is a text box that says 'Updated 1 rows successfully'. To the right of the main window is a sidebar menu with three options: 'Block\_Computers', 'Block\_Routers', and 'Block\_Server'. The 'Block\_Server' option is currently selected, and the area to its right is a solid magenta color.

## 3)delete internet\_server

The screenshot shows two windows from the same network management application. The main window is titled 'Remove Internet\_Server' and contains a list of MAC addresses on the left: 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:'. A button labeled 'Delete Internet\_Server' is positioned below the list. Below the button is a text box that says 'Deleted 1 rows successfully'. To the right of the main window is a sidebar menu with four options: 'internet', 'Block\_Computers', 'Block\_Routers', and 'Block\_Server'. The 'internet' option is currently selected, and the area to its right is a solid magenta color.



## **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/>