* **Name – Vinay Nitin Sarda**
* **Roll No. – 23**
* **Class – S.Y.B.tech(B)**
* **Batch – B2**

**Exp -8**

**Problem Statement –**

**Develop a Swing GUI based standard calculator program**

**Program –**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class calculator implements ActionListener{

JTextField t1,t2,t3;

JLabel firstNum,secondNum,resultOfNum;

JButton addButton,subButton,multiButton,divButton;

JFrame jf;

static double a=0,b=0,result=0;

calculator()

{

jf = new JFrame("Calculator");

jf.setSize(400,500);

jf.setVisible(true);

jf.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

jf.setFocusable(true);

jf.setLayout(new GridLayout(4,2));

firstNum=new JLabel("First Number : ");

t1=new JTextField();

firstNum.setBounds(20, 72, 100, 30);

t1.setBounds(250, 72, 100,30);

secondNum=new JLabel("Second Number : ");

t2=new JTextField();

secondNum.setBounds(20, 197, 100, 30);

t2.setBounds(250, 197, 100,30);

resultOfNum = new JLabel("Result is : ");

t3=new JTextField();

resultOfNum.setBounds(20, 322, 100, 30);

t3.setBounds(250, 197, 100,30);

addButton=new JButton("+");

subButton=new JButton("-");

multiButton=new JButton("\*");

divButton=new JButton("/");

JPanel p= new JPanel();

p.add(addButton);

p.add(subButton);

p.add(multiButton);

p.add(divButton);

// addButton.setBounds(30, 447, 40, 40);

// addButton.setBounds(80, 447, 40, 40);

// addButton.setBounds(130, 447, 40, 40);

// addButton.setBounds(180, 447, 40, 40);

jf.add(firstNum);

jf.add(t1);

jf.add(secondNum);

jf.add(t2);

jf.add(resultOfNum);

jf.add(t3);

jf.add(p);

// jf.add(addButton);

// jf.add(subButton);

// jf.add(multiButton);

// jf.add(divButton);

addButton.addActionListener(this);

subButton.addActionListener(this);

multiButton.addActionListener(this);

divButton.addActionListener(this);

jf.pack();

jf.setLocationRelativeTo(null);

}

public static void main(String[] args) {

new calculator();

}

@Override

public void actionPerformed(ActionEvent e) {

try {

a=Double.parseDouble(t1.getText());

b=Double.parseDouble(t2.getText());

if(e.getSource()==addButton)

{

result=a+b;

t3.setText(""+result);

}

else if(e.getSource()==subButton)

{

result=a-b;

t3.setText(""+result);

}

else if(e.getSource()==multiButton)

{

result=a\*b;

t3.setText(""+result);

}

else if(e.getSource()==divButton)

{

try {

result=a/b;

t3.setText(""+result);

}

catch (Exception e2) {

t3.setText("Invalid Trying to divide by zero");

}

}

}

catch(Exception e1){

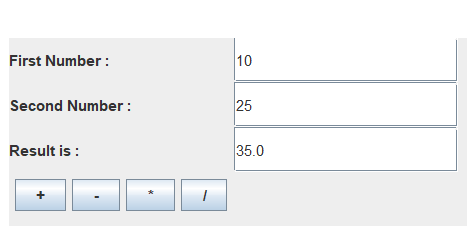
t3.setText("Invalid");

}

}

}

**Output –**

****

**Exp -9**

**Problem Statement –**

**Develop a GUI create a three menu item in a menu. If clicked on 1st item it should develop welcome message.**

**If clicked on 2nd display the dialog box to select the food items and display the bill.**

**If clicked on 3rd open a successful dialog box and once clicked on ok it should go to directory, select a file should display the content of file**

**Program –**

import java.io.\*;

import java.awt.event.\*;

import javax.swing.\*;

import java.awt.\*;

public class GUI extends WindowAdapter implements ActionListener,ItemListener{

JFrame f;

JTextArea ta;

JMenuBar mb;

JMenu exit,menu;

JMenuItem m4,item1,item2,item3;

File fi;

JFileChooser j;

JScrollPane s;

Checkbox c1,c2,c3;

JButton order;

int bill;

int co1,co2,co3;

GUI(){

f=new JFrame("GUI");

f.setLayout(null);

ta=new JTextArea(1000,1000);

//ta.setBounds(30,170,420,200);

s=new JScrollPane(ta);

s.setBounds(30,30,430,340);

ta.setEditable(false);

s.setVisible(false);

f.add(s);

c1=new Checkbox("Pizza@100");

c2=new Checkbox("Cold Coffee@60");

c3=new Checkbox("Fries@50");

order=new JButton("Order");

c1.setBounds(165,50 ,100,50);

c2.setBounds(165,100 ,100,50);

c3.setBounds(165,150 ,100,50);

order.setBounds(165,200,100,40);

c1.addItemListener(this);

c2.addItemListener(this);

c3.addItemListener(this);

order.addActionListener(this);

c1.setVisible(false);

c2.setVisible(false);

c3.setVisible(false);

order.setVisible(false);

f.add(order);

f.add(c1);

f.add(c2);

f.add(c3);

mb = new JMenuBar();

menu =new JMenu("Menu");

exit=new JMenu("Exit");

item1=new JMenuItem("Item 1");

item2=new JMenuItem("Item 2");

item3=new JMenuItem("Item 3");

item1.addActionListener(this);

item2.addActionListener(this);

item3.addActionListener(this);

m4=new JMenuItem("exit");

m4.addActionListener(this);

menu.add(item1);

menu.add(item2);

menu.add(item3);

exit.add(m4);

mb.add(menu);

mb.add(exit);

f.getContentPane().add(BorderLayout.NORTH,mb);

f.addWindowListener(this);

f.setJMenuBar(mb);

f.setSize(500,450);

f.setVisible(true);

f.setDefaultCloseOperation(JFrame.DO\_NOTHING\_ON\_CLOSE);

f.setResizable(false);

f.setLocationRelativeTo(null);

}

public static void main(String[] args) {

new GUI();

}

public void getFilePath() {

JOptionPane.showMessageDialog(f,"Success");

ta.setText("");

j=new JFileChooser("C:\\Users\\HP\\OneDrive\\Desktop\\java\\java

programs");

//j.setSelectedFile(new File("VCard.txt"));

j.showOpenDialog(null);

fi=j.getSelectedFile();

try {

readFile();

}catch(Exception e) {}

}

public void readFile() throws IOException{

FileReader fr=new FileReader(fi);

String s="";

int c;

while((c=fr.read())!=-1)

{

s+=(char)c;

}

s+="\n\nRead Succesfully";

ta.setText(s);

fr.close();

}

void getBill()

{

String s;

s="";

if(co1==0) {

s+="Pizza = 100\n";

}

if(co2==0) {

s+="Cold Coffee = 60\n";

}

if(co3==0) {

s+="Fries = 50\n";

}

s+="\nYour Bill is RS "+bill;

JOptionPane.showMessageDialog(f, s);

c1.setState(false);

c2.setState(false);

c3.setState(false);

co1=1;

co2=1;

co3=1;

bill=0;

}

@Override

public void actionPerformed(ActionEvent e) {

if(e.getSource()==item1) {

s.setVisible(false);

c1.setVisible(false);

c2.setVisible(false);

c3.setVisible(false);

order.setVisible(false);

ta.setText("Welcome");

s.setVisible(true);

}

if(e.getSource()==item2) {

s.setVisible(false);

bill=0;

co1=1;

co2=1;

co3=1;

c1.setVisible(true);

c2.setVisible(true);

c3.setVisible(true);

order.setVisible(true);

}

if(e.getSource()==item3) {

s.setVisible(false);

c1.setVisible(false);

c2.setVisible(false);

c3.setVisible(false);

order.setVisible(false);

getFilePath();

s.setVisible(true);

}

if(e.getSource()==m4) {

System.exit(0);

}

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

getBill();

}

}

@Override

public void windowClosing(WindowEvent e) {

int a=JOptionPane.showConfirmDialog(f,"Are you sure");

if(a==JOptionPane.YES\_OPTION) {

f.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

}

}

@Override

public void itemStateChanged(ItemEvent e) {

if(e.getSource()==c1 ) {

if(c1.getState()==true && co1==1) {

bill+=100;

co1--;

}else {

if(co1==0) {

bill-=100;

co1++;

}

}

}

if(e.getSource()==c2 ) {

if(c2.getState()==true && co2==1) {

bill+=60;

co2--;

}

else {

if(co2==0) {

bill-=60;

co2++;

}

}

}

if(e.getSource()==c3 ){

if(c3.getState()==true&& co3==1 ) {

bill+=50;

co3--;

}else {

if(co3==0) {

bill-=50;

co3++;

}

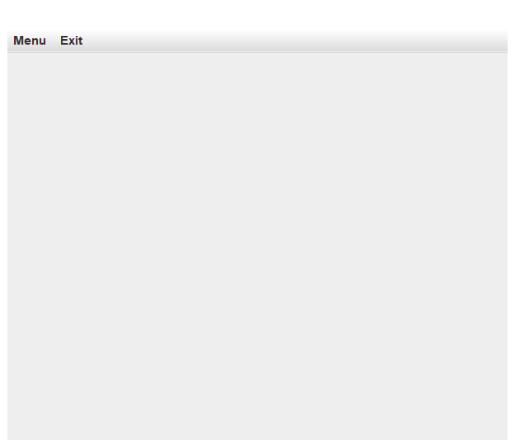
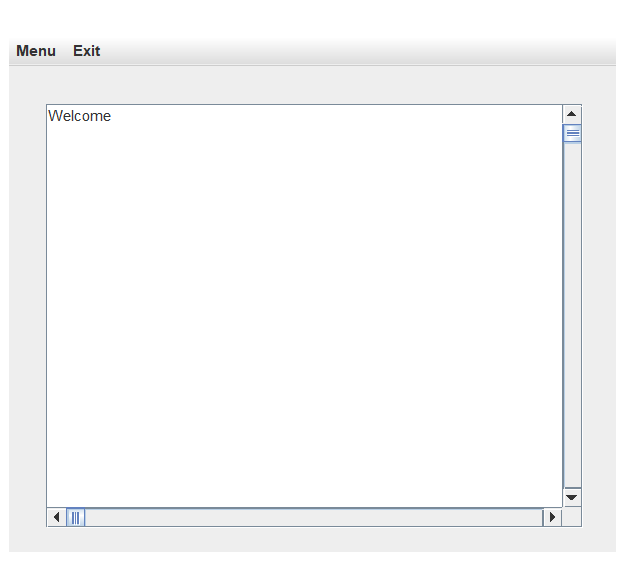
}

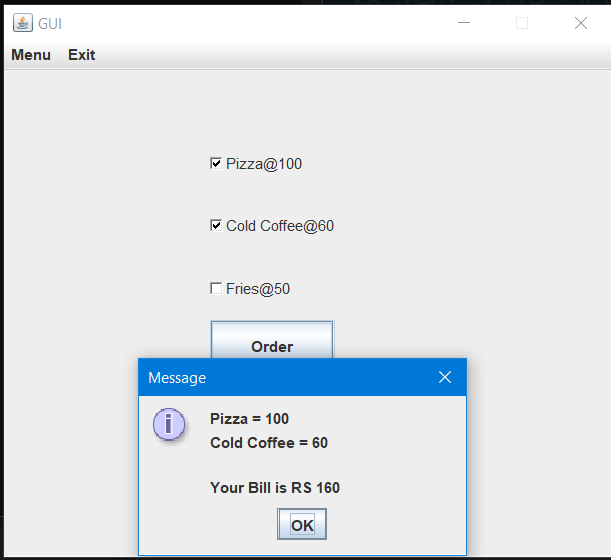
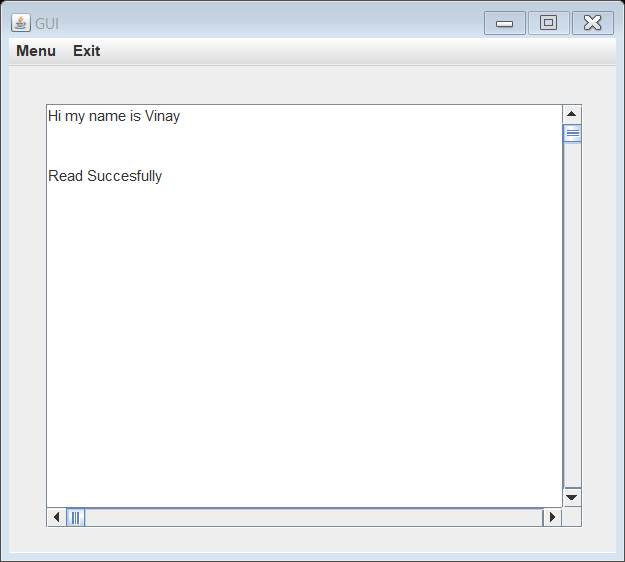
}

}

}

**Output –**

**Exp -10**

**Problem Statement –**

**Develop a socket programming where a client sends a message to server,server prints message and reply back to client ,after reading clients prints the message**

**Program –**

**//Server side->**

import java.io.\*;

import java.net.\*;

public class MyServer {

public static void main(String[] args) throws Exception {

ServerSocket ss=new ServerSocket(5555);

Socket s=ss.accept();

DataInputStream dis=new DataInputStream(s.getInputStream());

String str=(String)dis.readUTF();

System.out.println("mesage= "+str);

DataOutputStream dos=new DataOutputStream(s.getOutputStream());

dos.writeUTF("Reply= Hello Client");

dos.close();

ss.close();

}

}

**//client side->**

import java.io.\*;

import java.net.\*;

public class MyClient {

public static void main(String[] args) throws Exception {

Socket ss=new Socket("localhost",5555);

//Socket s=ss.accept();

DataOutputStream dos=new DataOutputStream(ss.getOutputStream());

dos.writeUTF("Hello Server");

DataInputStream dis=new DataInputStream(ss.getInputStream());

String str=(String)dis.readUTF();

System.out.println("mesage= "+str);

dos.flush();

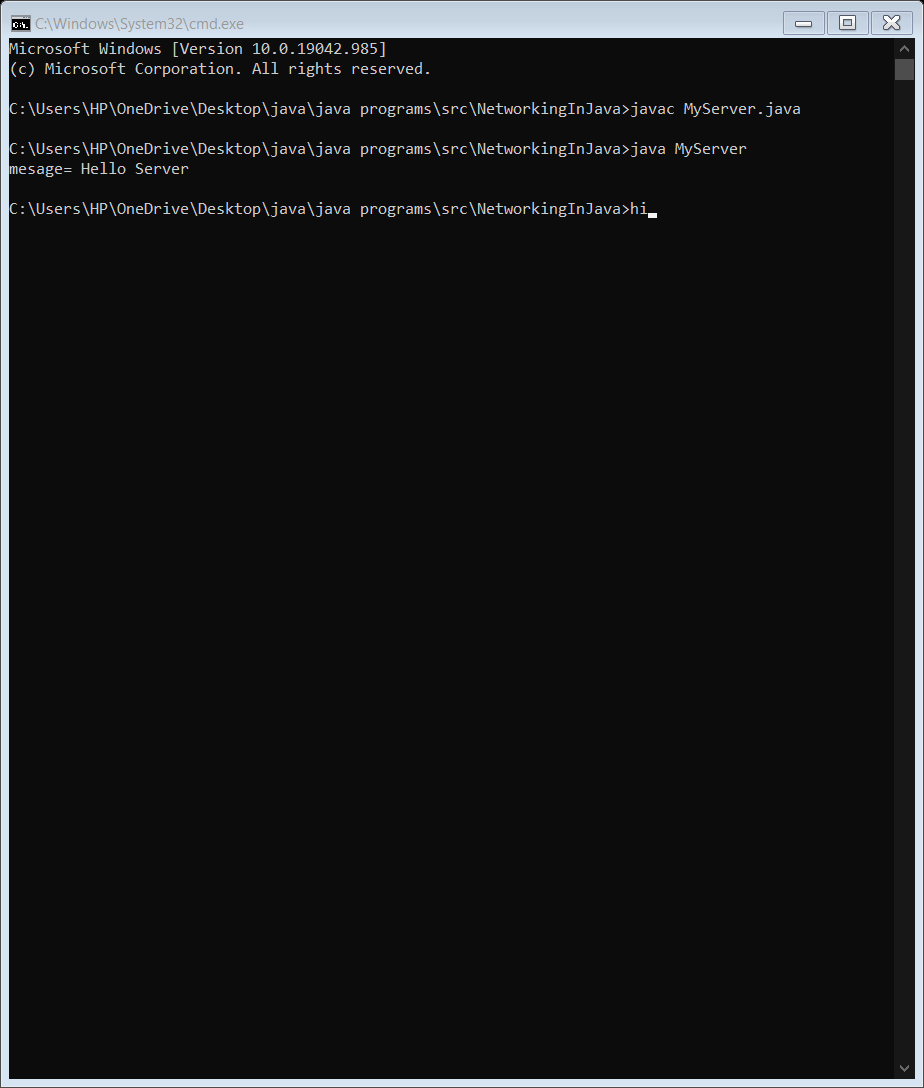
dos.close();

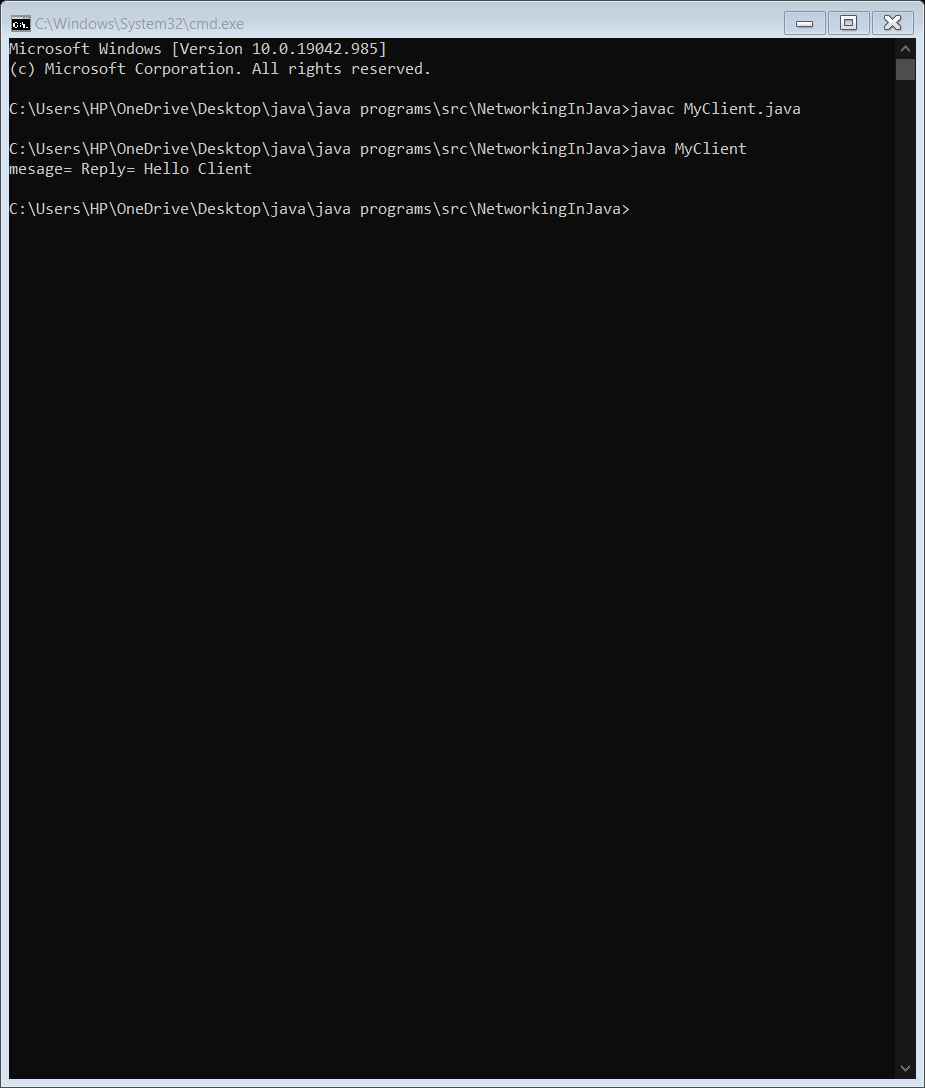
ss.close();

}

}

**Output –**





**Exp -11**

**Problem Statement –**

**write a Program that bounces a ball inside a JPanel. The ball should be moving with a mousePressed Event. when the ball hits the edge of the JPanel It should bounce off the edge and continue in opposite direction. The ball shoiuld be updated using a Runnable.**

**Program –**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class Exp\_BouncingBall extends JPanel implements Runnable

{

JFrame f;

int x=0,y=0;

int angleX=1,angleY=1;

int speed=2;

public void run()

{

while(true)

{

move();

repaint();

try

{

Thread.sleep(10);

}

catch(InterruptedException e)

{

e.printStackTrace();

}

}

}

public void move()

{

if(x+angleX<0)

angleX=+speed;

else if(x+angleX+25>getWidth())

angleX=-10;

if(y+angleY<0)

angleY=+speed;

else if(y+angleY+25>getHeight())

angleY=-10;

x=x+angleX;

y=y+angleY;

}

public void paintComponent(Graphics g)

{

super.paintComponent(g);

g.setColor(Color.yellow);

g.fillOval(x,y,25,25);

}

public static void main(String[] args) throws InterruptedException

{

JFrame f=new JFrame("Moving Ball");

Exp\_BouncingBall e=new Exp\_BouncingBall();

final Thread t1=new Thread(e);

f.add(e);

f.setSize(400,400);

f.setVisible(true);

f.setLayout(null);

f.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

e.addMouseListener(new MouseAdapter()

{

public void mousePressed(MouseEvent e)

{

t1.start();

}

Public void mouseReleased(MouseEvent e)

{

t1.stop();

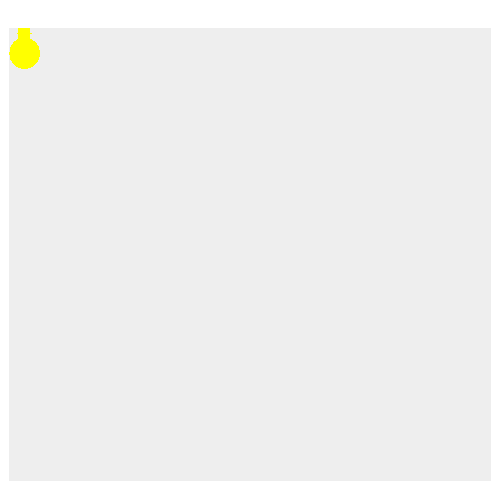
}

});

}

}

**Output –**

****

**Exp -12**

**Problem Statement –**

**Connect database and execute queries to fetch data and display on GUI**

**Program –**

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import java.sql.\*;

//set classpath=C:\oraclexe\app\oracle\product\10.2.0\server\jdbc\lib\ojdbc14.jar;.;

public class Exp12 implements ActionListener{

JFrame f;

JLabel l1,l2,l3,l4,l5,l6;

JTextField t1,t2,t3,t4,t5;

JButton b;

Exp12()

{

f=new JFrame("Student Database");

f.setLayout(null);

l1=new JLabel("Enter name : ");

l1.setBounds(30,30,200,30);

t1=new JTextField();

t1.setBounds(140,30,250,30);

f.add(l1);

f.add(t1);

b=new JButton("Enter");

b.setBounds(150,70,100,40);

b.addActionListener(this);

f.add(b);

l2=new JLabel("Student Details");

l2.setBounds(130,120,200,30);

l2.setForeground(Color.RED);

l2.setFont(new Font("Arial",Font.BOLD,16));

f.add(l2);

l3=new JLabel("Roll number: ");

l3.setBounds(30,160,200,30);

t2=new JTextField();

t2.setBounds(140,160,250,30);

t2.setEditable(false);

f.add(l3);

f.add(t2);

l4=new JLabel("Name : ");

l4.setBounds(30,200,200,30);

t3=new JTextField();

t3.setBounds(140,200,250,30);

t3.setEditable(false);

f.add(l4);

f.add(t3);

l5=new JLabel("CGPA : ");

l5.setBounds(30,240,200,30);

t4=new JTextField();

t4.setBounds(140,240,250,30);

t4.setEditable(false);

f.add(l5);

f.add(t4);

l6=new JLabel("College : ");

l6.setBounds(30,280,200,30);

t5=new JTextField();

t5.setBounds(140,280,250,30);

t5.setEditable(false);

f.add(l6);

f.add(t5);

f.setSize(500,400);

f.setVisible(true);

f.setResizable(false);

f.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

f.setLocationRelativeTo(null);

}

public void connectDatabase() throws Exception

{

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

Connection connect=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system","brockvs28");

Statement st=connect.createStatement();

String s=t1.getText();

ResultSet rs=st.executeQuery("select \* from Student\_Details where name='"+s+"'");

if(rs.next())

{

t2.setText(""+rs.getInt(1));

t3.setText(rs.getString(2));

t4.setText(""+rs.getDouble(3));

t5.setText(rs.getString(4));

}

else

{

t2.setText("Data Not Available");

t3.setText("Data Not Available");

t4.setText("Data Not Available");

t5.setText("Data Not Available");

}

connect.close();

}

public static void main(String[] args) {

new Exp12();

}

public void actionPerformed(ActionEvent e)

{

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

try {

connectDatabase();

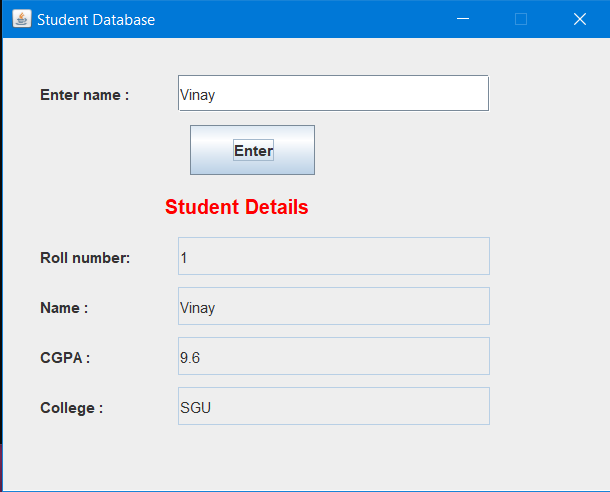
}catch(Exception e1) {System.out.println(e1);}

}

}

}

**Output –**

****