INDEX

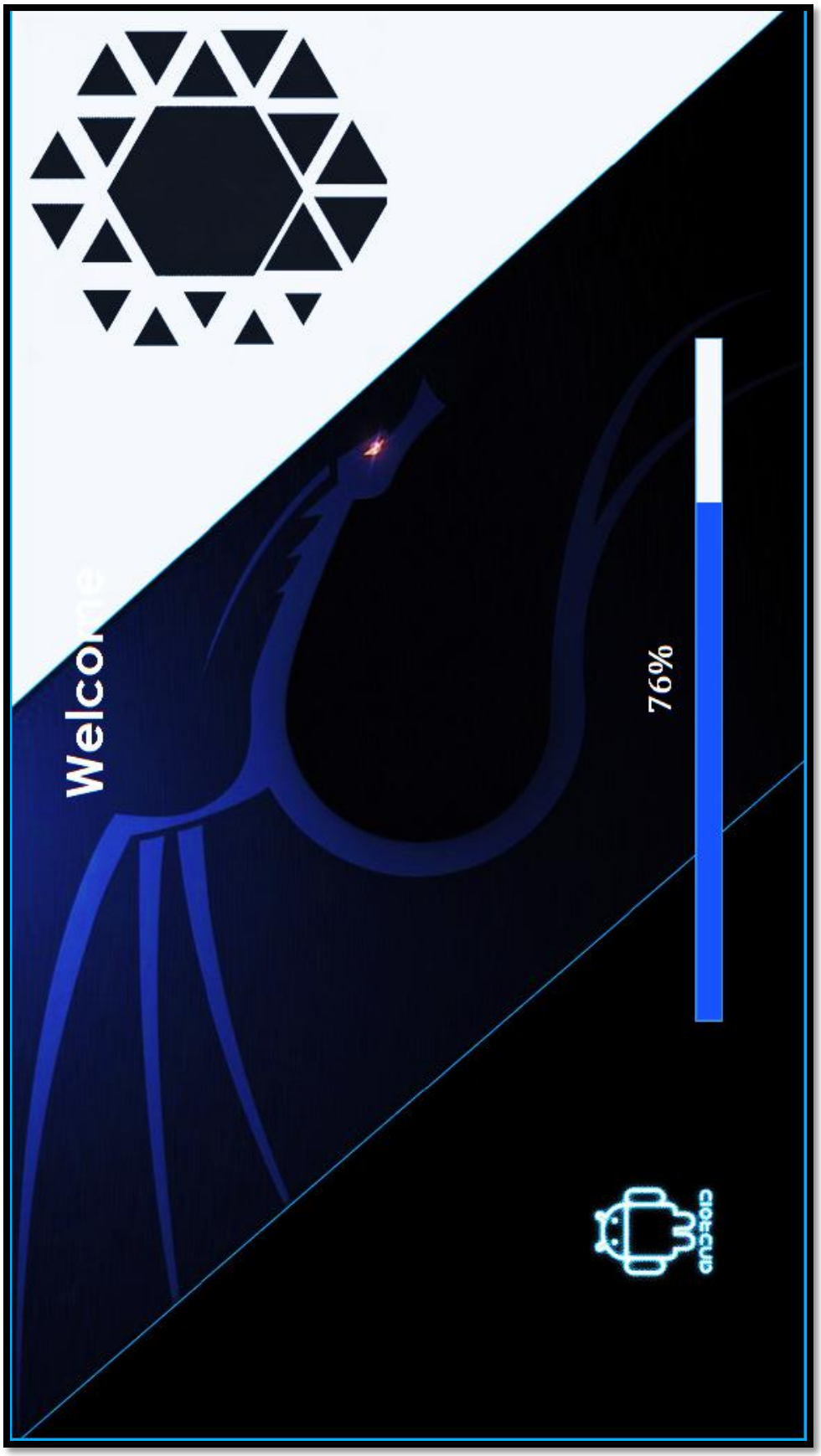
ACKNOWLEDGEMENT

In the accomplishment of this project successfully, many people have best owned upon me their blessings and the heart pledged support, this time I'm utilizing to thank all the people who have been concerned with the project.

Primarily I would thank god for being able to complete this project with success. Then I would like to thank my principal Mrs. Shraddha Sudame and Informatics Practices teacher Mrs. Mili Guha whose valuable guidance has been the ones that helped me patch this project and make it full proof success. There suggestions and instructions has served as the major contributor towards the completion of the project.

Then I would like to thank my parents and friends who have helped me with valuable suggestions and guidance has been helpful in various phases of the completion of the project.

FRAME 1 (SPLASH)

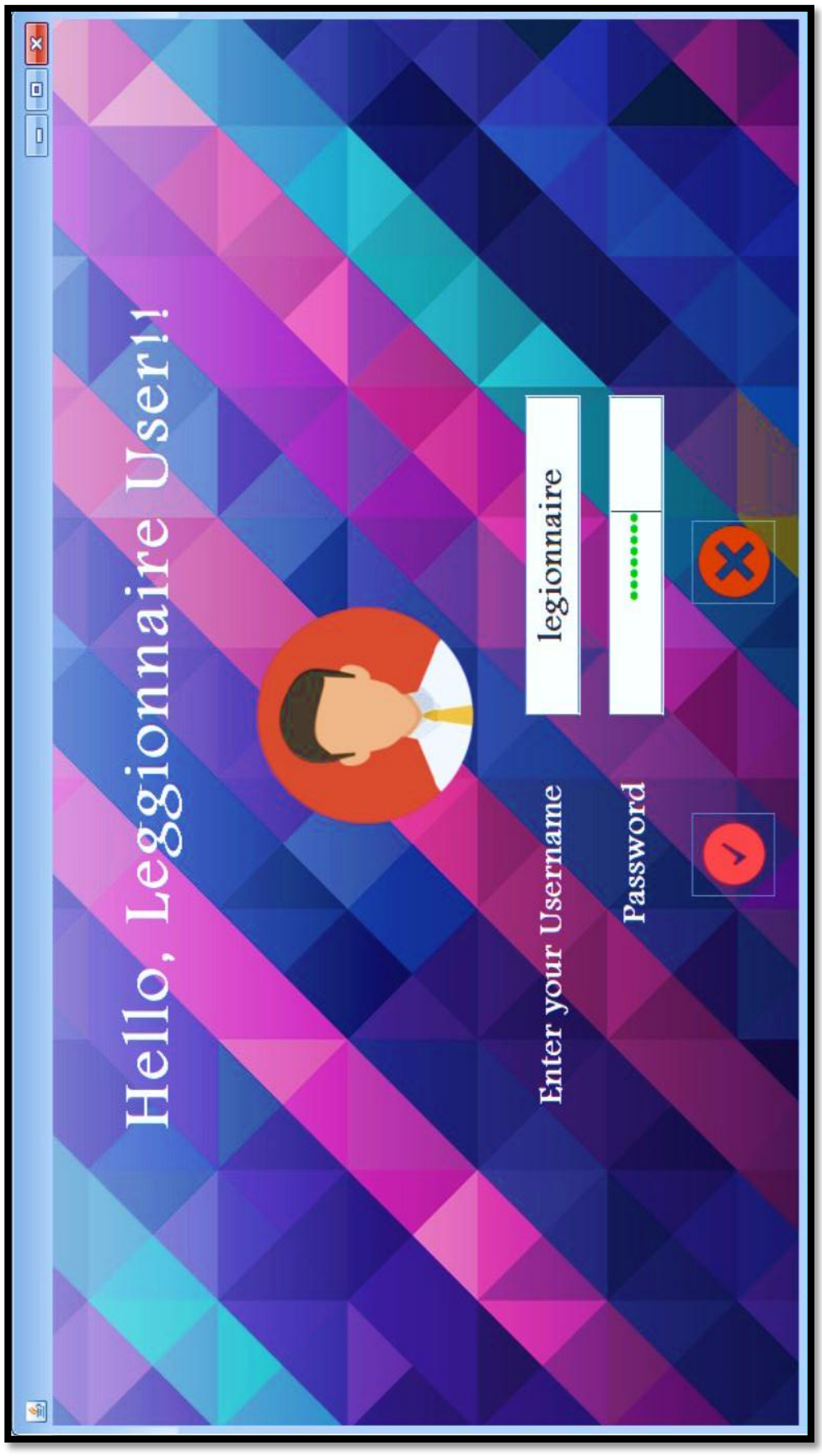


CODING:

CODED IN LEGIONNAIRE_OS. JAVA CLASS

```
package legionnaire_os;
import javax.swing.JOptionPane;
/* @author Mandeep M. Dalavi */
public class Legionnaire_OS {
   public static void main(String[] args) {
      Splash Splash = new Splash();
      Login Login = new Login();
      Splash.setVisible(true);
      try{
            for(int i = 0; i<=100; i++)
             {
               Thread.sleep(60);
               Splash.loadingnum.setText(Integer.toString(i)+"%");
               Splash.loadingbar.setValue(i);
               if(i==100)
               {
                  Splash.setVisible(false);
                  Login.setVisible(true);
               }
            }
       }
      catch(Exception e){
            JOptionPane.showMessageDialog(Splash, "Error in Connectivity");
        }
   }
}
```

FRAME 2 (LOGIN)



CODING:

```
package legionnaire_os;
/* @author Mandeep M. Dalavi */
import java.sql.*;
import javax.swing.ImageIcon;
import javax.swing.JOptionPane;
public class Login extends javax.swing.JFrame {
   String curusername;
   String curpassword;
   public Login() {
      initComponents();
//--------------------JDBC part getting username and password--------------------//
try {
      Class.forName("java.sql.Driver");
      Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
      Statement stmt = (Statement) con.createStatement();
      String query = "Select * from userpass";
      ResultSet rs = stmt.executeQuery(query);
      if (rs.first())
        {
          curusername = rs.getString(1);
          curpassword = rs.getString(2);
        }
   }
catch (Exception e) {
      JOptionPane.showMessageDialog(this, "Error in Connectivity!");
   }
```

```
//-----------------------JDBC part getting background-------------------------//
try {
      Class.forName("java.sql.Driver");
      Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
      Statement stmt = (Statement) con.createStatement();
      String query = "Select * from backgrounds";
      ResultSet rs = stmt.executeQuery(query);
      if (rs.first()) {
            String loginbackg = rs.getString(2);
            Loginlabel.setIcon(new ImageIcon(getClass().getResource(loginbackg)));
        }
  }
catch (Exception e) {
     JOptionPane.showMessageDialog(this, "Error in Connectivity!");
  }
//----------------------------JDBC part getting avatar-----------------------------//
try {
     Class.forName("java.sql.Driver");
     Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
     Statement stmt = (Statement) con.createStatement();
     String query = "Select * from useravatar";
     ResultSet rs = stmt.executeQuery(query);
     if (rs.first()) {
           String avatar = rs.getString(1);
           Userlabel.setIcon(new ImageIcon(getClass().getResource(avatar)));
        }
     }
```

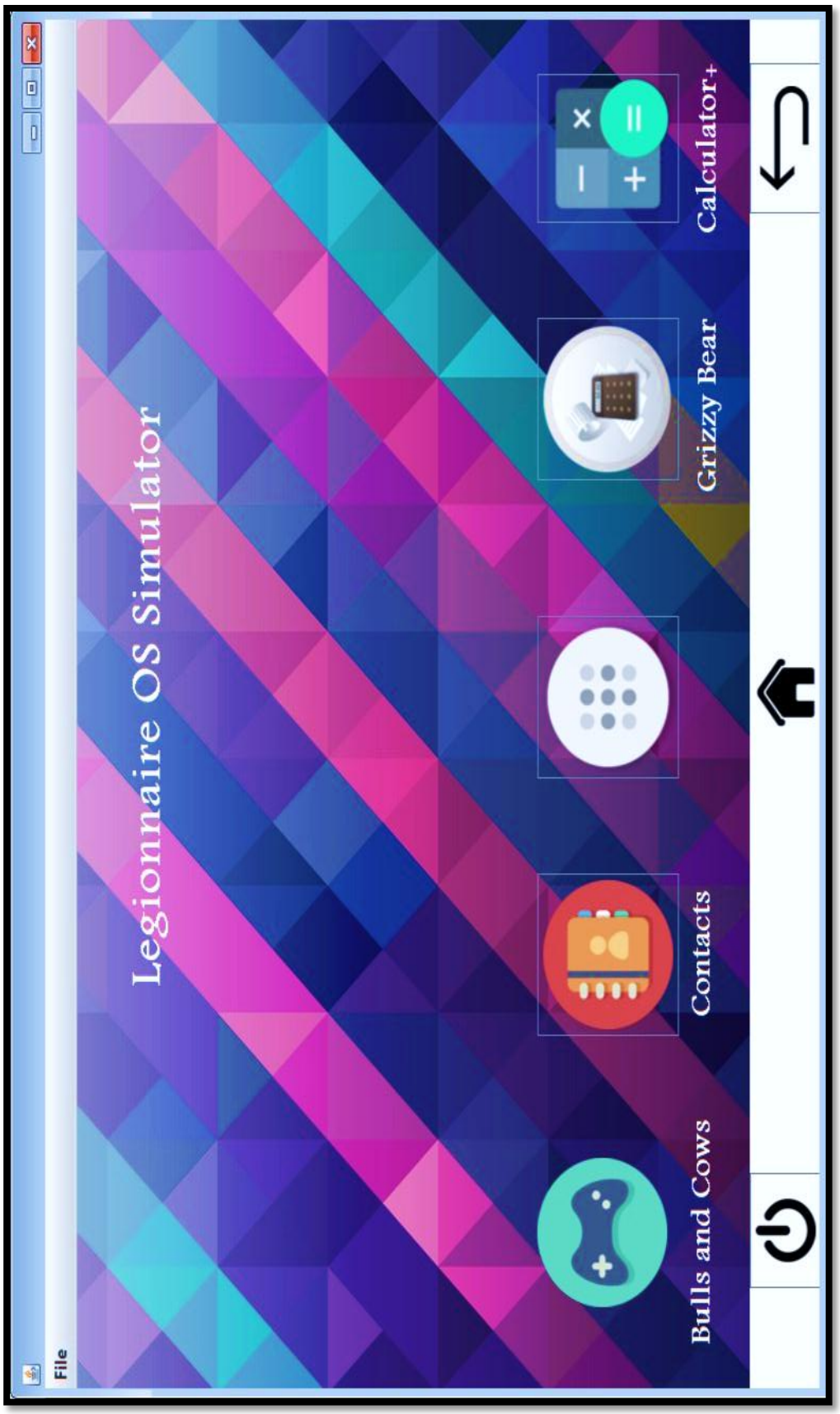
```
catch (Exception e)
   {
        JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
    }
```

```
private void SubmitbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    String user = UserTF.getText();
    String pass = new String(PassTF.getPassword());
    if (user.equalsIgnoreCase(curusername) && pass.equalsIgnoreCase(curpassword))
    {
       JOptionPane.showMessageDialog(this,"You have been successfully logged in." );
       new Home().setVisible(true);
       this.dispose();
    }
   else
    {
       JOptionPane.showMessageDialog(this, "Password and User ID do not match. Please try
again.", "Incorrect username or password", JOptionPane.ERROR_MESSAGE);
     }
}
```

```
private void CancelbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    if (JOptionPane.showConfirmDialog(this, "Are you sure you want to exit?", "Confirmation",
JOptionPane.YES_NO_OPTION) == JOptionPane.YES_OPTION)
    {
       System.exit(0);
     }
}
```

```
private void formWindowActivated(java.awt.event.WindowEvent evt) {
    this.setResizable(true);
    this.setSize(1026, 575);
}
-----------------------------------------------------------------------------------------------------
private void PassTFMouseClicked(java.awt.event.MouseEvent evt) {
     PassTF.setText("");
}
-----------------------------------------------------------------------------------------------------
private void UserTFCaretUpdate(javax.swing.event.CaretEvent evt) {
String user = UserTF.getText();
if(user.equalsIgnoreCase(curusername)) {
   UserTF.setForeground(Color.GREEN);
}
else {
   UserTF.setForeground(Color.RED);
}
   }
-----------------------------------------------------------------------------------------------------
   private void PassTFCaretUpdate(javax.swing.event.CaretEvent evt) {
String pass = new String(PassTF.getPassword());
if(pass.equalsIgnoreCase(curpassword)) {
   PassTF.setForeground(Color.GREEN);
}
else {
   PassTF.setForeground(Color.RED);
}
   }
```

FRAME 3 (HOME)


File
Legionnaire OS Simulator
Bulls and Cows
Contacts
Grizzy Bear
Calculator+

CODING:

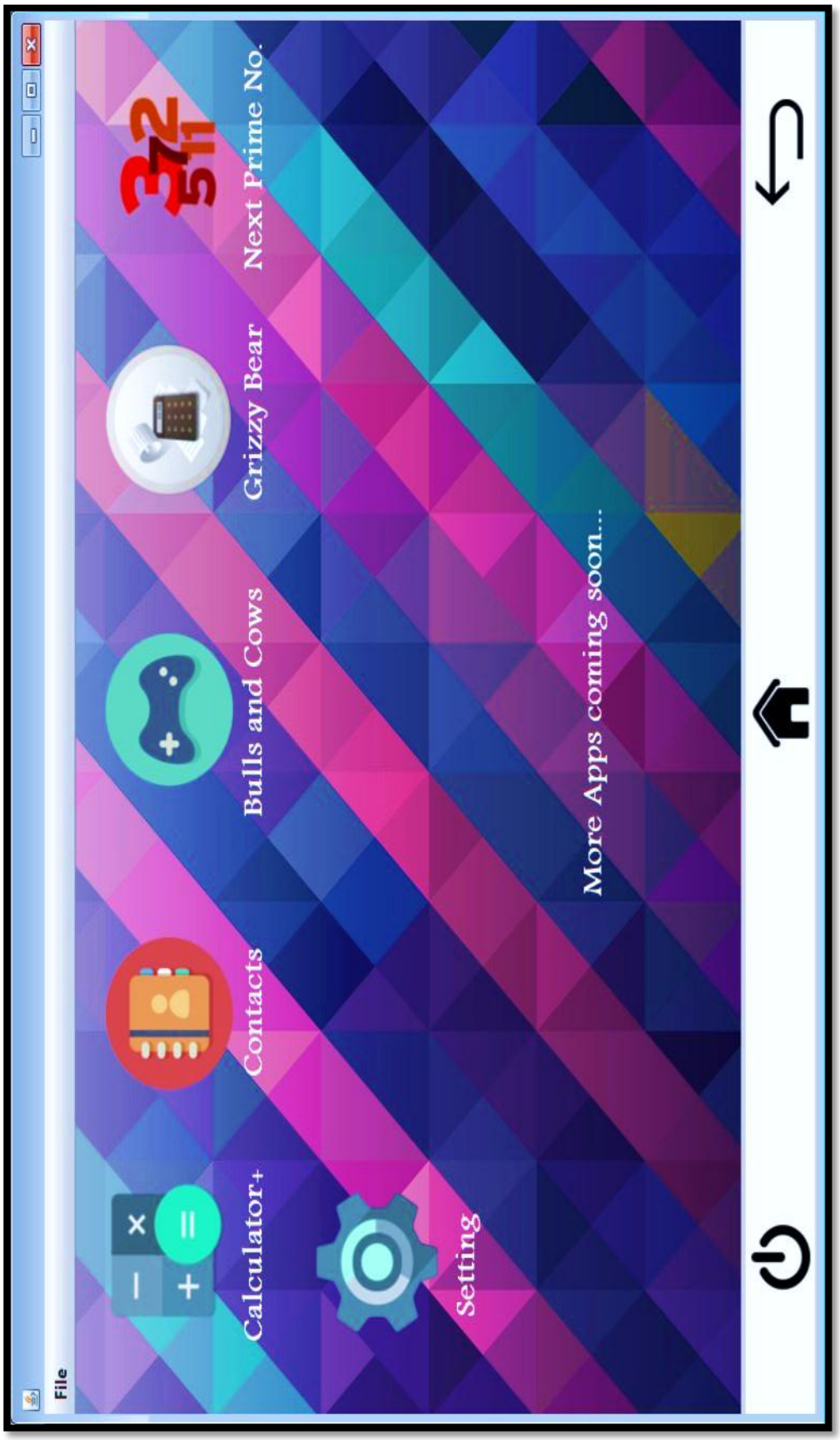
```
package legionnaire_os;

import java.sql.*;

import javax.swing.ImageIcon;

import javax.swing.JOptionPane;

/* @author Mandeep M. Dalavi */

public class Home extends javax.swing.JFrame {

   public Home() {

      initComponents();

//----------------------------JDBC part getting background-------------------------------//

try {

      Class.forName("java.sql.Driver");

      Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");

      Statement stmt = (Statement) con.createStatement();

      String query = "SELECT * from Backgrounds";

      ResultSet rs = stmt.executeQuery(query);

      if (rs.first())

      {

         String homebackg = rs.getString(1);

         Homelabel.setIcon(new ImageIcon(getClass().getResource(homebackg)));

      }

   }

catch (Exception e) {

      JOptionPane.showMessageDialog(this, “Error in Connectivity!”);

       }

  }
```

```
private void BackbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    new Login().setVisible(true);
    this.setVisible(false);
 }
-----------------------------------------------------------------------------------------------------
private void MenubuttonActionPerformed(java.awt.event.ActionEvent evt) {
    new Apps().setVisible(true);
    this.setVisible(false);
 }
-----------------------------------------------------------------------------------------------------
private void formWindowActivated(java.awt.event.WindowEvent evt) {
    this.setResizable(true);
    this.setSize(1026, 694);
 }
-----------------------------------------------------------------------------------------------------
private void JumpHomemenuActionPerformed(java.awt.event.ActionEvent evt) {
    new Home().setVisible(true);
    this.setVisible(false);
 }
-----------------------------------------------------------------------------------------------------
private void JumpAppmenuActionPerformed(java.awt.event.ActionEvent evt) {
    new Apps().setVisible(true);
    this.setVisible(false);
 }
-----------------------------------------------------------------------------------------------------
private void ExitOSmenuActionPerformed(java.awt.event.ActionEvent evt) {
    if (JOptionPane.showConfirmDialog(this, "Are you sure you want to exit?", "Confirmation",
JOptionPane.YES_NO_OPTION) == JOptionPane.YES_OPTION)
{
```

```
            System.exit(0);
        }   }
-----------------------------------------------------------------------------------------------------
private void PowerbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    if (JOptionPane.showConfirmDialog(this, "Are you sure you want to exit?", "Confirmation",
JOptionPane.YES_NO_OPTION) == JOptionPane.YES_OPTION)
        {
            System.exit(0);
        }   }
-----------------------------------------------------------------------------------------------------
private void HomebuttonActionPerformed(java.awt.event.ActionEvent evt) {
    new Home().setVisible(true);
    this.setVisible(false);
   }
-----------------------------------------------------------------------------------------------------
private void ContactsbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    new Contacts_App().setVisible(true);
    this.setVisible(false);
   }
-----------------------------------------------------------------------------------------------------
private void GrizzyBearbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    new GrizzyBear_User_Login_Page().setVisible(true);
    this.setVisible(false);
   }
-----------------------------------------------------------------------------------------------------
private void CalculatorbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    new Calculator_App().setVisible(true);
     this.setVisible(false);
 }
```

```
private void BullsandCowsbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    new Bulls_and_Cows().setVisible(true);
    new Enter_no().setVisible(true);
    this.setVisible(false);
 }
```

FRAME 4 (APPS)


File
Calculator+
Contacts
Bulls and Cows
Grizzy Bear
Next Prime No.
Setting
More Apps coming soon...

CODING:

```
package legionnaire_os;

import java.sql.*;

import javax.swing.ImageIcon;

import javax.swing.JOptionPane;

/* @author Mandeep M. Dalavi */

public class Apps extends javax.swing.JFrame {

   public Apps() {

      initComponents();

//---------------------------------JDBC part getting background-----------------------------//

try {

      Class.forName("java.sql.Driver");

      Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");

      Statement stmt = (Statement) con.createStatement();

      String query = "SELECT * from Backgrounds";

      ResultSet rs = stmt.executeQuery(query);

      if (rs.first())

      {

         String homebackg = rs.getString(1);

         Appslabel.setIcon(new ImageIcon(getClass().getResource(homebackg)));

      }

   }

catch (Exception e) {

      JOptionPane.showMessageDialog(this, “Error in Connectivity!”);

   }

}
```

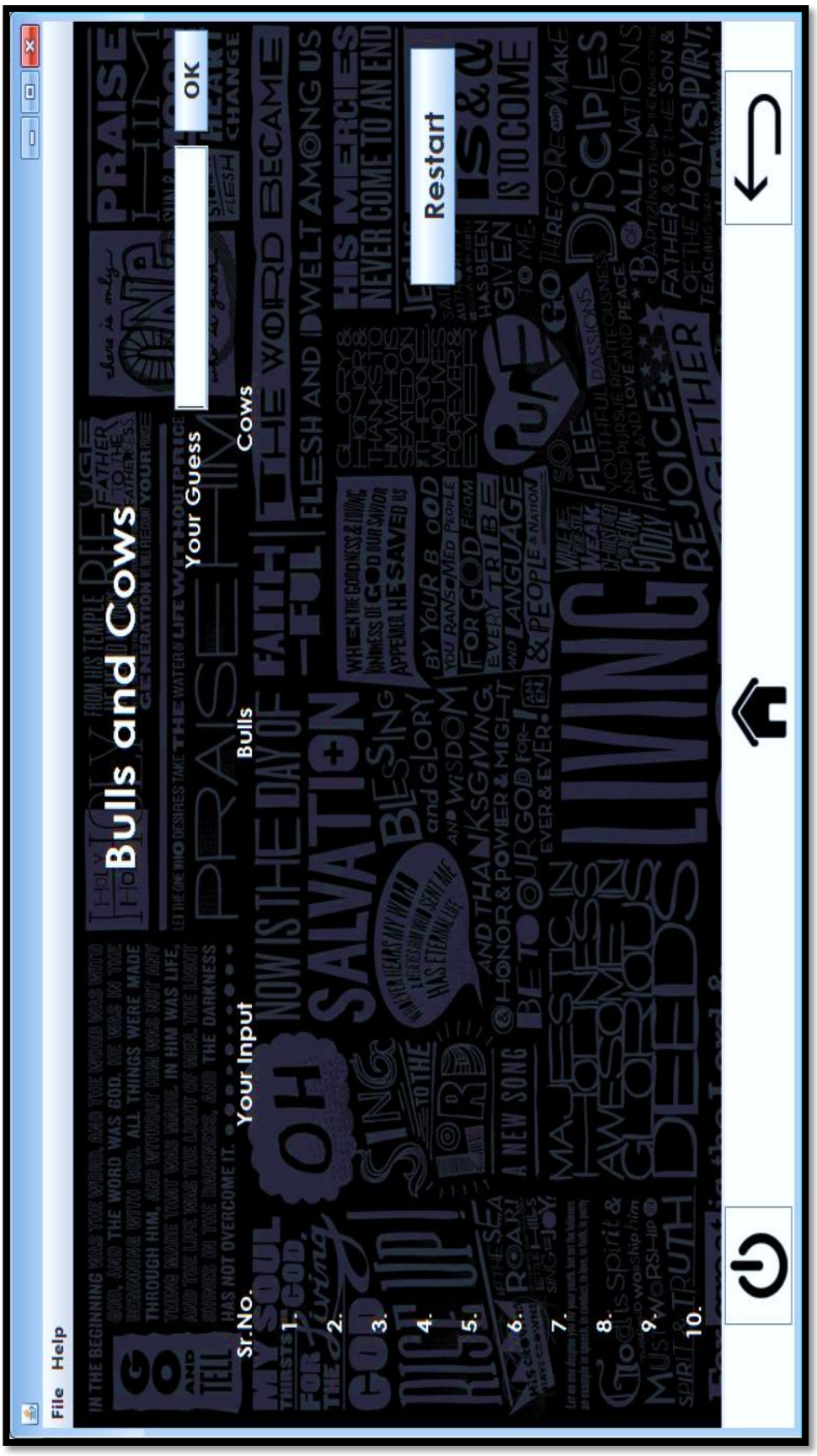
```
private void SettingappActionPerformed(java.awt.event.ActionEvent evt) {
    new Setting_App().setVisible(true);
    this.setVisible(false);
  }
-----------------------------------------------------------------------------------------------------
private void BackbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    new Home().setVisible(true);
    this.setVisible(false);
  }
-----------------------------------------------------------------------------------------------------
private void JumpHomemenuActionPerformed(java.awt.event.ActionEvent evt) {
    new Home().setVisible(true);
    this.setVisible(false);
  }
-----------------------------------------------------------------------------------------------------
private void JumpAppmenuActionPerformed(java.awt.event.ActionEvent evt) {
    new Apps().setVisible(true);
    this.setVisible(false);
  }
-----------------------------------------------------------------------------------------------------
private void ExitOSmenuActionPerformed(java.awt.event.ActionEvent evt) {
    if(JOptionPane.showConfirmDialog(this, "Are you sure you want to exit?", "Confirmation",
JOptionPane.YES_NO_CANCEL_OPTION)==JOptionPane.YES_OPTION)
      {
        System.exit(0);
      }
  }
```

```
private void CalculatorappActionPerformed(java.awt.event.ActionEvent evt) {
    new Calculator_App().setVisible(true);
    this.setVisible(false);  }
-----------------------------------------------------------------------------------------------------
private void ContactsappActionPerformed(java.awt.event.ActionEvent evt) {
    new Contacts_App().setVisible(true);
    this.setVisible(false);
  }
-----------------------------------------------------------------------------------------------------
private void GrizzyBearappActionPerformed(java.awt.event.ActionEvent evt) {
    new GrizzyBear_User_Login_Page().setVisible(true);
    this.setVisible(false);
  }
-----------------------------------------------------------------------------------------------------
private void NextPrimeNoappActionPerformed(java.awt.event.ActionEvent evt) {
    new Next_Prime_No_App().setVisible(true);
    this.setVisible(false);
  }
-----------------------------------------------------------------------------------------------------
private void PowerbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    if(JOptionPane.showConfirmDialog(this, "Are you sure you want to exit?", "Confirmation",
JOptionPane.YES_NO_CANCEL_OPTION)==JOptionPane.YES_OPTION)
        {
            System.exit(0);
        }   }
-----------------------------------------------------------------------------------------------------
private void HomebuttonActionPerformed(java.awt.event.ActionEvent evt) {
    new Home().setVisible(true);
    this.setVisible(false);    }
```

```
private void formWindowActivated(java.awt.event.WindowEvent evt) {
    this.setResizable(true);
    this.setSize(1025, 707);
  }
```

```
private void BullandCowsappActionPerformed(java.awt.event.ActionEvent evt) {
    new Bulls_and_Cows().setVisible(true);
    new Enter_no().setVisible(true);
    this.setVisible(false);
  }
```

FRAME 5 (BULLS AND COWS)



CODING:

```
package legionnaire_os;

import java.sql.*;

import javax.swing.JOptionPane;

/* @author Mandeep M. Dalavi */

-----------------------------------------------------------------------------------------------------

public class Bulls_and_Cows extends javax.swing.JFrame {

   int wordcount = 1;

   int trycount = 1;

   public Bulls_and_Cows() {

      initComponents();

   }

-----------------------------------------------------------------------------------------------------

private void JumpHomemenuActionPerformed(java.awt.event.ActionEvent evt) {

      new Home().setVisible(true);

      this.setVisible(false);

   }

-----------------------------------------------------------------------------------------------------

private void JumpAppmenuActionPerformed(java.awt.event.ActionEvent evt) {

      new Apps().setVisible(true);

      this.setVisible(false);

   }

-----------------------------------------------------------------------------------------------------

private void ExitOSmenuActionPerformed(java.awt.event.ActionEvent evt) {

      if(JOptionPane.showConfirmDialog(this, "Are you sure you want to exit?", "Confirmation",
JOptionPane.YES_NO_CANCEL_OPTION)==JOptionPane.YES_OPTION)

         {        System.exit(0);        }

   }
```

```
private void HomebuttonActionPerformed(java.awt.event.ActionEvent evt) {
      new Home().setVisible(true);
      this.setVisible(false);
   }
-----------------------------------------------------------------------------------------------------
private void BackbuttonActionPerformed(java.awt.event.ActionEvent evt) {
      new Home().setVisible(true);
      this.setVisible(false);
   }
-----------------------------------------------------------------------------------------------------
private void PowerbuttonActionPerformed(java.awt.event.ActionEvent evt) {
      if(JOptionPane.showConfirmDialog(this, "Are you sure you want to exit?", "Confirmation",
JOptionPane.YES_NO_CANCEL_OPTION)==JOptionPane.YES_OPTION)
         {
                  System.exit(0);
         }
   }
-----------------------------------------------------------------------------------------------------
private void formWindowActivated(java.awt.event.WindowEvent evt) {
      this.setResizable(true);
      this.setSize(1025, 707);
   }
-----------------------------------------------------------------------------------------------------
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
      String input = inputTF.getText();
      if (input.length() != 4)
      {
         JOptionPane.showMessageDialog(this, "Please enter a 4 digit word");
         inputTF.setText("");    }
```

```
    else
    {
      if (trycount <= 10)
      {
        int bulls = 0;
        int cows = 0;
        String word = new String();
        char inputch;
        char wordch;
        try {
          Class.forName("java.sql.Driver");
          Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
          Statement stmt = (Statement) con.createStatement();
          String query = "Select word from words where srno=" + wordcount;
          ResultSet rs = stmt.executeQuery(query);
          if (rs.first())
          {
            word = rs.getString(1);
          }
        }
        catch (Exception e) {
          JOptionPane.showMessageDialog(this, "Error in Connectivity!");
        }
        for (int i = 0; i < word.length(); i++)
        {
          for (int j = 0; j < input.length();)
          {
            wordch = word.charAt(i);
```

```
        if (i == j)
        {
            j++;
        }
        else
        {
            inputch = input.charAt(j);
            if (inputch == wordch)
            {
                cows++;
            }
            j++;
        }
    }
}
for (int i = 0; i < word.length(); i++)
{
    wordch = word.charAt(i);
    inputch = input.charAt(i);
    if (inputch == wordch)
    {
        bulls++;
    }
}
switch (trycount) {
    case 1:
        inputlabel1.setText(input);
        bullslabel1.setText(bulls + "");
```

```
    cowslabel1.setText(cows + "");
    break;
  case 2:
    inputlabel2.setText(input);
    bullslabel2.setText(bulls + "");
    cowslabel2.setText(cows + "");
    break;
  case 3:
    inputlabel3.setText(input);
    bullslabel3.setText(bulls + "");
    cowslabel3.setText(cows + "");
    break;
  case 4:
    inputlabel4.setText(input);
    bullslabel4.setText(bulls + "");
    cowslabel4.setText(cows + "");
    break;
  case 5:
    inputlabel5.setText(input);
    bullslabel5.setText(bulls + "");
    cowslabel5.setText(cows + "");
    break;
  case 6:
    inputlabel6.setText(input);
    bullslabel6.setText(bulls + "");
    cowslabel6.setText(cows + "");
    break;
  case 7:
```

```
            inputlabel7.setText(input);
            bullslabel7.setText(bulls + "");
            cowslabel7.setText(cows + "");
            break;
        case 8:
            inputlabel8.setText(input);
            bullslabel8.setText(bulls + "");
            cowslabel8.setText(cows + "");
            break;
        case 9:
            inputlabel9.setText(input);
            bullslabel9.setText(bulls + "");
            cowslabel9.setText(cows + "");
            break;
        case 10:
            inputlabel10.setText(input);
            bullslabel10.setText(bulls + "");
            cowslabel10.setText(cows + "");
            break;
        default:
            break;
    }
    if (trycount == 10 && bulls != 4)
    {
        JOptionPane.showMessageDialog(this, "Sorry! You lost. The word is- " + word + ".",
"You Lost", JOptionPane.PLAIN_MESSAGE);
        inputTF.setText("");
        inputlabel1.setText("");
        bullslabel1.setText("");
```

```
cowslabel1.setText("");
inputlabel2.setText("");
bullslabel2.setText("");
cowslabel2.setText("");
inputlabel3.setText("");
bullslabel3.setText("");
cowslabel3.setText("");
inputlabel4.setText("");
bullslabel4.setText("");
cowslabel4.setText("");
inputlabel5.setText("");
bullslabel5.setText("");
cowslabel5.setText("");
inputlabel6.setText("");
bullslabel6.setText("");
cowslabel6.setText("");
inputlabel7.setText("");
bullslabel7.setText("");
cowslabel7.setText("");
inputlabel8.setText("");
bullslabel8.setText("");
cowslabel8.setText("");
inputlabel9.setText("");
bullslabel9.setText("");
cowslabel9.setText("");
inputlabel10.setText("");
bullslabel10.setText("");
cowslabel10.setText("");
```

```
            trycount = 0;
            if (wordcount <= 40)
            {
               wordcount++;
            }
            else
            {
               wordcount = 1;
            }
         }
         trycount++;
         if (bulls == 4)
         {
            JOptionPane.showMessageDialog(this, "You got that!", "You won",
JOptionPane.PLAIN_MESSAGE);
            inputTF.setText("");
            inputlabel1.setText("");
            bullslabel1.setText("");
            cowslabel1.setText("");
            inputlabel2.setText("");
            bullslabel2.setText("");
            cowslabel2.setText("");
            inputlabel3.setText("");
            bullslabel3.setText("");
            cowslabel3.setText("");
            inputlabel4.setText("");
            bullslabel4.setText("");
            cowslabel4.setText("");
            inputlabel5.setText("");
```

```
        bullslabel5.setText("");
        cowslabel5.setText("");
        inputlabel6.setText("");
        bullslabel6.setText("");
        cowslabel6.setText("");
        inputlabel7.setText("");
        bullslabel7.setText("");
        cowslabel7.setText("");
        inputlabel8.setText("");
        bullslabel8.setText("");
        cowslabel8.setText("");
        inputlabel9.setText("");
        bullslabel9.setText("");
        cowslabel9.setText("");
        inputlabel10.setText("");
        bullslabel10.setText("");
        cowslabel10.setText("");
        trycount = 1;
        if (wordcount <= 40)
        {
           wordcount++;
        }
        else
        {
           wordcount = 1;
        }
     }
  }
```

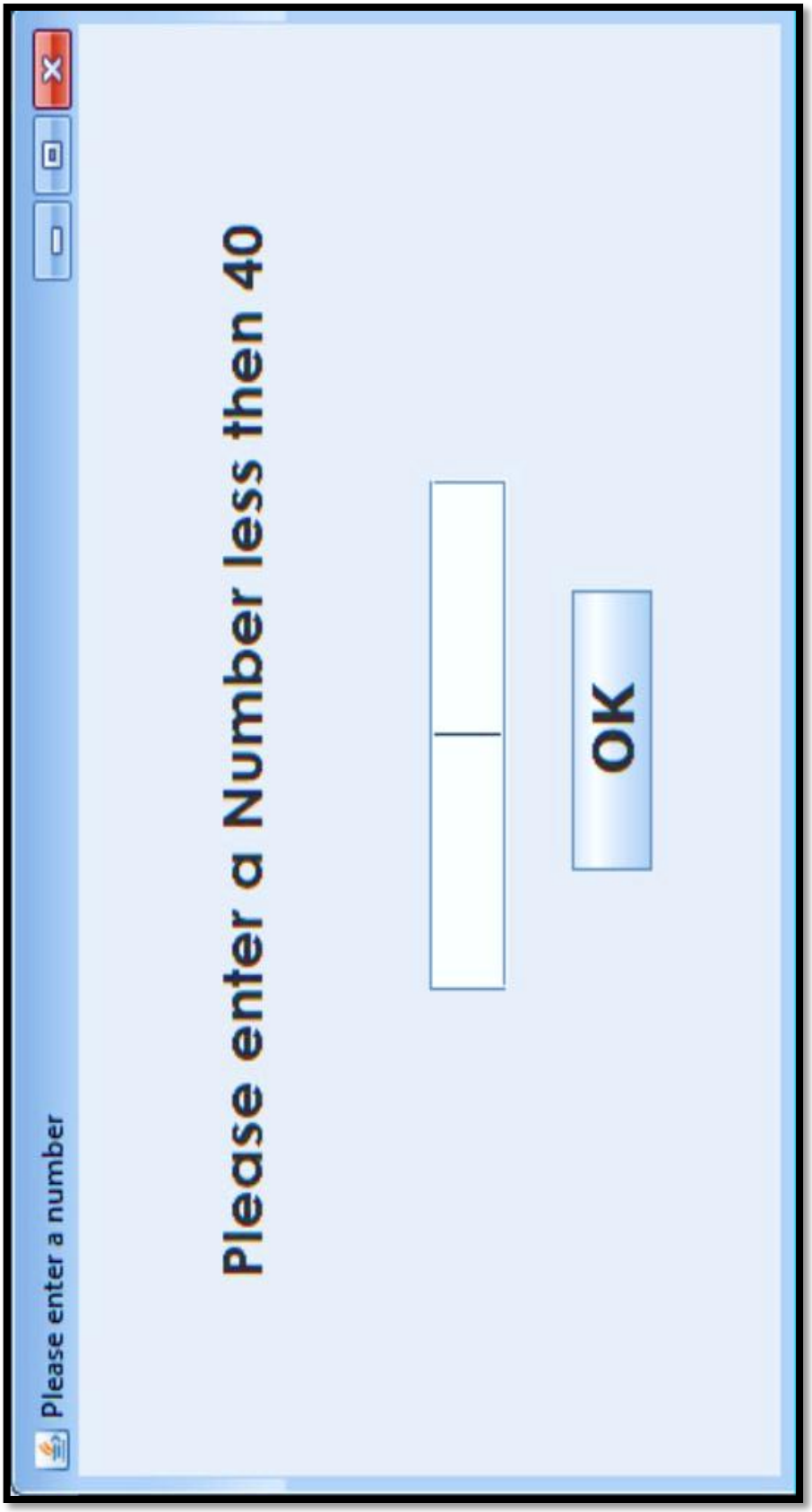
```
        else
        {
            JOptionPane.showMessageDialog(this, "Sorry! You have no more tries left..", "No tries left", JOptionPane.WARNING_MESSAGE);
        }
    }
  }
```

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    inputTF.setText("");
    inputlabel1.setText("");
    bullslabel1.setText("");
    cowslabel1.setText("");
    inputlabel2.setText("");
    bullslabel2.setText("");
    cowslabel2.setText("");
    inputlabel3.setText("");
    bullslabel3.setText("");
    cowslabel3.setText("");
    inputlabel4.setText("");
    bullslabel4.setText("");
    cowslabel4.setText("");
    inputlabel5.setText("");
    bullslabel5.setText("");
    cowslabel5.setText("");
    inputlabel6.setText("");
    bullslabel6.setText("");
    cowslabel6.setText("");
    inputlabel7.setText("");
```

```
        bullslabel7.setText("");
        cowslabel7.setText("");
        inputlabel8.setText("");
        bullslabel8.setText("");
        cowslabel8.setText("");
        inputlabel9.setText("");
        bullslabel9.setText("");
        cowslabel9.setText("");
        inputlabel10.setText("");
        bullslabel10.setText("");
        cowslabel10.setText("");
        trycount = 1;
        if (wordcount <= 40)
        {
            wordcount++;
        }
        else
        {
            wordcount = 1;
        }
    }
```

```
private void JumpHomemenu1ActionPerformed(java.awt.event.ActionEvent evt) {
        new How_to_Play().setVisible(true);
    }
```

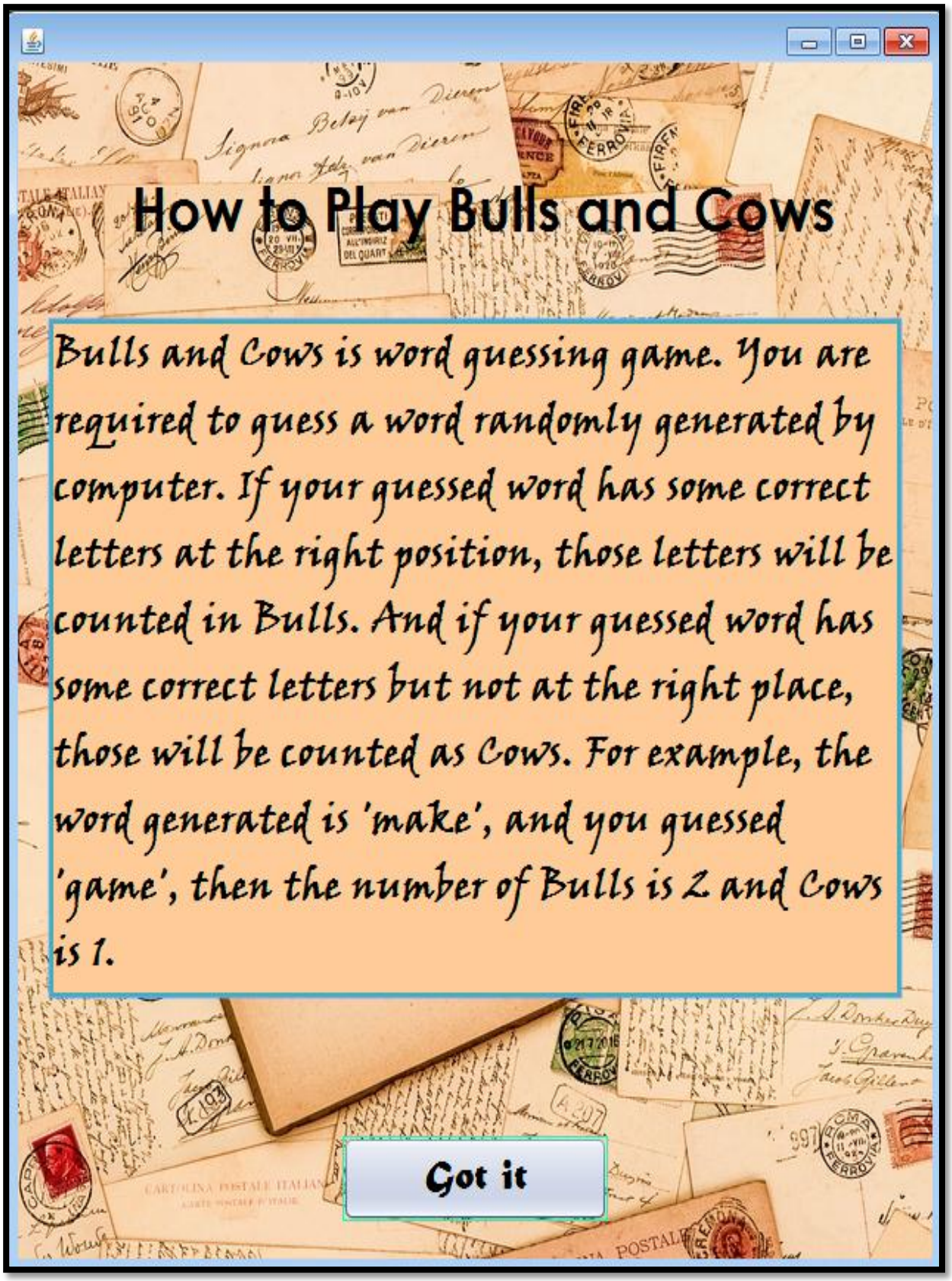
FRAME 6 (ENTER NO.)



CODING:

```
package legionnaire_os;

import javax.swing.JOptionPane;

/* @author Mandeep M. Dalavi */

public class Enter_no extends javax.swing.JFrame {

   int wordcount = 1;

   public Enter_no() {

      initComponents();

   }

-----------------------------------------------------------------------------------------------

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

     try {

        int num = Integer.parseInt(numTF.getText());

        if (num > 40)

        {

           JOptionPane.showMessageDialog(this, "Please enter a number less than 40", "Wrong
Input", JOptionPane.WARNING_MESSAGE);

           numTF.setText("");

        }

        else {

           wordcount = num;

           this.setVisible(false);

           numTF.setText("");

        }

     }

     catch (NumberFormatException e) {

        JOptionPane.showMessageDialog(this, "Your input is invalid.", "Invalid Input",
JOptionPane.ERROR_MESSAGE);

     }    }
```

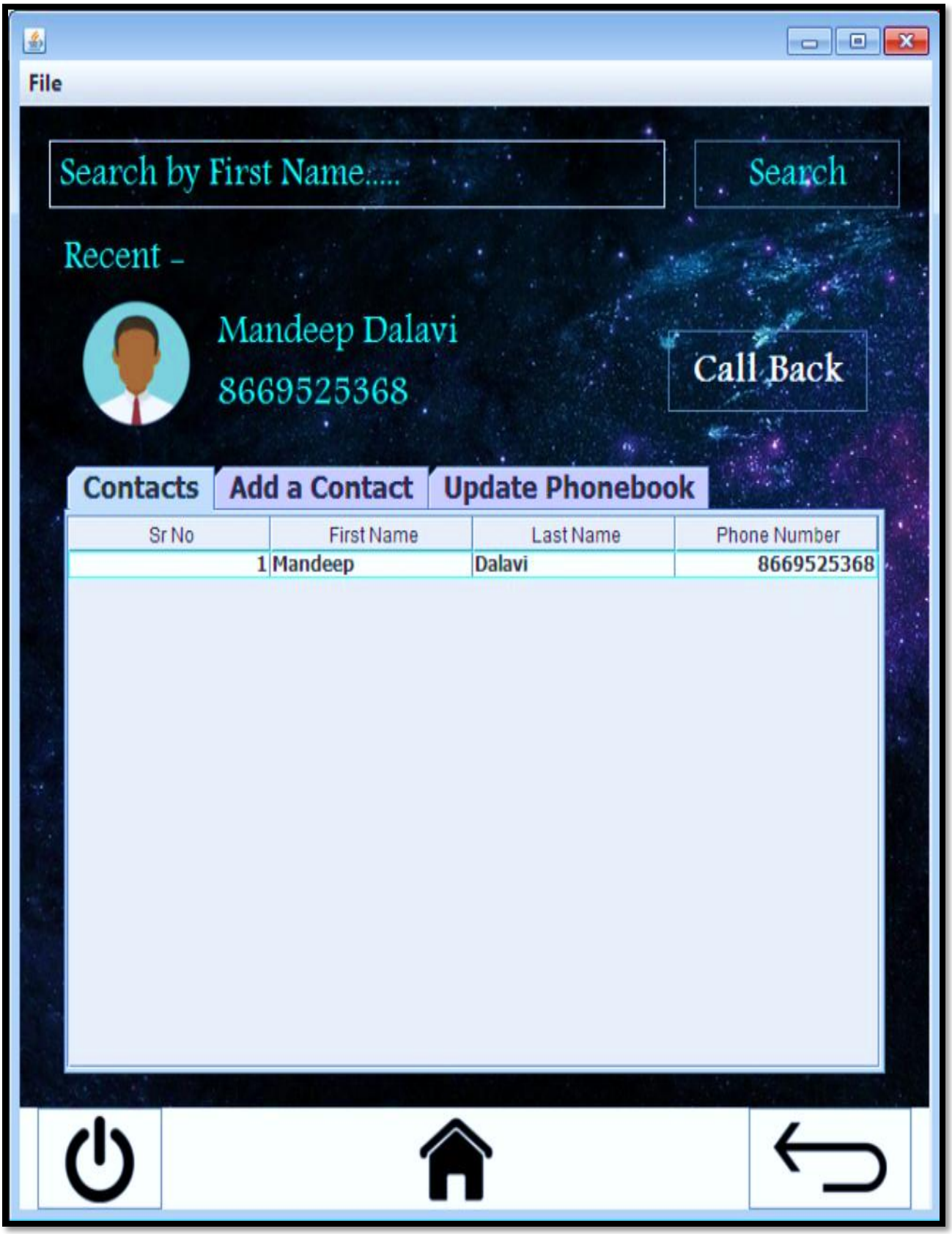
FRAME 7 (HOW TO PLAY)



CODING:

```
package legionnaire_os;
/* @author Mandeep M. Dalavi  */
public class How_to_Play extends javax.swing.JFrame {
   public How_to_Play() {
      initComponents();
   }
-----------------------------------------------------------------------------------------------------
private void formWindowActivated(java.awt.event.WindowEvent evt) {
      this.setResizable(true);
      this.setSize(636, 740);
   }
-----------------------------------------------------------------------------------------------------
private void GotitActionPerformed(java.awt.event.ActionEvent evt) {
      this.setVisible(false);
   }
-----------------------------------------------------------------------------------------------------
```

FRAME 8 (CONTACTS APP)



CODING:

```
package legionnaire_os;
/* @author Mandeep M. Dalavi */
import java.sql.*;
import java.math.BigInteger;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
public class Contacts_App extends javax.swing.JFrame {
   public Contacts_App() {
      initComponents();
//------------------------------Contacts display in table------------------------------//
   DefaultTableModel model = (DefaultTableModel)
   Contacttable.getModel();
   try {
         Class.forName("java.sql.DriverManager");
         Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire","root", "admin");
         Statement stmt = (Statement) con.createStatement();
         String query="SELECT * FROM contacts;";
         ResultSet rs = stmt.executeQuery(query);
         while(rs.next()) {
         String SrNo = rs.getString("Sr_No");
         String FirstName = rs.getString("First_Name");
         String LastName = rs.getString("Last_Name");
         String Number = rs.getString("Number");
         model.addRow (new Object[] {SrNo, FirstName, LastName,Number});
         }
      }
```

```
   catch (Exception e) {
        JOptionPane.showMessageDialog (this, "Error in Connectivity!");
     }
//-------------------------Name Display in Recent Called--------------------------//
   try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "SELECT * FROM calling";
        ResultSet rs = stmt.executeQuery(query);
        while (rs.next())
          {
             String Name = rs.getString(1);
             String Number = rs.getString(2);
             Contactname.setText(""+Name);
             Contactnumber.setText("" +Number);    }
        con.close();
        stmt.close();
        rs.close();
     }
   catch (Exception e) {
        JOptionPane.showMessageDialog(this, "Error in Connectivity!");
     }   }
-----------------------------------------------------------------------------------------------------
private void BackbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    new Apps().setVisible(true);
    this.setVisible(false);
   }
```

```
private void AddcontactbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    String first = firsttext.getText();
    String last = lasttext.getText();
    BigInteger num = new BigInteger(numbertext.getText());
  try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "Select max(Sr_No) from contacts";
        ResultSet rs1 = stmt.executeQuery(query);
        rs1.next();
        int srno = rs1.getInt(1) + 1;
  String query2 = "Insert into contacts values(" + srno + ",'" + first + "','" + last + "'," + num + ")";
        stmt.executeUpdate(query2);
        JOptionPane.showMessageDialog(this, "Your Contact has been added successfully",
"Contact Added", JOptionPane.INFORMATION_MESSAGE);
        firsttext.setText("");
        lasttext.setText("");
        numbertext.setText("");
      }
  catch (Exception e) {
        JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
      }
  }
```

```
private void formWindowActivated(java.awt.event.WindowEvent evt) {
    this.setResizable(true);
    this.setSize(638, 715);
```

```
//Setting Buttons Invisible
    first.setVisible(false);
    last.setVisible(false);
    number.setVisible(false);
    firstTF.setVisible(false);
    lastTF.setVisible(false);
    numberTF.setVisible(false);
    Deletebutton.setVisible(false);
    Updatebutton.setVisible(false);
    Resetbutton.setVisible(false);
  }
-----------------------------------------------------------------------------------------------------
private void PowerbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    if(JOptionPane.showConfirmDialog(this, "Are you sure you want to exit?", "Confirmation",
JOptionPane.YES_NO_CANCEL_OPTION)==JOptionPane.YES_OPTION)
        {
                System.exit(0);
        }
  }
-----------------------------------------------------------------------------------------------------
private void HomebuttonActionPerformed(java.awt.event.ActionEvent evt) {
     new Home().setVisible(true);
     this.setVisible(false);
  }
-----------------------------------------------------------------------------------------------------
private void JumpHomemenuActionPerformed(java.awt.event.ActionEvent evt) {
     new Home().setVisible(true);
     this.setVisible(false);
  }
```

```
private void JumpAppmenuActionPerformed(java.awt.event.ActionEvent evt) {
    new Apps().setVisible(true);
    this.setVisible(false);   }
-----------------------------------------------------------------------------------------------------
private void ExitOSmenuActionPerformed(java.awt.event.ActionEvent evt) {
    if(JOptionPane.showConfirmDialog(this, "Are you sure you want to exit?", "Confirmation",
JOptionPane.YES_NO_CANCEL_OPTION)==JOptionPane.YES_OPTION)
        {
            System.exit(0);
        }   }
-----------------------------------------------------------------------------------------------------
private void DeletebuttonActionPerformed(java.awt.event.ActionEvent evt) {
    int SrNo = Integer.parseInt(Srenter.getText());
    try {
        Class.forName("java.sql.DriverManager");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire","root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query="DELETE FROM Contacts where Sr_no = "+SrNo+";";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog (this, "Contact Deleted Successfully.");
        Srenter.setText("");
        firstTF.setText("");
        lastTF.setText("");
        numberTF.setText("");
        }
    catch(Exception e) {
        JOptionPane.showMessageDialog (this, "Error in Connectivity!");
        }   }
```

```
private void ShowdetailbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    String SrNo = Srenter.getText();
    if(SrNo.isEmpty())
      {
          JOptionPane.showMessageDialog(this, "Please enter Serial Number");
      }
    else {
      try {
          Class.forName("java.sql.Driver");
          Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
          Statement stmt = (Statement) con.createStatement();
          String query = "SELECT * FROM contacts where Sr_No = "+SrNo+";";
          ResultSet rs = stmt.executeQuery(query);
          while (rs.next()) {
              String firstname = rs.getString(2);
              String lastname = rs.getString(3);
              String number = rs.getString(4);
              firstTF.setText(""+firstname);
              lastTF.setText("" +lastname);
              numberTF.setText(""+number);
           }
        con.close();
        stmt.close();
        rs.close();
      }
   catch (Exception e) {
        JOptionPane.showMessageDialog (this, "Error in Connectivity!");
      }    }
```

```
    first.setVisible(true);
    last.setVisible(true);
    number.setVisible(true);
    firstTF.setVisible(true);
    lastTF.setVisible(true);
    numberTF.setVisible(true);
    Deletebutton.setVisible(true);
    Updatebutton.setVisible(true);
    Resetbutton.setVisible(true);
  }
-----------------------------------------------------------------------------------------------------
private void ResetbuttonActionPerformed(java.awt.event.ActionEvent evt) {
        Srenter.setText("");
        firstTF.setText("");
        lastTF.setText("");
        numberTF.setText("");
        first.setVisible(false);
        last.setVisible(false);
        number.setVisible(false);
        firstTF.setVisible(false);
        lastTF.setVisible(false);
        numberTF.setVisible(false);
        Deletebutton.setVisible(false);
        Updatebutton.setVisible(false);
        Resetbutton.setVisible(false);
    }
```

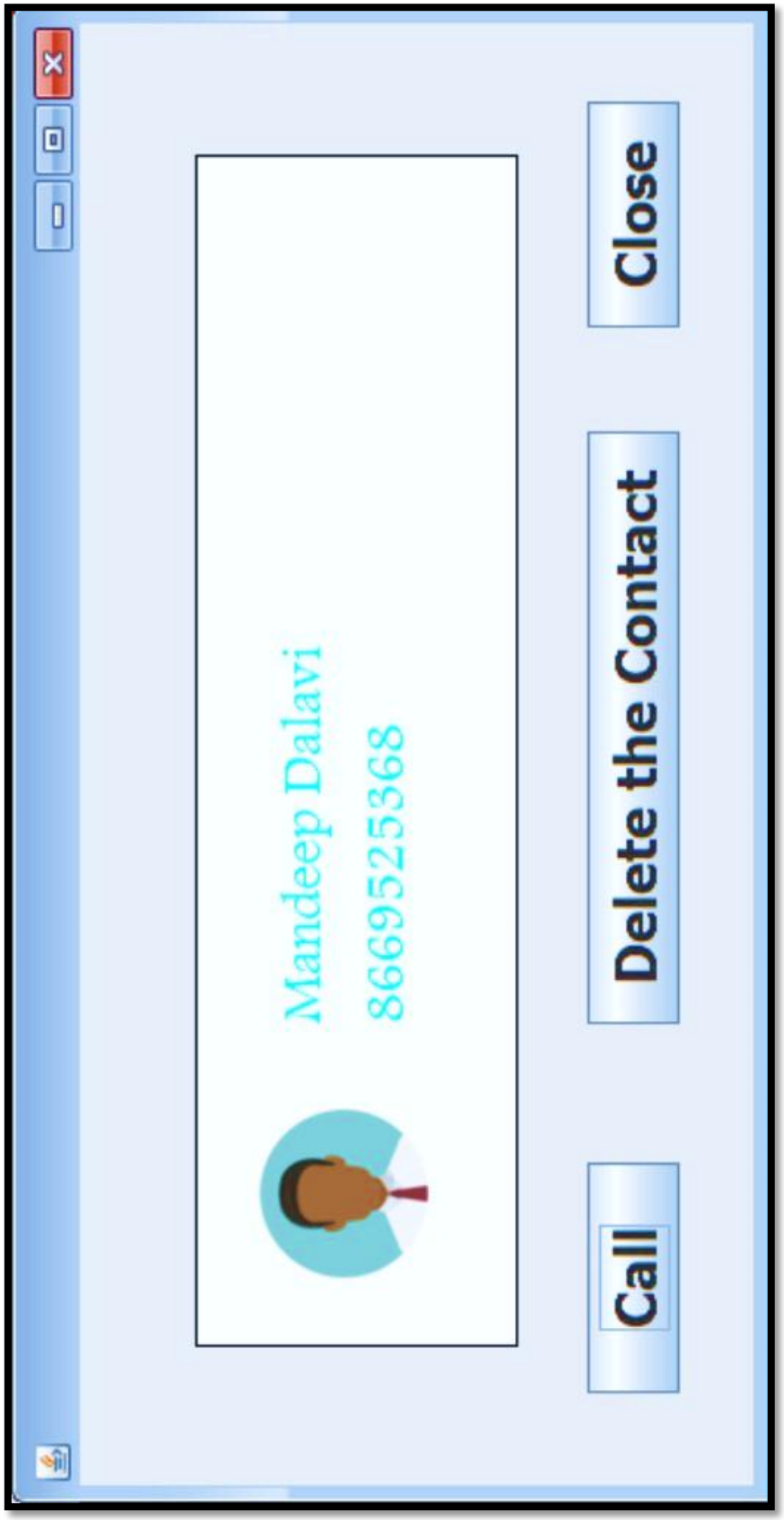
```
private void UpdatebuttonActionPerformed(java.awt.event.ActionEvent evt) {
      String SrNo = Srenter.getText();
      String First = firstTF.getText();
      String Last = lastTF.getText();
      String Number = numberTF.getText();
      try {
         Class.forName("java.sql.Driver");
         Connection con = (Connection)
      DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
         Statement stmt = (Statement) con.createStatement();
         String query = ("UPDATE Contacts SET First_Name ='"+First+"', Last_Name ='"+Last+"',
Number = "+Number+" where Sr_No = "+SrNo+" ;");
         ResultSet rs = stmt.executeQuery(query);
         con.close();
         stmt.close();
         rs.close();      }
     catch (Exception e) {
         JOptionPane.showMessageDialog (this, "Error in Connectivity!");
   }
//-----------------------------------------------------------------------------------------------//
DefaultTableModel model = (DefaultTableModel)
Contacttable.getModel(); int rows=model.getRowCount();
if (rows>0)
   {
      for (int i=0; i<rows; i++)
      model.removeRow(0);
   }
try {
      Class.forName("java.sql.DriverManager");
```

```
    Connection con = (Connection)
    DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire","root", "admin");
    Statement stmt = (Statement) con.createStatement();
    String query="SELECT * FROM Contacts;";
    ResultSet rs = stmt.executeQuery(query);
    while(rs.next())
    {
       String Name = rs.getString("Name");
       String Mobile = rs.getString("Mobile");
       String Email = rs.getString("Email");
       model.addRow (new Object[] {Name, Mobile,Email});
    }
  }
catch (Exception e) {
    JOptionPane.showMessageDialog (this, "Error in Connectivity!");
  }   }
-----------------------------------------------------------------------------------------------------
private void SearchbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    String Search = Searchtext.getText();
    try {
       Class.forName("java.sql.DriverManager");
       Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire","root", "admin");
       Statement stmt = (Statement) con.createStatement();
       String query = "INSERT INTO Search VALUES('"+Search+"');";
       stmt.executeUpdate(query);
       new Contact_Details().setVisible(true);
       this.setVisible(false);
       }
```

```
    catch(Exception e) {
        JOptionPane.showMessageDialog (this, “Error in Connectivity!”);
        }
  }
-----------------------------------------------------------------------------------------------------
private void SearchtextMouseClicked(java.awt.event.MouseEvent evt) {
    Searchtext.setText("");
  }
-----------------------------------------------------------------------------------------------------
private void CallbackbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    JOptionPane.showMessageDialog(this, "Sorry You have no SIM card in phone. Please insert a SIM to call");
  }
-----------------------------------------------------------------------------------------------------
private void TabsMouseClicked(java.awt.event.MouseEvent evt) {
    DefaultTableModel model = (DefaultTableModel)
    Contacttable.getModel(); int rows=model.getRowCount();
if (rows>0)
  {
    for (int i=0; i<rows; i++)
    model.removeRow(0);
  }
//-----------------------------------------------------------------------------------------------------//
Contacttable.getModel();
try {
  Class.forName("java.sql.DriverManager");
  Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire","root", "admin");
  Statement stmt = (Statement) con.createStatement();
```

```
   String query="SELECT * FROM contacts;";
   ResultSet rs = stmt.executeQuery(query);
   while(rs.next())
   {
      String SrNo = rs.getString("Sr_No");
      String FirstName = rs.getString("First_Name");
      String LastName = rs.getString("Last_Name");
      String Number = rs.getString("Number");
      model.addRow (new Object[] {SrNo, FirstName, LastName,Number});
   }
}
catch (Exception e) {
   JOptionPane.showMessageDialog (this, "Error in Connectivity!");
}
}
```

FRAME 9 (CONTACTS DETAILS)



CODING:

```
package legionnaire_os;
import java.sql.*;;
import java.math.BigInteger;
import javax.swing.JOptionPane;
/* @author Mandeep M. Dalavi */
public class Contact_Details extends javax.swing.JFrame {
   String firstname, lastname, number;
   public Contact_Details() {
      initComponents();
   }
-----------------------------------------------------------------------------------------------------
private void CloseActionPerformed(java.awt.event.ActionEvent evt) {
      new Contacts_App().setVisible(true);
      this.setVisible(false);
      try {
         Class.forName("java.sql.DriverManager");
         Connection con = (Connection)
      DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire","root", "admin");
         Statement stmt = (Statement) con.createStatement();
         String query="DELETE from Search;";
         stmt.executeUpdate(query);
         }
      catch(Exception e){
      JOptionPane.showMessageDialog (this, "Error in Connectivity!");
   }
}
```

```
private void formWindowOpened(java.awt.event.WindowEvent evt) {
      this.setSize(650, 330);
      try {
         Class.forName("java.sql.Driver");
         Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
         Statement stmt = (Statement) con.createStatement();
         String query = "SELECT First_Name, Last_Name, Number FROM Contacts C, Search S
where C.First_Name = S.Name;";
         ResultSet rs = stmt.executeQuery(query);
         while (rs.next())
         {
            firstname = rs.getString(1);
            lastname = rs.getString(2);
            number = rs.getString(3);
            Contactname.setText(""+firstname+" "+lastname);
            Contactnumber.setText(""+number);
         }
         con.close();
         stmt.close();
         rs.close();
    }
    catch (Exception e) {
      JOptionPane.showMessageDialog (this, "Error in Connectivity!");
   }
 }
```

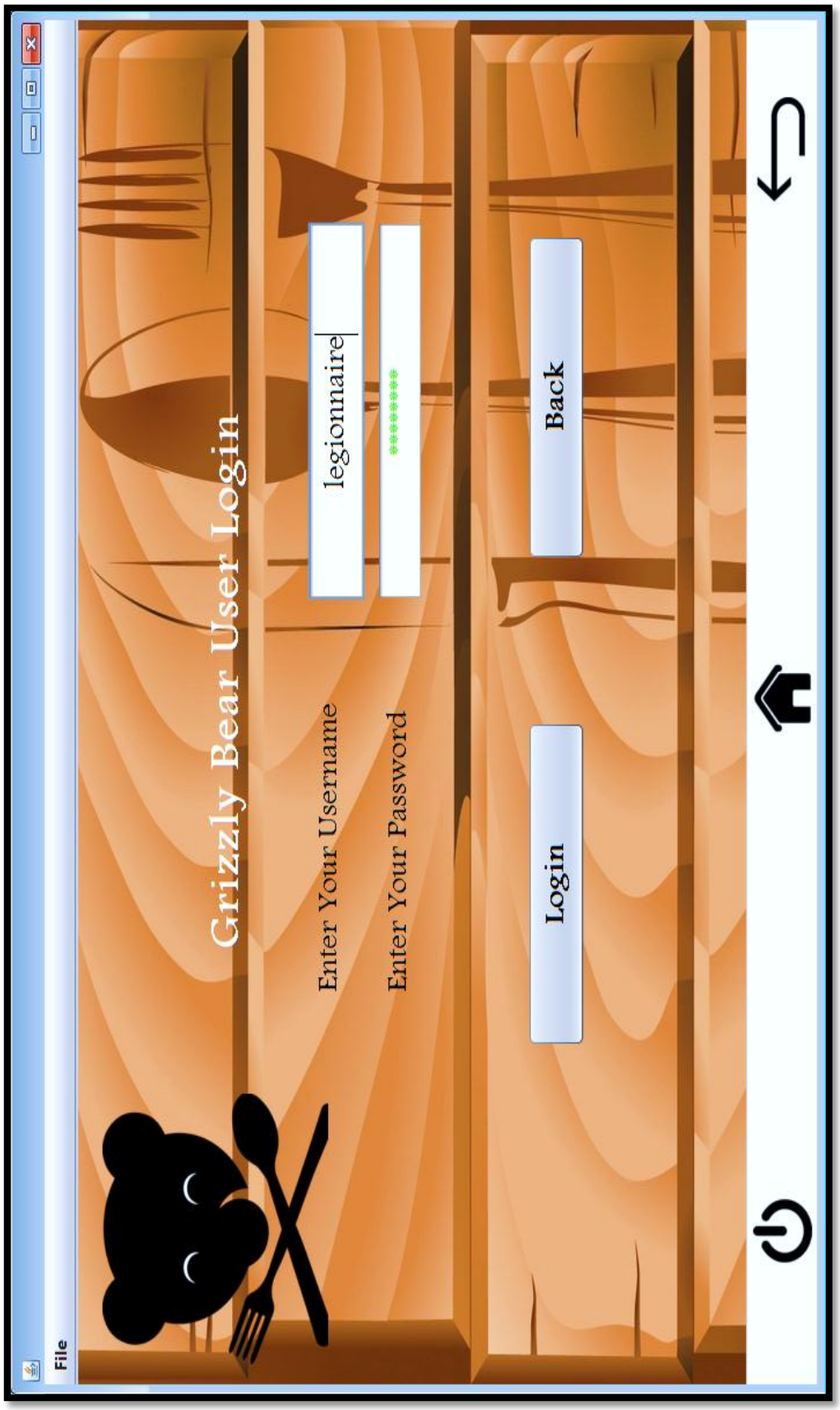
```
private void DeleteActionPerformed(java.awt.event.ActionEvent evt) {
     number = Contactnumber.getText();
```

```
    try {
        Class.forName("java.sql.DriverManager");
        Connection con = (Connection)
     DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire","root", "admin”);
        Statement stmt = (Statement) con.createStatement();
        String query="DELETE from Contacts where (First_Name = '"+firstname+"' && Number =
"+number+");";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog (this, "Contact Deleted Successfully.");
        new Contacts_App().setVisible(true);
        this.setVisible(false);
        }
    catch(Exception e) {
        JOptionPane.showMessageDialog (this, “Error in Connectivity!”);
        }
  }
```

```
private void CallActionPerformed(java.awt.event.ActionEvent evt) {
     String name = Contactname.getText();
     BigInteger num = new BigInteger(Contactnumber.getText());
     try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query1 = "DELETE from calling";
        int rs1 = stmt.executeUpdate(query1);
        String query = "INSERT into calling VALUES('" + name + "'," + num + ")";
        int rs = stmt.executeUpdate(query);
```

```
        con.close();
        stmt.close();
    }
  catch (Exception e) {
        JOptionPane.showMessageDialog (this, "Error in connectivity!");
    }
  JOptionPane.showMessageDialog(null, "Sorry you have no SIM Card in phone. Please insert a
SIM to call", "No SIM Card Detected", JOptionPane.ERROR_MESSAGE);
 }
```

FRAME 10 (GRIZZLY BEAR LOGIN)



CODING:

```
package legionnaire_os;

import java.sql.*;

import javax.swing.JOptionPane;

/* @author Mandeep M. Dalavi */

public class GrizzlyBear_Login extends javax.swing.JFrame {

String curusername;

String curpassword;

   public GrizzlyBear_Login() {

      initComponents();

//------------------------JDBC part getting username and password----------------------//

try {

      Class.forName("java.sql.Driver");

      Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");

      Statement stmt = (Statement) con.createStatement();

      String query = "Select * from userpass";

      ResultSet rs = stmt.executeQuery(query);

      if (rs.first())

        {

          curusername = rs.getString(1);

          curpassword = rs.getString(2);

        }

   }

catch (Exception e) {

      JOptionPane.showMessageDialog (this, “Error in Connectivity”);

   }
```

```
private void Backbutton1ActionPerformed(java.awt.event.ActionEvent evt) {
      new Apps().setVisible(true);
      this.setVisible(false);
   }
-----------------------------------------------------------------------------------------------------
private void JumpHomemenuActionPerformed(java.awt.event.ActionEvent evt) {
      new Home().setVisible(true);
      this.setVisible(false);
   }
-----------------------------------------------------------------------------------------------------
private void JumpAppmenuActionPerformed(java.awt.event.ActionEvent evt) {
      new Apps().setVisible(true);
      this.setVisible(false);
   }
-----------------------------------------------------------------------------------------------------
private void ExitOSmenuActionPerformed(java.awt.event.ActionEvent evt) {
      if(JOptionPane.showConfirmDialog(this, "Are you sure you want to exit?", "Confirmation",
JOptionPane.YES_NO_CANCEL_OPTION)==JOptionPane.YES_OPTION)
         {
                  System.exit(0);
         }
   }
-----------------------------------------------------------------------------------------------------
private void formWindowActivated(java.awt.event.WindowEvent evt) {
      this.setResizable(true);
      this.setSize(1025, 706);
   }
```

```
private void PowerbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    if(JOptionPane.showConfirmDialog(this, "Are you sure you want to exit?", "Confirmation",
JOptionPane.YES_NO_CANCEL_OPTION)==JOptionPane.YES_OPTION)
        {
            System.exit(0);
        }
  }
-----------------------------------------------------------------------------------------------------
private void HomebuttonActionPerformed(java.awt.event.ActionEvent evt) {
    new Home().setVisible(true);
    this.setVisible(false);
  }
-----------------------------------------------------------------------------------------------------
private void BackbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    new Apps().setVisible(true);
    this.setVisible(false);
  }
-----------------------------------------------------------------------------------------------------
private void LoginbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    String user = Usernametext.getText();
    String pass = new String(Password.getPassword());
    if (user.equalsIgnoreCase(curusername) && pass.equalsIgnoreCase(curpassword))
        {
            JOptionPane.showMessageDialog(this,"You have been successfully logged in." );
            new GrizzlyBear_Menu().setVisible(true);
            this.dispose();
        }
```

```
    else
        {
            JOptionPane.showMessageDialog(this, "Password and User ID do not match. Please try again.", "Incorrect username or password", JOptionPane.ERROR_MESSAGE);
        }
  }
-----------------------------------------------------------------------------------------------------------
private void PasswordMouseClicked(java.awt.event.MouseEvent evt) {
    Password.setText("");
  }
-----------------------------------------------------------------------------------------------------------
```

FRAME 11 (GRIZZLY BEAR MENU)

CODING:

```
package legionnaire_os;

import java.sql.*;

import javax.swing.JOptionPane;

/* @author Mandeep M. Dalavi */

public class GrizzlyBear_Menu extends javax.swing.JFrame {

   public GrizzlyBear_Menu() {

      initComponents();   }

----------------------------------------------------------------------------------------------------

private void BackbuttonActionPerformed(java.awt.event.ActionEvent evt) {

     new GrizzlyBear_Login().setVisible(true);

     this.setVisible(false);

   }

----------------------------------------------------------------------------------------------------

private void JumpHomemenuActionPerformed(java.awt.event.ActionEvent evt) {

     new Home().setVisible(true);

     this.setVisible(false);

   }

----------------------------------------------------------------------------------------------------

private void JumpAppmenuActionPerformed(java.awt.event.ActionEvent evt) {

     new Apps().setVisible(true);

     this.setVisible(false);   }

----------------------------------------------------------------------------------------------------

private void ExitOSmenuActionPerformed(java.awt.event.ActionEvent evt) {

     if(JOptionPane.showConfirmDialog(this, "Are you sure you want to exit?", "Confirmation",
JOptionPane.YES_NO_CANCEL_OPTION)==JOptionPane.YES_OPTION)

         {         System.exit(0);         }

   }
```

```
private void formWindowActivated(java.awt.event.WindowEvent evt) {
    this.setResizable(true);
    this.setSize(1025, 705);
  }
-----------------------------------------------------------------------------------------------------
private void PowerbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    if(JOptionPane.showConfirmDialog(this, "Are you sure you want to exit?", "Confirmation",
JOptionPane.YES_NO_CANCEL_OPTION)==JOptionPane.YES_OPTION)
        {
            System.exit(0);
        }
  }
-----------------------------------------------------------------------------------------------------
private void HomebuttonActionPerformed(java.awt.event.ActionEvent evt) {
    new Home().setVisible(true);
    this.setVisible(false);
  }
-----------------------------------------------------------------------------------------------------
private void ShowBillbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    new GrizzlyBear_Bill().setVisible(true);
    this.setVisible(false);
  }
-----------------------------------------------------------------------------------------------------
private void AddBillbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    String Food = FoodName.getText();
    int price = Integer.parseInt(Price.getText());
    int quantity = (int) Quantity.getValue();
    int total = Integer.parseInt(Amount.getText());
```

```
    try {
        Class.forName("java.sql.DriverManager");
        Connection con = (Connection)
     DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire","root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query="INSERT INTO BILL VALUES('"+Food+"', "+price+", "+quantity+", "+total+");";
        stmt.executeUpdate(query);
        }
    catch(Exception e) {
       JOptionPane.showMessageDialog(this, "Error in Connectivity!");
        }
  }
```

```
private void DrinksItemStateChanged(java.awt.event.ItemEvent evt) {
    int food = Drinks.getSelectedIndex();
    switch (food) {
        case 1:
          FoodName.setText("");
          FoodType.setText("");
          Price.setText("");
          FoodName.setText("Tea");
          FoodType.setText("Drinks");
          Price.setText("" + 20);
          break;
        case 2:
          FoodName.setText("");
          FoodType.setText("");
          Price.setText("");
```

```
        FoodName.setText("Coffee");
        FoodType.setText("Drinks");
        Price.setText("" + 25);
        break;
    case 3:
        FoodName.setText("");
        FoodType.setText("");
        Price.setText("");
        FoodName.setText("Soda");
        FoodType.setText("Drinks");
        Price.setText("" + 15);
        break;
    case 4:
        FoodName.setText("");
        FoodType.setText("");
        Price.setText("");
        FoodName.setText("Soft Drink");
        FoodType.setText("Drinks");
        Price.setText("" + 35);
        break;
    case 5:
        FoodName.setText("");
        FoodType.setText("");
        Price.setText("");
        FoodName.setText("Lime Soda");
        FoodType.setText("Drinks");
        Price.setText("" + 25);
        break;
```

```
case 6:
   FoodName.setText("");
   FoodType.setText("");
   Price.setText("");
   FoodName.setText("Milk Shake");
   FoodType.setText("Drinks");
   Price.setText("" + 40);
   break;
case 7:
   FoodName.setText("");
   FoodType.setText("");
   Price.setText("");
   FoodName.setText("Vanilla Shake");
   FoodType.setText("Drinks");
   Price.setText("" + 60);
   break;
case 8:
   FoodName.setText("");
   FoodType.setText("");
   Price.setText("");
   FoodName.setText("Chocolate Shake");
   FoodType.setText("Drinks");
   Price.setText("" + 80);
   break;
case 9:
   FoodName.setText("");
   FoodType.setText("");
   Price.setText("");
```

```
            FoodName.setText("Strawberry Shake");
            FoodType.setText("Drinks");
            Price.setText("" + 60);
            break;
        default:
            break;
    }
    Quantity.setValue(0);
    Amount.setText("");
  }
```

```
private void SouthIndianItemStateChanged(java.awt.event.ItemEvent evt) {
    int food = SouthIndian.getSelectedIndex();
    switch (food) {
        case 1:
            FoodName.setText("");
            FoodType.setText("");
            Price.setText("");
            FoodName.setText("Sada Dosa");
            FoodType.setText("South Indian");
            Price.setText("" + 50);
            break;
        case 2:
            FoodName.setText("");
            FoodType.setText("");
            Price.setText("");
            FoodName.setText("Masala Dosa");
            FoodType.setText("South Indian");
```

```
    Price.setText("" + 60);
    break;
  case 3:
    FoodName.setText("");
    FoodType.setText("");
    Price.setText("");
    FoodName.setText("Uttapam");
    FoodType.setText("South Indian");
    Price.setText("" + 55);
    break;
  case 4:
    FoodName.setText("");
    FoodType.setText("");
    Price.setText("");
    FoodName.setText("Rava Dosa");
    FoodType.setText("South Indian");
    Price.setText("" + 55);
    break;
  case 5:
    FoodName.setText("");
    FoodType.setText("");
    Price.setText("");
    FoodName.setText("Rava Masala");
    FoodType.setText("South Indian");
    Price.setText("" + 65);
    break;
  case 6:
    FoodName.setText("");
```

```
            FoodType.setText("");
            Price.setText("");
            FoodName.setText("Spring Masala");
            FoodType.setText("South Indian");
            Price.setText("" + 75);
            break;
        default:
            break;
    }
    Quantity.setValue(0);
    Amount.setText("");
  }
```

```
private void SandwichItemStateChanged(java.awt.event.ItemEvent evt) {
    int food = Sandwich.getSelectedIndex();
    switch (food) {
        case 1:
            FoodName.setText("");
            FoodType.setText("");
            Price.setText("");
            FoodName.setText("Veg Sandwich");
            FoodType.setText("Sandwich");
            Price.setText("" + 40);
            break;
        case 2:
            FoodName.setText("");
            FoodType.setText("");
            Price.setText("");
```

```
    FoodName.setText("Chatni Sandwich");
    FoodType.setText("Sandwich");
    Price.setText("" + 30);
    break;
  case 3:
    FoodName.setText("");
    FoodType.setText("");
    Price.setText("");
    FoodName.setText("Omlet Sandwich");
    FoodType.setText("Sandwich");
    Price.setText("" + 55);
    break;
  case 4:
    FoodName.setText("");
    FoodType.setText("");
    Price.setText("");
    FoodName.setText("Cheese Sandwich");
    FoodType.setText("Sandwich");
    Price.setText("" + 70);
    break;
  case 5:
    FoodName.setText("");
    FoodType.setText("");
    Price.setText("");
    FoodName.setText("Club Sandwich");
    FoodType.setText("Sandwich");
    Price.setText("" + 80);
    break;
```

```
        case 6:
           FoodName.setText("");
           FoodType.setText("");
           Price.setText("");
           FoodName.setText("Veg Grill Sandwich");
           FoodType.setText("Sandwich");
           Price.setText("" + 65);
           break;
        case 7:
           FoodName.setText("");
           FoodType.setText("");
           Price.setText("");
           FoodName.setText("Diamond Club Sandwich");
           FoodType.setText("Sandwich");
           Price.setText("" + 100);
           break;
        default:
           break;
     }
    Quantity.setValue(0);
    Amount.setText("");
  }
```

```
private void ShakahariKhazanaItemStateChanged(java.awt.event.ItemEvent evt) {
     int food = ShakahariKhazana.getSelectedIndex();
      switch (food) {
        case 1:
           FoodName.setText("");
```

```
    FoodType.setText("");
    Price.setText("");
    FoodName.setText("Veg Handi");
    FoodType.setText("Shakahari Khazana");
    Price.setText("" + 145);
    break;
case 2:
    FoodName.setText("");
    FoodType.setText("");
    Price.setText("");
    FoodName.setText("Mix Vegetables");
    FoodType.setText("Shakahari Khazana");
    Price.setText("" + 120);
    break;
case 3:
    FoodName.setText("");
    FoodType.setText("");
    Price.setText("");
    FoodName.setText("Dum Allu");
    FoodType.setText("Shakahari Khazana");
    Price.setText("" + 135);
    break;
case 4:
    FoodName.setText("");
    FoodType.setText("");
    Price.setText("");
    FoodName.setText("Allu Mutter");
    FoodType.setText("Shakahari Khazana");
```

```
    Price.setText("" + 105);
    break;
  case 5:
    FoodName.setText("");
    FoodType.setText("");
    Price.setText("");
    FoodName.setText("Chhole Bhature");
    FoodType.setText("Shakahari Khazana");
    Price.setText("" + 140);
    break;
  case 6:
    FoodName.setText("");
    FoodType.setText("");
    Price.setText("");
    FoodName.setText("Mutter Paneer");
    FoodType.setText("Shakahari Khazana");
    Price.setText("" + 145);
    break;
  case 7:
    FoodName.setText("");
    FoodType.setText("");
    Price.setText("");
    FoodName.setText("Paneer Tikka");
    FoodType.setText("Shakahari Khazana");
    Price.setText("" + 160);
    break;
  case 8:
    FoodName.setText("");
```

```
            FoodType.setText("");
            Price.setText("");
            FoodName.setText("Kaju Masala");
            FoodType.setText("Shakahari Khazana");
            Price.setText("" + 185);
            break;
        case 9:
            FoodName.setText("");
            FoodType.setText("");
            Price.setText("");
            FoodName.setText("Mushroom Fry");
            FoodType.setText("Shakahari Khazana");
            Price.setText("" + 145);
            break;
        default:
            break;
    }
    Quantity.setValue(0);
    Amount.setText("");
  }
-----------------------------------------------------------------------------------------------
private void ChapatiItemStateChanged(java.awt.event.ItemEvent evt) {
    int food = Chapati.getSelectedIndex();
    switch (food) {
        case 1:
            FoodName.setText("");
            FoodType.setText("");
            Price.setText("");
```

```
    FoodName.setText("Chapati");
    FoodType.setText("Chapati");
    Price.setText("" + 8);
    break;
case 2:
    FoodName.setText("");
    FoodType.setText("");
    Price.setText("");
    FoodName.setText("Roti");
    FoodType.setText("Chapati");
    Price.setText("" + 15);
    break;
case 3:
    FoodName.setText("");
    FoodType.setText("");
    Price.setText("");
    FoodName.setText("Paratha");
    FoodType.setText("Chapati");
    Price.setText("" + 24);
    break;
case 4:
    FoodName.setText("");
    FoodType.setText("");
    Price.setText("");
    FoodName.setText("Naan");
    FoodType.setText("Chapati");
    Price.setText("" + 25);
    break;
```

```
        case 5:
            FoodName.setText("");
            FoodType.setText("");
            Price.setText("");
            FoodName.setText("Garlic Naan");
            FoodType.setText("Chapati");
            Price.setText("" + 50);
            break;
        case 6:
            FoodName.setText("");
            FoodType.setText("");
            Price.setText("");
            FoodName.setText("Kashmiri Naan");
            FoodType.setText("Chapati");
            Price.setText("" + 50);
            break;
        default:
            break;
    }
    Quantity.setValue(0);
    Amount.setText("");
  }
```

```
private void SoupItemStateChanged(java.awt.event.ItemEvent evt) {
    int food = Soup.getSelectedIndex();
    switch (food) {
        case 1:
            FoodName.setText("");
```

```
    FoodType.setText("");
    Price.setText("");
    FoodName.setText("Tomato Soup");
    FoodType.setText("Soup");
    Price.setText("" + 85);
    break;
  case 2:
    FoodName.setText("");
    FoodType.setText("");
    Price.setText("");
    FoodName.setText("Veg Manchow");
    FoodType.setText("Soup");
    Price.setText("" + 95);
    break;
  case 3:
    FoodName.setText("");
    FoodType.setText("");
    Price.setText("");
    FoodName.setText("Veg Noddles");
    FoodType.setText("Soup");
    Price.setText("" + 90);
    break;
  case 4:
    FoodName.setText("");
    FoodType.setText("");
    Price.setText("");
    FoodName.setText("Palak Soup");
    FoodType.setText("Soup");
```

```
            Price.setText("" + 90);
            break;
        case 5:
            FoodName.setText("");
            FoodType.setText("");
            Price.setText("");
            FoodName.setText("Sweet Corn Noddles");
            FoodType.setText("Soup");
            Price.setText("" + 90);
            break;
        default:
            break;
    }
    Quantity.setValue(0);
    Amount.setText("");
  }
```

```
private void RiceItemStateChanged(java.awt.event.ItemEvent evt) {
    int food = Rice.getSelectedIndex();
    switch (food) {
        case 1:
            FoodName.setText("");
            FoodType.setText("");
            Price.setText("");
            FoodName.setText("Plain Rice");
            FoodType.setText("Rice");
            Price.setText("" + 90);
            break;
```

```
case 2:
   FoodName.setText("");
   FoodType.setText("");
   Price.setText("");
   FoodName.setText("Jeera Rice");
   FoodType.setText("Rice");
   Price.setText("" + 100);
   break;
case 3:
   FoodName.setText("");
   FoodType.setText("");
   Price.setText("");
   FoodName.setText("Lemon Rice");
   FoodType.setText("Rice");
   Price.setText("" + 100);
   break;
case 4:
   FoodName.setText("");
   FoodType.setText("");
   Price.setText("");
   FoodName.setText("Veg Fried Rice");
   FoodType.setText("Rice");
   Price.setText("" + 120);
   break;
case 5:
   FoodName.setText("");
   FoodType.setText("");
   Price.setText("");
```

```
            FoodName.setText("Jeera Fried Rice");
            FoodType.setText("Rice");
            Price.setText("" + 120);
            break;
         default:
            break;
      }
    Quantity.setValue(0);
    Amount.setText("");
  }
```

```
private void DalItemStateChanged(java.awt.event.ItemEvent evt) {
      int food = Dal.getSelectedIndex();
      switch (food) {
         case 1:
            FoodName.setText("");
            FoodType.setText("");
            Price.setText("");
            FoodName.setText("Dal Fry");
            FoodType.setText("Dal");
            Price.setText("" + 95);
            break;
         case 2:
            FoodName.setText("");
            FoodType.setText("");
            Price.setText("");
            FoodName.setText("Dal Tadka");
            FoodType.setText("Dal");
```

```
    Price.setText("" + 110);
    break;
  case 3:
    FoodName.setText("");
    FoodType.setText("");
    Price.setText("");
    FoodName.setText("Dal Makhani");
    FoodType.setText("Dal");
    Price.setText("" + 110);
    break;
  case 4:
    FoodName.setText("");
    FoodType.setText("");
    Price.setText("");
    FoodName.setText("Dal Palak");
    FoodType.setText("Dal");
    Price.setText("" + 120);
    break;
  case 5:
    FoodName.setText("");
    FoodType.setText("");
    Price.setText("");
    FoodName.setText("Dal Moghlai");
    FoodType.setText("Dal");
    Price.setText("" + 120);
    break;
  default:
    break;
```

```
    }
    Quantity.setValue(0);
    Amount.setText("");
  }
---------------------------------------------------------------------------------------------------
private void WesternItemStateChanged(java.awt.event.ItemEvent evt) {
      int food = Western.getSelectedIndex();
      switch (food) {
         case 1:
            FoodName.setText("");
            FoodType.setText("");
            Price.setText("");
            FoodName.setText("Pandi with Pita");
            FoodType.setText("Western");
            Price.setText("" + 350);
            break;
         case 2:
            FoodName.setText("");
            FoodType.setText("");
            Price.setText("");
            FoodName.setText("Naga Pork Curry");
            FoodType.setText("Western");
            Price.setText("" + 390);
            break;
         case 3:
            FoodName.setText("");
            FoodType.setText("");
            Price.setText("");
```

```
        FoodName.setText("Mushroom Crepes");
        FoodType.setText("Western");
        Price.setText("" + 350);
        break;
    case 4:
        FoodName.setText("");
        FoodType.setText("");
        Price.setText("");
        FoodName.setText("Champagne Risotto");
        FoodType.setText("Western");
        Price.setText("" + 360);
        break;
    case 5:
        FoodName.setText("");
        FoodType.setText("");
        Price.setText("");
        FoodName.setText("Peri-Peri Grilled Chicken");
        FoodType.setText("Western");
        Price.setText("" + 390);
        break;
    case 6:
        FoodName.setText("");
        FoodType.setText("");
        Price.setText("");
        FoodName.setText("Lamb Chops with Beer Sauce");
        FoodType.setText("Western");
        Price.setText("" + 500);
        break;
```

```
        default:
            break;
      }
    Quantity.setValue(0);
    Amount.setText("");
  }
```

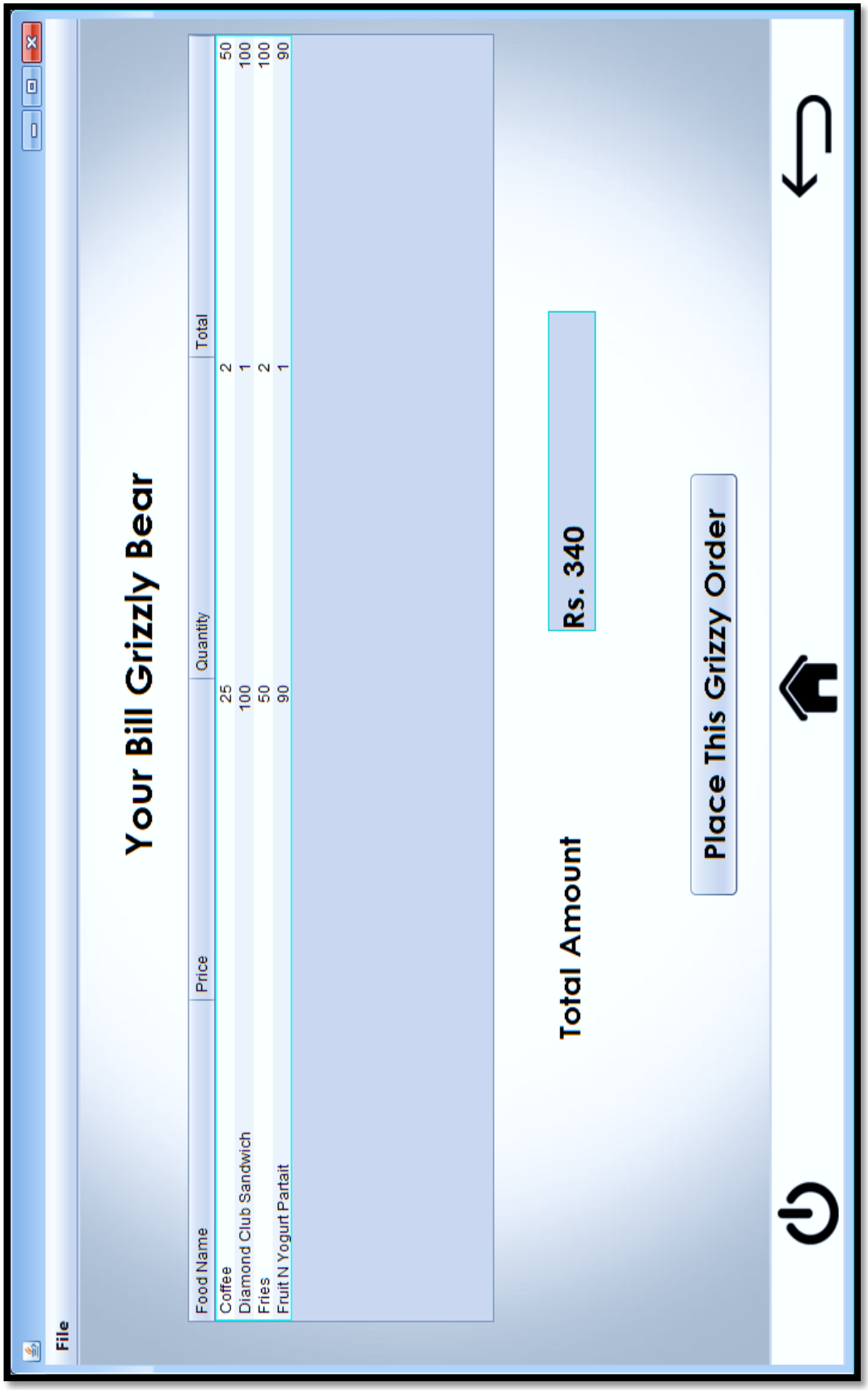
```
private void BreakfastItemStateChanged(java.awt.event.ItemEvent evt) {
      int food = Breakfast.getSelectedIndex();
      switch (food) {
         case 1:
            FoodName.setText("");
            FoodType.setText("");
            Price.setText("");
            FoodName.setText("Maharaja Mac");
            FoodType.setText("Breakfast");
            Price.setText("" + 159);
            break;
         case 2:
            FoodName.setText("");
            FoodType.setText("");
            Price.setText("");
            FoodName.setText("Fries");
            FoodType.setText("Breakfast");
            Price.setText("" + 50);
            break;
         case 3:
            FoodName.setText("");
```

```
    FoodType.setText("");
    Price.setText("");
    FoodName.setText("Happy Meal");
    FoodType.setText("Breakfast");
    Price.setText("" + 46);
    break;
case 4:
    FoodName.setText("");
    FoodType.setText("");
    Price.setText("");
    FoodName.setText("Fruit 'N Yogurt Partait");
    FoodType.setText("Breakfast");
    Price.setText("" + 90);
    break;
case 5:
    FoodName.setText("");
    FoodType.setText("");
    Price.setText("");
    FoodName.setText("Maxican Green Wave");
    FoodType.setText("Breakfast");
    Price.setText("" + 385);
    break;
case 6:
    FoodName.setText("");
    FoodType.setText("");
    Price.setText("");
    FoodName.setText("Veg Extravaganza");
    FoodType.setText("Breakfast");
```

```
            Price.setText("" + 450);
            break;
        case 7:
            FoodName.setText("");
            FoodType.setText("");
            Price.setText("");
            FoodName.setText("Veggie Paradise");
            FoodType.setText("Breakfast");
            Price.setText("" + 385);
            break;
        default:
            break;
     }
    Quantity.setValue(0);
    Amount.setText("");
  }
```

```
private void QuantityStateChanged(javax.swing.event.ChangeEvent evt) {
     int price = Integer.parseInt(Price.getText());
     int quantity = (int) Quantity.getValue();
     int b = price*quantity;
     Amount.setText(""+b);
  }
```

FRAME 12 (GRIZZLY BEAR BILL)



CODING:

```
package legionnaire_os;
import java.sql.*;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
/* @author Mandeep M. Dalavi */
public class GrizzlyBear_Bill extends javax.swing.JFrame {
   public GrizzlyBear_Bill() {
      initComponents();
//-------------------------JDBC displaying Bill--------------------------//
DefaultTableModel model = (DefaultTableModel)
BillTable.getModel();
try {
      Class.forName("java.sql.DriverManager");
      Connection con = (Connection)
      DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire","root", "admin");
      Statement stmt = (Statement) con.createStatement();
      String query="SELECT*from Bill;";
      ResultSet rs = stmt.executeQuery(query);
      while(rs.next())
      {
      String foodname = rs.getString(1);
      String price = rs.getString(2);
      String quantity = rs.getString(3);
      String total = rs.getString(4);
      model.addRow (new Object[] {foodname, price, quantity, total});
      }
      String query1 = "SELECT sum(total) from Bill";
```

```
      ResultSet RS = stmt.executeQuery(query1);
      while(RS.next())
      {
         String sum = RS.getString(1);
         TotalAmount.setText("Rs. "+sum);
      }
   }
catch (Exception e) {
      JOptionPane.showMessageDialog(this, "Error in Connectivity!");
   }
}
```

```
private void BackbuttonActionPerformed(java.awt.event.ActionEvent evt) {
      new GrizzlyBear_Menu().setVisible(true);
      this.setVisible(false);
   }
```

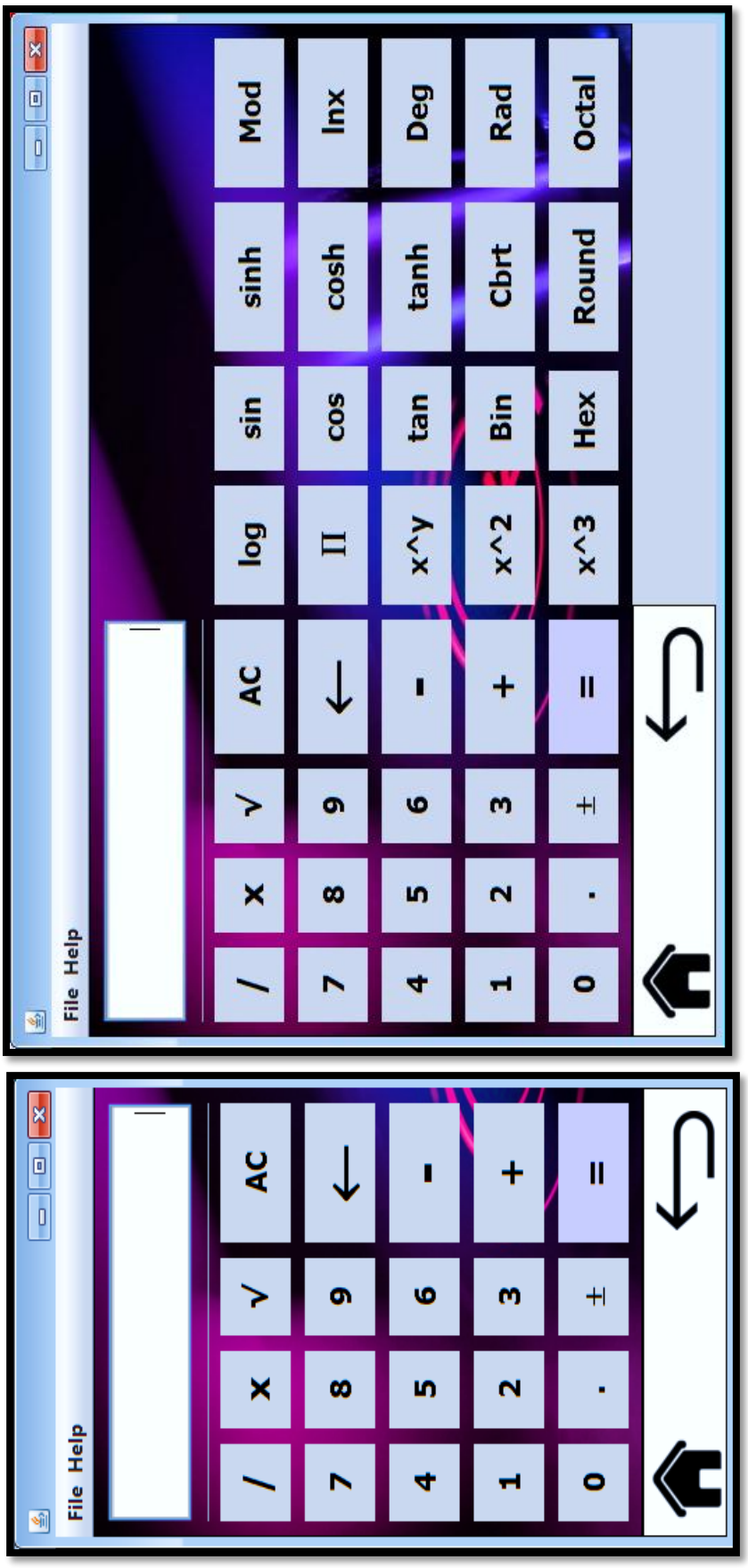
```
private void JumpHomemenuActionPerformed(java.awt.event.ActionEvent evt) {
      new Home().setVisible(true);
      this.setVisible(false);
   }
```

```
private void JumpAppmenuActionPerformed(java.awt.event.ActionEvent evt) {
      new Apps().setVisible(true);
      this.setVisible(false);
   }
```

```
private void ExitOSmenuActionPerformed(java.awt.event.ActionEvent evt) {
    if(JOptionPane.showConfirmDialog(this, "Are you sure you want to exit?", "Confirmation",
JOptionPane.YES_NO_CANCEL_OPTION)==JOptionPane.YES_OPTION)
        {
            System.exit(0);
        }
  }
-----------------------------------------------------------------------------------------------------
private void formWindowActivated(java.awt.event.WindowEvent evt) {
     this.setResizable(true);
     this.setSize(1025, 707);
  }
-----------------------------------------------------------------------------------------------------
private void HomebuttonActionPerformed(java.awt.event.ActionEvent evt) {
    new Home().setVisible(true);
    this.setVisible(false);
  }
-----------------------------------------------------------------------------------------------------
private void PowerbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    if(JOptionPane.showConfirmDialog(this, "Are you sure you want to exit?", "Confirmation",
JOptionPane.YES_NO_CANCEL_OPTION)==JOptionPane.YES_OPTION)
        {
            System.exit(0);
        }
  }
-----------------------------------------------------------------------------------------------------
private void PlaceOrderbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    DefaultTableModel model = (DefaultTableModel)
    BillTable.getModel(); int rows=model.getRowCount();
```

```
    if (rows>0)
      {
        for (int i=0; i<rows; i++)
        model.removeRow(0);
      }
    try {
      Class.forName("java.sql.DriverManager");
      Connection con = (Connection)
    DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire","root", "admin");
      Statement stmt = (Statement) con.createStatement();
      String query="DELETE from Bill;";
      stmt.executeUpdate(query);
      }
    catch(Exception e) {
      JOptionPane.showMessageDialog (this, “Error in Connectivity!”);
      }
    JOptionPane.showMessageDialog(this, "Your Order has been place."+'\n'+"Order will be
Delivered in 20-30min"+'\n'+"Please give the Total Amount to the Delivery Man."+'\n'+"Thank
You for using Grizzy Bear"+'\n'+"Please give feedbacks");
  }
```

FRAME 13 (CALCULATOR APP)



CODING:

```
package legionnaire_os;
import javax.swing.JOptionPane;
/* @author Mandeep M. Dalavi */
public class Calculator_App extends javax.swing.JFrame {
   double firstnum, secondnum, result;
   String operations;
   private Object math;
   public Calculator_App() {
      initComponents();
   }
-----------------------------------------------------------------------------------------------------
private void BackbuttonActionPerformed(java.awt.event.ActionEvent evt) {
      new Apps().setVisible(true);
      this.setVisible(false);
   }
-----------------------------------------------------------------------------------------------------
private void DividebuttonActionPerformed(java.awt.event.ActionEvent evt) {
      firstnum = Double.parseDouble(input.getText());
      input.setText(null);
      operations=("/");
   }
-----------------------------------------------------------------------------------------------------
private void formWindowActivated(java.awt.event.WindowEvent evt) {
      this.setResizable(true);
      this.setSize(306, 515);
   }
```

```
-----------------------------------------------------------------------------------------------------
private void StandardActionPerformed(java.awt.event.ActionEvent evt) {
      this.setResizable(true);
      this.setSize(306, 515);
      input.setSize(270, 60);
      Separator.setSize(270, 5);
      Powerpanel.setSize(290, 60);
      input.setSize(270, 60);
   }
-----------------------------------------------------------------------------------------------------
private void ScientificActionPerformed(java.awt.event.ActionEvent evt) {
      this.setResizable(true);
      this.setSize(696, 515);
      input.setSize(660, 60);
      Separator.setSize(660, 5);
      Powerpanel.setSize(680, 60);
      input.setSize(660, 60);
   }
-----------------------------------------------------------------------------------------------------
private void SevenbuttonActionPerformed(java.awt.event.ActionEvent evt) {
      String iNum = input.getText()+Sevenbutton.getText();
      input.setText(iNum);
   }
-----------------------------------------------------------------------------------------------------
private void EightbuttonActionPerformed(java.awt.event.ActionEvent evt) {
     String iNum = input.getText()+Eightbutton.getText();
     input.setText(iNum);
   }
```

```
private void NinebuttonActionPerformed(java.awt.event.ActionEvent evt) {
    String iNum = input.getText()+Ninebutton.getText();
    input.setText(iNum);
  }
-----------------------------------------------------------------------------------------------------
private void FourbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    String iNum = input.getText()+Fourbutton.getText();
    input.setText(iNum);
  }
-----------------------------------------------------------------------------------------------------
private void FivebuttonActionPerformed(java.awt.event.ActionEvent evt) {
    String iNum = input.getText()+Fivebutton.getText();
    input.setText(iNum);
  }
-----------------------------------------------------------------------------------------------------
private void SixbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    String iNum = input.getText()+Sixbutton.getText();
    input.setText(iNum);
  }
-----------------------------------------------------------------------------------------------------
private void OnebuttonActionPerformed(java.awt.event.ActionEvent evt) {
    String iNum = input.getText()+Onebutton.getText();
    input.setText(iNum);
  }
-----------------------------------------------------------------------------------------------------
private void TwobuttonActionPerformed(java.awt.event.ActionEvent evt) {
    String iNum = input.getText()+Twobutton.getText();
```

```
    input.setText(iNum);    }
private void ThreebuttonActionPerformed(java.awt.event.ActionEvent evt) {
    String iNum = input.getText()+Threebutton.getText();
    input.setText(iNum);
  }
-----------------------------------------------------------------------------------------------------
private void DeletebuttonActionPerformed(java.awt.event.ActionEvent evt) {
    String backspace = null;
    if(input.getText().length() > 0)
        {
          StringBuilder strB = new StringBuilder(input.getText());
          strB.deleteCharAt(input.getText().length()-1);
          backspace = strB.toString();
          input.setText(backspace);
        }
  }
-----------------------------------------------------------------------------------------------------
private void ZerobuttonActionPerformed(java.awt.event.ActionEvent evt) {
    String iNum = input.getText()+Zerobutton.getText();
    input.setText(iNum);
  }
-----------------------------------------------------------------------------------------------------
private void DotButtonActionPerformed(java.awt.event.ActionEvent evt) {
    if(!input.getText().contains("."))
        {
          input.setText(input.getText()+DotButton.getText());
        }
  }
```

```
private void PlusminusbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    double A = Double.parseDouble(String.valueOf(input.getText()));
    A = A*(-1);
    input.setText(String.valueOf(A));
  }
-----------------------------------------------------------------------------------------------------
private void SqrtbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    double A = Double.parseDouble(String.valueOf(input.getText()));
    A = Math.sqrt(A);
    input.setText(String.valueOf(A));
  }
-----------------------------------------------------------------------------------------------------
private void Logbutton1ActionPerformed(java.awt.event.ActionEvent evt) {
    double A = Double.parseDouble(String.valueOf(input.getText()));
    A = Math.log(A);
    input.setText(String.valueOf(A));
  }
-----------------------------------------------------------------------------------------------------
private void SinbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    double A = Double.parseDouble(String.valueOf(input.getText()));
    A = Math.sin(A);
    input.setText(String.valueOf(A));
  }
-----------------------------------------------------------------------------------------------------
private void SinhbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    double A = Double.parseDouble(String.valueOf(input.getText()));
    A = Math.sinh(A);
```

```
    input.setText(String.valueOf(A));    }
private void CbrbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    double A = Double.parseDouble(String.valueOf(input.getText()));
    A = Math.cbrt(A);
    input.setText(String.valueOf(A));
  }
-----------------------------------------------------------------------------------------------------
private void RoundbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    double A = Double.parseDouble(String.valueOf(input.getText()));
    A = Math.round(A);
    input.setText(String.valueOf(A));
  }
-----------------------------------------------------------------------------------------------------
private void CosbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    double A = Double.parseDouble(String.valueOf(input.getText()));
    A = Math.cos(A);
    input.setText(String.valueOf(A));
  }
-----------------------------------------------------------------------------------------------------
private void CoshbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    double A = Double.parseDouble(String.valueOf(input.getText()));
    A = Math.cosh(A);
    input.setText(String.valueOf(A));
  }
-----------------------------------------------------------------------------------------------------
private void TanbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    double A = Double.parseDouble(String.valueOf(input.getText()));
    A = Math.tan(A);
```

```
    input.setText(String.valueOf(A));    }
private void TanhbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    double A = Double.parseDouble(String.valueOf(input.getText()));
    A = Math.tanh(A);
    input.setText(String.valueOf(A));
  }
-----------------------------------------------------------------------------------------------------
private void xybuttonActionPerformed(java.awt.event.ActionEvent evt) {
    double A = Double.parseDouble(String.valueOf(input.getText()));
    A = Math.pow(A, A);
    input.setText(String.valueOf(A));
  }
-----------------------------------------------------------------------------------------------------
private void x2buttonActionPerformed(java.awt.event.ActionEvent evt) {
    double A = Double.parseDouble(String.valueOf(input.getText()));
    A = (A*A);
    input.setText(String.valueOf(A));
  }
-----------------------------------------------------------------------------------------------------
private void x3buttonActionPerformed(java.awt.event.ActionEvent evt) {
    double A = Double.parseDouble(String.valueOf(input.getText()));
    A = (A*A*A);
    input.setText(String.valueOf(A));
  }
-----------------------------------------------------------------------------------------------------
private void piebuttonActionPerformed(java.awt.event.ActionEvent evt) {
    double A;
    A = (3.14159365358979323846264338327 95);
```

```
    input.setText(String.valueOf(A));    }
private void ACbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    input.setText("");
  }
-----------------------------------------------------------------------------------------------------
private void HomebuttonActionPerformed(java.awt.event.ActionEvent evt) {
    new Home().setVisible(true);
    this.setVisible(false);
  }
-----------------------------------------------------------------------------------------------------
private void HexbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    int A = Integer.parseInt(input.getText());
    input.setText(Integer.toHexString(A));
  }
-----------------------------------------------------------------------------------------------------
private void BinbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    int A = Integer.parseInt(input.getText());
    input.setText(Integer.toBinaryString(A));
  }
-----------------------------------------------------------------------------------------------------
private void EqualbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    String answer;
    secondnum = Double.parseDouble(input.getText());
    if(operations=="+")
        {
            result = firstnum+secondnum;
            answer = String.format("%.2f", result);
            input.setText(answer);
```

```
        }
      else if(operations=="-")
        {
           result = firstnum-secondnum;
           answer = String.format("%.2f", result);
           input.setText(answer);
        }
     if(operations=="*")
        {
           result = firstnum*secondnum;
           answer = String.format("%.2f", result);
           input.setText(answer);
        }
      if(operations=="/")
        {
           result = firstnum/secondnum;
           answer = String.format("%.2f", result);
           input.setText(answer);
        }
      if(operations=="%")
        {
           result = firstnum%secondnum;
           answer = String.format("%.2f", result);
           input.setText(answer);
        }
   }
```

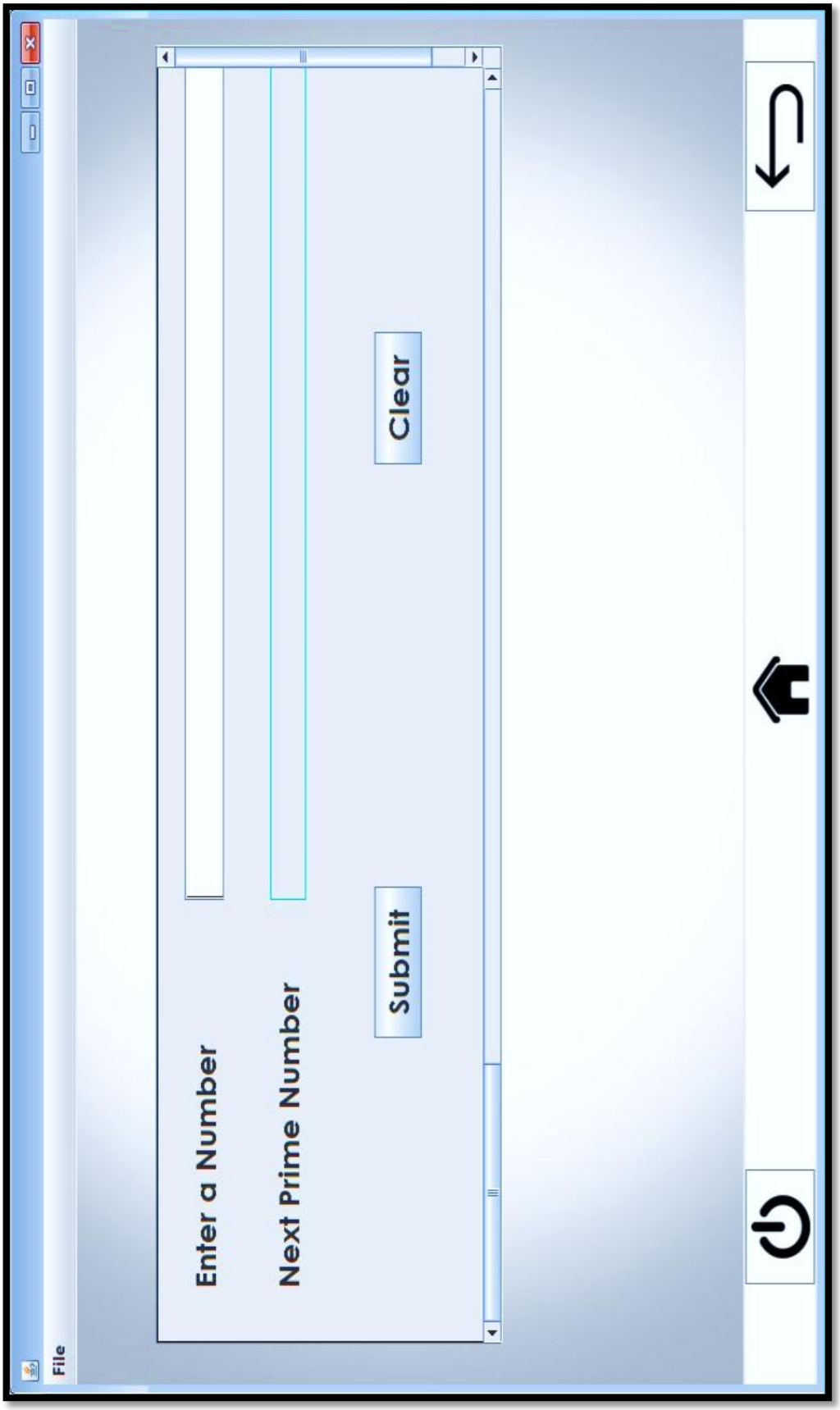
```
private void AddbuttonActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    firstnum = Double.parseDouble(input.getText());
     input.setText(null);
     operations=("+");
  }
----------------------------------------------------------------------------------------------------
private void SubtractbuttonActionPerformed(java.awt.event.ActionEvent evt) {
     firstnum = Double.parseDouble(input.getText());
     input.setText(null);
     operations=("-");
  }
----------------------------------------------------------------------------------------------------
private void MultiplybuttonActionPerformed(java.awt.event.ActionEvent evt) {
    firstnum = Double.parseDouble(input.getText());
    input.setText(null);
    operations=("*");
  }
----------------------------------------------------------------------------------------------------
private void ModbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    firstnum = Double.parseDouble(input.getText());
    input.setText(null);
    operations=("%");
  }
----------------------------------------------------------------------------------------------------
private void lnxbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    double A = Double.parseDouble(String.valueOf(input.getText()));
    A = Math.log10(A);
    input.setText(String.valueOf(A));
  }
```

```
private void DegbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    double A = Double.parseDouble(String.valueOf(input.getText()));
    A = Math.toDegrees(A);
    input.setText(String.valueOf(A));
  }
-----------------------------------------------------------------------------------------------------
private void RadbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    double A = Double.parseDouble(String.valueOf(input.getText()));
    A = Math.toRadians(A);
    input.setText(String.valueOf(A));
  }
-----------------------------------------------------------------------------------------------------
private void OctalbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    int A = Integer.parseInt(input.getText());
    input.setText(Integer.toOctalString(A));
  }
-----------------------------------------------------------------------------------------------------
private void JumpHomemenuActionPerformed(java.awt.event.ActionEvent evt) {
    new Home().setVisible(true);
    this.setVisible(false);
  }
-----------------------------------------------------------------------------------------------------
private void JumpAppmenuActionPerformed(java.awt.event.ActionEvent evt) {
    new Apps().setVisible(true);
    this.setVisible(false);
  }
```

```
private void ExitOSmenuActionPerformed(java.awt.event.ActionEvent evt) {

    if(JOptionPane.showConfirmDialog(this, "Are you sure you want to exit?", "Confirmation",
JOptionPane.YES_NO_CANCEL_OPTION)==JOptionPane.YES_OPTION)

        {

          System.exit(0);

        }

  }
```

FRAME 14 (NEXT PRIME NO.)

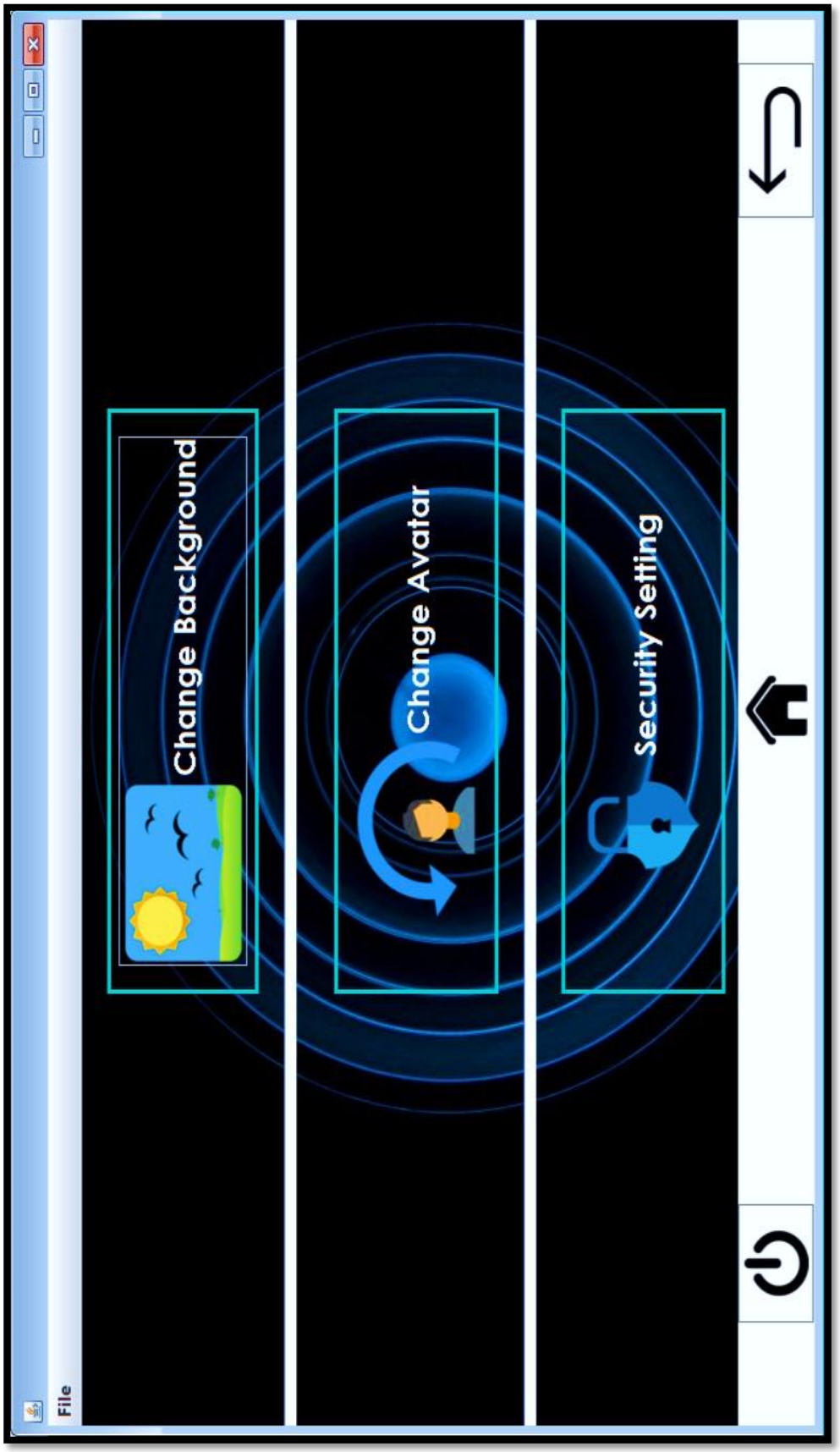


CODING:

```
package legionnaire_os;
import java.math.BigInteger;
import javax.swing.JOptionPane;
/* @author Mandeep M. Dalavi */
public class Next_Prime_No_App extends javax.swing.JFrame {
   public Next_Prime_No_App() {
      initComponents();   }
-----------------------------------------------------------------------------------------------------
private void BackbuttonActionPerformed(java.awt.event.ActionEvent evt) {
     new Apps().setVisible(true);
     this.setVisible(false);
   }
-----------------------------------------------------------------------------------------------------
private void SubmitActionPerformed(java.awt.event.ActionEvent evt) {
     BigInteger n = new BigInteger(Enterno.getText());
     BigInteger p = new BigInteger("" + n.nextProbablePrime());
     Nextno.setText(p + "");
   }
-----------------------------------------------------------------------------------------------------
private void JumpHomemenuActionPerformed(java.awt.event.ActionEvent evt) {
     new Home().setVisible(true);
     this.setVisible(false);
  }
-----------------------------------------------------------------------------------------------------
private void JumpAppmenuActionPerformed(java.awt.event.ActionEvent evt) {
     new Apps().setVisible(true);
     this.setVisible(false);   }
```

```
private void ExitOSmenuActionPerformed(java.awt.event.ActionEvent evt) {
    if(JOptionPane.showConfirmDialog(this, "Are you sure you want to exit?", "Confirmation",
JOptionPane.YES_NO_CANCEL_OPTION)==JOptionPane.YES_OPTION)
        {
           System.exit(0);
        }
  }
-----------------------------------------------------------------------------------------------------
private void formWindowActivated(java.awt.event.WindowEvent evt) {
    this.setResizable(true);
    this.setSize(1025, 707);
  }
-----------------------------------------------------------------------------------------------------
private void HomebuttonActionPerformed(java.awt.event.ActionEvent evt) {
    new Home().setVisible(true);
    this.setVisible(false);
  }
-----------------------------------------------------------------------------------------------------
private void PowerbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    if(JOptionPane.showConfirmDialog(this, "Are you sure you want to exit?", "Confirmation",
JOptionPane.YES_NO_CANCEL_OPTION)==JOptionPane.YES_OPTION)
        {   System.exit(0);   }
   }
-----------------------------------------------------------------------------------------------------
private void ClearActionPerformed(java.awt.event.ActionEvent evt) {
     Enterno.setText("");
     Nextno.setText("");
  }
```

FRAME 15 (SETTING APP)



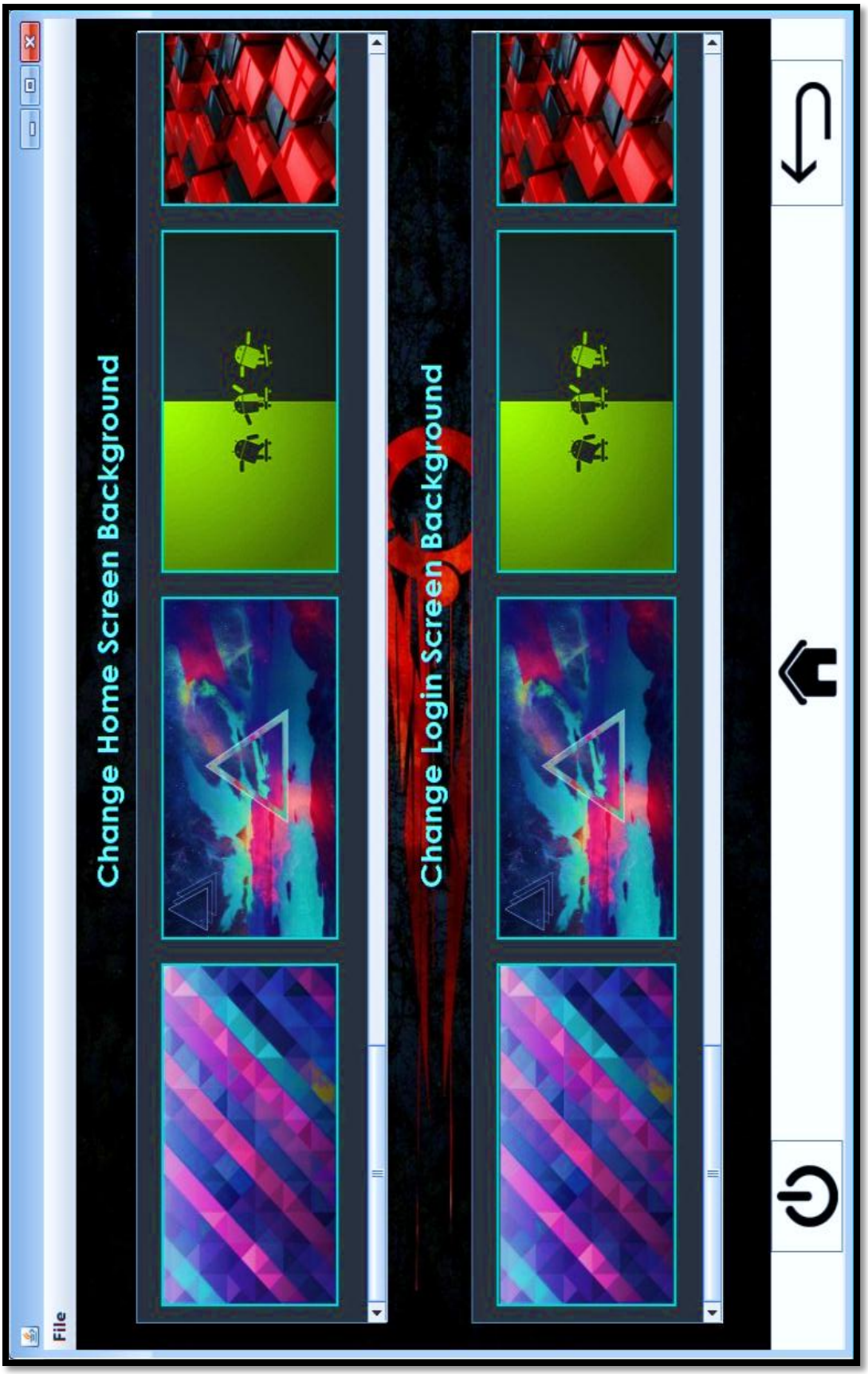
CODING:

```
package legionnaire_os;
import javax.swing.JOptionPane;
/* @author Mandeep M. Dalavi */
public class Setting_App extends javax.swing.JFrame {
   public Setting_App() {
      initComponents();
   }
-----------------------------------------------------------------------------------------------------
private void BackbuttonActionPerformed(java.awt.event.ActionEvent evt) {
      new Apps().setVisible(true);
      this.setVisible(false);
   }
-----------------------------------------------------------------------------------------------------
private void ChangeBackgroundbuttonActionPerformed(java.awt.event.ActionEvent evt) {
      new Change_Background().setVisible(true);
      this.setVisible(false);
   }
-----------------------------------------------------------------------------------------------------
private void ChangeAvatarbuttonActionPerformed(java.awt.event.ActionEvent evt) {
      new Change_Avatar().setVisible(true);
      this.setVisible(false);
   }
-----------------------------------------------------------------------------------------------------
private void SecuritySettingbuttonActionPerformed(java.awt.event.ActionEvent evt) {
      new Security_Setting().setVisible(true);
      this.setVisible(false);    }
```

```
private void PowerbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    if(JOptionPane.showConfirmDialog(this, "Are you sure you want to exit?", "Confirmation",
JOptionPane.YES_NO_CANCEL_OPTION)==JOptionPane.YES_OPTION)
        {
           System.exit(0);
        }
  }
-----------------------------------------------------------------------------------------------------
private void HomebuttonActionPerformed(java.awt.event.ActionEvent evt) {
     new Home().setVisible(true);
     this.setVisible(false);
  }
-----------------------------------------------------------------------------------------------------
private void formWindowActivated(java.awt.event.WindowEvent evt) {
    this.setResizable(true);
    this.setSize(1025, 646);
  }
-----------------------------------------------------------------------------------------------------
private void JumpHomemenuActionPerformed(java.awt.event.ActionEvent evt) {
    new Home().setVisible(true);
    this.setVisible(false);
  }
-----------------------------------------------------------------------------------------------------
private void JumpAppmenuActionPerformed(java.awt.event.ActionEvent evt) {
    new Apps().setVisible(true);
    this.setVisible(false);
  }
```

```
private void ExitOSmenuActionPerformed(java.awt.event.ActionEvent evt) {

    if(JOptionPane.showConfirmDialog(this, "Are you sure you want to exit?", "Confirmation",
JOptionPane.YES_NO_CANCEL_OPTION)==JOptionPane.YES_OPTION)

        {

          System.exit(0);

        }

  }
```

FRAME 16 (CHANGE BACKGROUND)



CODING:

```
package legionnaire_os;

import java.sql.*;

import javax.swing.JOptionPane;

/* @author Mandeep M. Dalavi */

public class Change_Background extends javax.swing.JFrame {

   public Change_Background() {

      initComponents();

   }

-----------------------------------------------------------------------------------------------------

private void BackbuttonActionPerformed(java.awt.event.ActionEvent evt) {

      new Setting_App().setVisible(true);

      this.setVisible(false);

   }

-----------------------------------------------------------------------------------------------------

private void JumpHomemenuActionPerformed(java.awt.event.ActionEvent evt) {

      new Home().setVisible(true);

      this.setVisible(false);

   }

-----------------------------------------------------------------------------------------------------

private void JumpAppmenuActionPerformed(java.awt.event.ActionEvent evt) {

      new Apps().setVisible(true);

      this.setVisible(false);

   }

-----------------------------------------------------------------------------------------------------

private void ExitOSmenuActionPerformed(java.awt.event.ActionEvent evt) {

      if(JOptionPane.showConfirmDialog(this, "Are you sure you want to exit?", "Confirmation",
JOptionPane.YES_NO_CANCEL_OPTION)==JOptionPane.YES_OPTION)
```

```
        {
           System.exit(0);
        }
  }
-----------------------------------------------------------------------------------------------------
private void formWindowActivated(java.awt.event.WindowEvent evt) {
    this.setResizable(true);
    this.setSize(1025, 707);
  }
-----------------------------------------------------------------------------------------------------
private void HomebuttonActionPerformed(java.awt.event.ActionEvent evt) {
    new Home().setVisible(true);
    this.setVisible(false);
  }
-----------------------------------------------------------------------------------------------------
private void PowerbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    if(JOptionPane.showConfirmDialog(this, "Are you sure you want to exit?", "Confirmation",
JOptionPane.YES_NO_CANCEL_OPTION)==JOptionPane.YES_OPTION)
        {
           System.exit(0);
        }
  }
-----------------------------------------------------------------------------------------------------
private void homebackground1MouseClicked(java.awt.event.MouseEvent evt) {
    try
        {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
```

```
        Statement stmt = (Statement) con.createStatement();
        String query = "Update backgrounds set homescreen='back1.jpg';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Background set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
     catch (Exception e)
        {
           JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }
  }
-----------------------------------------------------------------------------------------------
private void homebackground2MouseClicked(java.awt.event.MouseEvent evt) {
    try
        {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "Update backgrounds set homescreen='back2.jpg';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Background set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e)
        {
           JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }
  }
```

```
private void homebackground3MouseClicked(java.awt.event.MouseEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "Update backgrounds set homescreen='back3.jpg';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Background set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
        JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }
  }
```

```
private void homebackground4MouseClicked(java.awt.event.MouseEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "Update backgrounds set homescreen='back4.jpg';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Background set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
            JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }   }
```

```
private void homebackground5MouseClicked(java.awt.event.MouseEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "Update backgrounds set homescreen='back5.jpg';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Background set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
            JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }
  }
```

```
private void homebackground6MouseClicked(java.awt.event.MouseEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "Update backgrounds set homescreen='back6.jpg';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Background set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
            JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }   }
```

```
private void homebackground8MouseClicked(java.awt.event.MouseEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "Update backgrounds set homescreen='back8.jpg';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Background set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
            JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }
  }
```

```
private void homebackground9MouseClicked(java.awt.event.MouseEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "Update backgrounds set homescreen='back9.jpg';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Background set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
            JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }   }
```

```
private void homebackground10MouseClicked(java.awt.event.MouseEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "Update backgrounds set homescreen='back10.jpg';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Background set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
           JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }
  }
```

```
private void homebackground11MouseClicked(java.awt.event.MouseEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "Update backgrounds set homescreen='back11.jpg';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Background set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
           JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }   }
```

```
private void homebackground12MouseClicked(java.awt.event.MouseEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "Update backgrounds set homescreen='back12.jpg';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Background set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
           JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }
  }
```

```
private void homebackground13MouseClicked(java.awt.event.MouseEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "Update backgrounds set homescreen='back13.jpg';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Background set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
catch (Exception e) {
           JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }   }
```

```
private void homebackground14MouseClicked(java.awt.event.MouseEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "Update backgrounds set homescreen='back14.jpg';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Background set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
            JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }
  }
```

```
 private void homebackground15MouseClicked(java.awt.event.MouseEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "Update backgrounds set homescreen='back15.jpg';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Background set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
            JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }   }
```

```
private void homebackground16MouseClicked(java.awt.event.MouseEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "Update backgrounds set homescreen='back16.jpg';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Background set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
            JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }
  }
```

```
private void homebackground17MouseClicked(java.awt.event.MouseEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "Update backgrounds set homescreen='back17.jpg';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Background set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
          JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }   }
```

```
private void homebackground18MouseClicked(java.awt.event.MouseEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "Update backgrounds set homescreen='back18.jpg';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Background set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
            JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }
  }
```

```
private void loginbackground1MouseClicked(java.awt.event.MouseEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "Update backgrounds set loginscreen='back1.jpg';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Background set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
          JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }   }
```

```
private void loginbackground2MouseClicked(java.awt.event.MouseEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "Update backgrounds set loginscreen='back2.jpg';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Background set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
            JOptionPane.showMessageDialog(this, "Error in Connectivity!");
        }
  }
```

```
private void loginbackground3MouseClicked(java.awt.event.MouseEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "Update backgrounds set loginscreen='back3.jpg';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Background set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
            JOptionPane.showMessageDialog(this, "Error in Connectivity!");
        }   }
```

```
private void loginbackground4MouseClicked(java.awt.event.MouseEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "Update backgrounds set loginscreen='back4.jpg';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Background set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
            JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }
  }
```

```
private void loginbackground5MouseClicked(java.awt.event.MouseEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "Update backgrounds set loginscreen='back5.jpg';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Background set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
           JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }   }
```

```
private void loginbackground6MouseClicked(java.awt.event.MouseEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "Update backgrounds set loginscreen='back6.jpg';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Background set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
            JOptionPane.showMessageDialog(this, "Error in Connectivity!");
        }
  }
```

```
private void loginbackground8MouseClicked(java.awt.event.MouseEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "Update backgrounds set loginscreen='back8.jpg';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Background set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
            JOptionPane.showMessageDialog(this, "Error in Connectivity!");
        }   }
```

```
private void loginbackground9MouseClicked(java.awt.event.MouseEvent evt) {
    try{
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "Update backgrounds set loginscreen='back9.jpg';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Background set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
            JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }
  }
---------------------------------------------------------------------------------------------------
private void loginbackground10MouseClicked(java.awt.event.MouseEvent evt) {
try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "Update backgrounds set loginscreen='back10.jpg';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Background set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
catch (Exception e) {
            JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }   }
```

```
private void loginbackground11MouseClicked(java.awt.event.MouseEvent evt) {
try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "Update backgrounds set loginscreen='back11.jpg';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Background set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
catch (Exception e) {
            JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }
  }
---------------------------------------------------------------------------------------------------
private void loginbackground12MouseClicked(java.awt.event.MouseEvent evt) {
try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "Update backgrounds set loginscreen='back12.jpg';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Background set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
catch (Exception e) {
            JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }   }
```

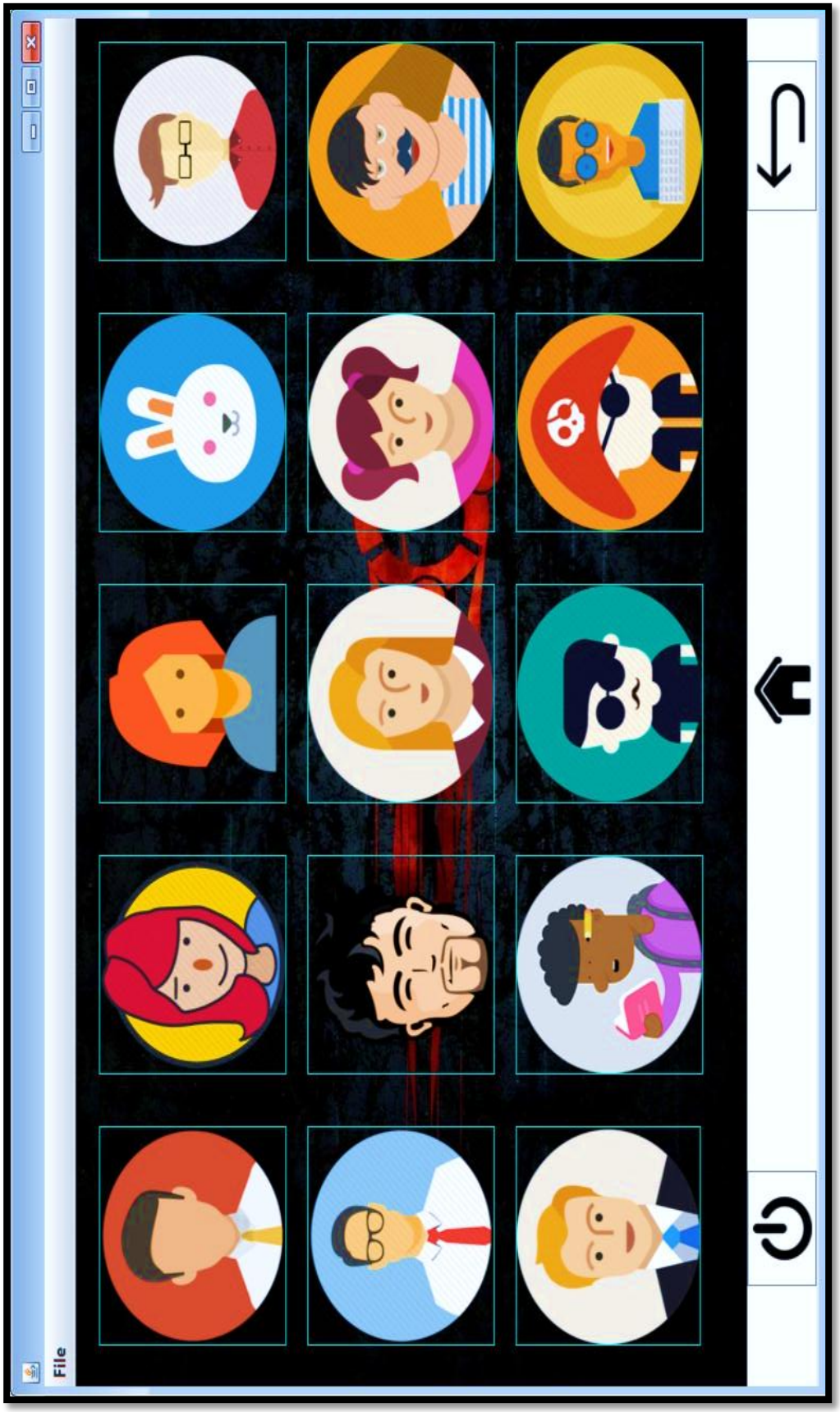
```
private void loginbackground13MouseClicked(java.awt.event.MouseEvent evt) {
try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "Update backgrounds set loginscreen='back13.jpg';";
         stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Background set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
            JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }
  }
---------------------------------------------------------------------------------------------------
private void loginbackground14MouseClicked(java.awt.event.MouseEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "Update backgrounds set loginscreen='back14.jpg';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Background set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
           JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }   }
```

```
private void loginbackground15MouseClicked(java.awt.event.MouseEvent evt) {
      try {
         Class.forName("java.sql.Driver");
         Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
         Statement stmt = (Statement) con.createStatement();
         String query = "Update backgrounds set loginscreen='back15.jpg';";
         stmt.executeUpdate(query);
         JOptionPane.showMessageDialog(this, "Background set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
         }
     catch (Exception e) {
            JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
         }
   }
---------------------------------------------------------------------------------------------------
private void loginbackground16MouseClicked(java.awt.event.MouseEvent evt) {
     try {
         Class.forName("java.sql.Driver");
         Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
         Statement stmt = (Statement) con.createStatement();
         String query = "Update backgrounds set loginscreen='back16.jpg';";
         stmt.executeUpdate(query);
         JOptionPane.showMessageDialog(this, "Background set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
         }
     catch (Exception e) {
             JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
         }   }
```

```
private void loginbackground17MouseClicked(java.awt.event.MouseEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "Update backgrounds set loginscreen='back17.jpg';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Background set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
            JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }
  }
```

```
private void loginbackground18MouseClicked(java.awt.event.MouseEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "Update backgrounds set loginscreen='back18.jpg';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Background set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
            JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }   }
```

FRAME 17 (CHANGE AVATAR)



CODING:

```
package legionnaire_os;

import java.sql.*;

import javax.swing.JOptionPane;

/* @author Mandeep M. Dalavi  */

public class Change_Avatar extends javax.swing.JFrame {

   public Change_Avatar() {

      initComponents();   }

-----------------------------------------------------------------------------------------------------

private void BackbuttonActionPerformed(java.awt.event.ActionEvent evt) {

     new Setting_App().setVisible(true);

     this.setVisible(false);   }

-----------------------------------------------------------------------------------------------------

private void JumpHomemenuActionPerformed(java.awt.event.ActionEvent evt) {

     new Home().setVisible(true);

     this.setVisible(false);

   }

-----------------------------------------------------------------------------------------------------

private void JumpAppmenuActionPerformed(java.awt.event.ActionEvent evt) {

     new Apps().setVisible(true);

     this.setVisible(false);

   }

-----------------------------------------------------------------------------------------------------

private void ExitOSmenuActionPerformed(java.awt.event.ActionEvent evt) {

     if(JOptionPane.showConfirmDialog(this, "Are you sure you want to exit?", "Confirmation",
JOptionPane.YES_NO_CANCEL_OPTION)==JOptionPane.YES_OPTION)

         {   System.exit(0);   }

   }
```

```
private void formWindowActivated(java.awt.event.WindowEvent evt) {
      this.setResizable(true);
      this.setSize(1025, 707);
  }
-----------------------------------------------------------------------------------------------------
private void HomebuttonActionPerformed(java.awt.event.ActionEvent evt) {
    new Home().setVisible(true);
    this.setVisible(false);
  }
-----------------------------------------------------------------------------------------------------
private void PowerbuttonActionPerformed(java.awt.event.ActionEvent evt) {
    if(JOptionPane.showConfirmDialog(this, "Are you sure you want to exit?", "Confirmation",
JOptionPane.YES_NO_CANCEL_OPTION)==JOptionPane.YES_OPTION)
        {   System.exit(0);   }    }
-----------------------------------------------------------------------------------------------------
private void user1buttonActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "UPDATE UserAvatar SET avatar='user1.png';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Avatar set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
  }
    catch (Exception e) {
            JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }   }
```

```
private void user2buttonActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "UPDATE UserAvatar SET avatar='user2.png';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Avatar set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
           JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }
  }
```

```
private void user3buttonActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "UPDATE UserAvatar SET avatar='user3.png';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Avatar set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
           JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }   }
```

```
private void user4buttonActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "UPDATE UserAvatar SET avatar='user4.png';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Avatar set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
            JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }
  }
-----------------------------------------------------------------------------------------------
private void user5buttonActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "UPDATE UserAvatar SET avatar='user5.png';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Avatar set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
           JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }   }
```

```
private void user6buttonActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "UPDATE UserAvatar SET avatar='user6.png';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Avatar set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
           JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }
  }
-----------------------------------------------------------------------------------------------
private void user7buttonActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "UPDATE UserAvatar SET avatar='user7.png';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Avatar set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
           JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }   }
```

```
private void user8buttonActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection
)DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "UPDATE UserAvatar SET avatar='user8.png';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Avatar set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
            JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }
  }
```

```
private void user9buttonActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "UPDATE UserAvatar SET avatar='user9.png';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Avatar set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
           JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }   }
```

```
private void user10buttonActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "UPDATE UserAvatar SET avatar='user10.png';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Avatar set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
           JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }
  }
```

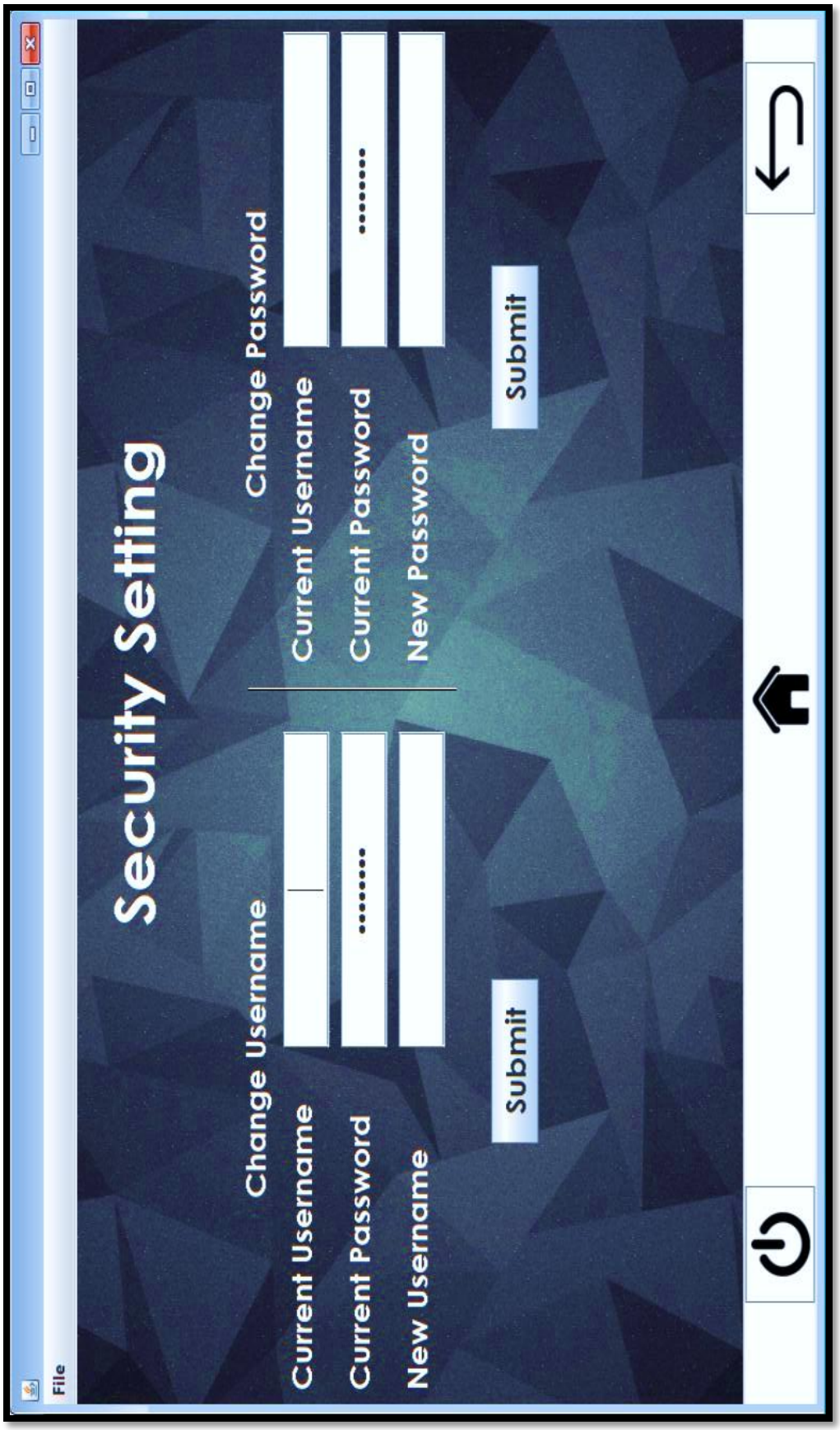
```
private void user11buttonActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "UPDATE UserAvatar SET avatar='user11.png';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Avatar set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
           JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }   }
```

```
private void user12buttonActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "UPDATE UserAvatar SET avatar='user12.png';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Avatar set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
           JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }
  }
-----------------------------------------------------------------------------------------------
private void user13buttonActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "UPDATE UserAvatar SET avatar='user13.png';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Avatar set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
           JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }   }
```

```
private void user14buttonActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "UPDATE UserAvatar SET avatar='user14.png';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Avatar set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
           JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }
  }
```

```
private void user15buttonActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        Class.forName("java.sql.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
        Statement stmt = (Statement) con.createStatement();
        String query = "UPDATE UserAvatar SET avatar='user15.png';";
        stmt.executeUpdate(query);
        JOptionPane.showMessageDialog(this, "Avatar set successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        }
    catch (Exception e) {
           JOptionPane.showMessageDialog(this, “Error in Connectivity!”);
        }   }
```

FRAME 18 (SECURITY SETTING)



CODING:

```
package legionnaire_os;
import java.sql.*;
import java.awt.Color;
import javax.swing.JOptionPane;
/* @author Mandeep M. Dalavi */
public class Security_Setting extends javax.swing.JFrame {
   String curusername, curpassword;
   String user, pass, newuser, newpass;
   public Security_Setting() {
      initComponents();
//--------------------JDBC part getting username and password--------------------//
try {
      Class.forName("java.sql.Driver");
      Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");
      Statement stmt = (Statement) con.createStatement();
      String query = "Select * from userpass";
      ResultSet rs = stmt.executeQuery(query);
      if (rs.first())
        {
          curusername = rs.getString(1);
          curpassword = rs.getString(2);
        }
   }
catch (Exception e) {
      JOptionPane.showMessageDialog(this, "Error in Connectivity!");
   }   }
```

```
private void JumpHomemenuActionPerformed(java.awt.event.ActionEvent evt) {
      new Home().setVisible(true);
      this.setVisible(false);
   }
-----------------------------------------------------------------------------------------------------
private void JumpAppmenuActionPerformed(java.awt.event.ActionEvent evt) {
      new Apps().setVisible(true);
      this.setVisible(false);
   }
-----------------------------------------------------------------------------------------------------
private void ExitOSmenuActionPerformed(java.awt.event.ActionEvent evt) {
      if(JOptionPane.showConfirmDialog(this, "Are you sure you want to exit?", "Confirmation",
JOptionPane.YES_NO_CANCEL_OPTION)==JOptionPane.YES_OPTION)
         {
            System.exit(0);
         }
   }
-----------------------------------------------------------------------------------------------------
private void HomebuttonActionPerformed(java.awt.event.ActionEvent evt) {
      new Home().setVisible(true);
      this.setVisible(false);
   }
-----------------------------------------------------------------------------------------------------
private void BackbuttonActionPerformed(java.awt.event.ActionEvent evt) {
      new Setting_App().setVisible(true);
      this.setVisible(false);
   }
```

```
private void PowerbuttonActionPerformed(java.awt.event.ActionEvent evt) {

     if(JOptionPane.showConfirmDialog(this, "Are you sure you want to exit?", "Confirmation",
JOptionPane.YES_NO_CANCEL_OPTION)==JOptionPane.YES_OPTION)

        {        System.exit(0);       }

  }

-----------------------------------------------------------------------------------------------------

private void formWindowActivated(java.awt.event.WindowEvent evt) {

    this.setResizable(true);

    this.setSize(1025, 707);

  }

-----------------------------------------------------------------------------------------------------

private void SubmitleftActionPerformed(java.awt.event.ActionEvent evt) {

    user = CurUserleft.getText();

    pass = CurPassleft.getText();

    newuser = NewUserleft.getText();

    if(user.isEmpty() || pass.isEmpty() || newuser.isEmpty())

        {

           JOptionPane.showMessageDialog(this, "Fields can't be Empty");

        }

    else {

           if(user.equals(curusername) && pass.equals(curpassword))

        {

     try {

        Class.forName("java.sql.DriverManager");

        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/legionnaire", "root", "admin");

        Statement stmt = (Statement) con.createStatement();

        String query = "UPDATE userpass SET username = '"+newuser+"' WHERE (username =
'"+user+"' && password = '"+pass+"')";
```

```
        ResultSet rs = stmt.executeQuery(query);
        if(rs.first())
          {
            curusername = rs.getString(1);
            curpassword = rs.getString(2);
            JOptionPane.showMessageDialog(this, "Username changed successfully");
          }
        }
      catch(Exception e)
        {
          JOptionPane.showMessageDialog(this, "Error in Connectivity!");
        }
      }
    else
        {
          JOptionPane.showMessageDialog(this, "Username and Password doesn't match. Please
try again", "Incorrect Username and Password", JOptionPane.ERROR_MESSAGE);
        }
      }
    }
```

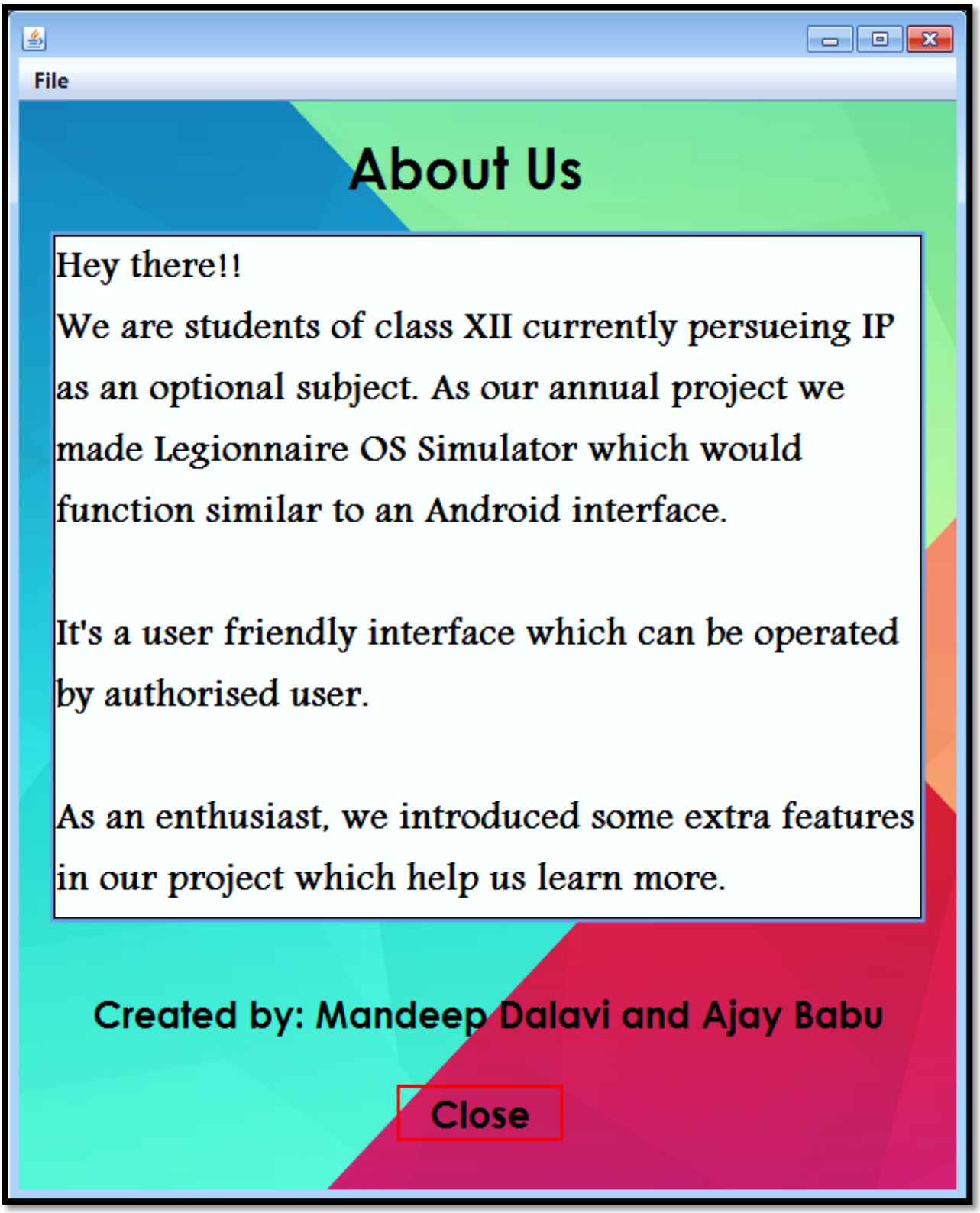
```
private void SubmitrightActionPerformed(java.awt.event.ActionEvent evt) {
      user = CurUserright.getText();
      pass = CurPassright.getText();
      newpass = NewPassright.getText();
      if(user.isEmpty() || pass.isEmpty() || newpass.isEmpty())
        {
          JOptionPane.showMessageDialog(this, "Fields can't be Empty");
        }
```

```
    else{
        if(user.equals(curusername) && pass.equals(curpassword))
          {
            try {
                Class.forName("java.sql.DriverManager");
                Connection con = (Connection)
        DriverManager.getConnection("jdbc:mysql::://localhost:3306/legionnaire", "root",
        "admin");
                Statement stmt = (Statement) con.createStatement();
                String query = "UPDATE userpass SET password = '"+newpass+"' WHERE (username
        = '"+user+"' && password = '"+pass+"')";
                ResultSet rs = stmt.executeQuery(query);
                if(rs.first())
                {
                   curusername = rs.getString(1);
                   curpassword = rs.getString(2);
                   JOptionPane.showMessageDialog(this, "Password changed successfully");
                }
            }
            catch (Exception e) {
                JOptionPane.showMessageDialog(this, "Error in Connectivity!");
            }
          }
          else   {
            JOptionPane.showMessageDialog(this, "Username and Password dosn't match. Please
        try again", "Incorrect Username or Password", JOptionPane.ERROR_MESSAGE);
          }
        }
}
```

```
private void CurPassleftMouseClicked(java.awt.event.MouseEvent evt) {
    CurPassleft.setText("");
  }
-----------------------------------------------------------------------------------------------------
private void CurPassrightMouseClicked(java.awt.event.MouseEvent evt) {
    CurPassright.setText("");
  }
-----------------------------------------------------------------------------------------------------
private void CurUserleftCaretUpdate(javax.swing.event.CaretEvent evt) {
    user = CurUserleft.getText();
    if(user.equals(curusername))
       {
          CurUserleft.setForeground(Color.GREEN);
       }
    else {
          CurUserleft.setForeground(Color.RED);
       }
  }
-----------------------------------------------------------------------------------------------------
private void CurPassleftCaretUpdate(javax.swing.event.CaretEvent evt) {
    pass = CurPassleft.getText();
    if(pass.equals(curpassword))
       {
          CurPassleft.setForeground(Color.GREEN);
       }
    else {
          CurPassleft.setForeground(Color.RED);
       }   }
```

```
private void CurUserrightCaretUpdate(javax.swing.event.CaretEvent evt) {
    user = CurUserright.getText();
    if(user.equals(curusername))
        {
          CurUserright.setForeground(Color.GREEN);
        }
    else {
          CurUserright.setForeground(Color.RED);
        }
  }

private void CurPassrightCaretUpdate(javax.swing.event.CaretEvent evt) {
    pass = CurPassright.getText();
    if(pass.equals(curpassword))
        {
          CurPassright.setForeground(Color.GREEN);
        }
    else {
          CurPassright.setForeground(Color.RED);
        }
  }
```

FRAME 19 (ABOUT US)



CODING:

```
package legionnaire_os;
import javax.swing.JOptionPane;
/* @author Mandeep M. Dalavi  */
public class About_Us extends javax.swing.JFrame {
   public About_Us() {
      initComponents();
   }
-----------------------------------------------------------------------------------------------------
private void formWindowActivated(java.awt.event.WindowEvent evt) {
      this.setResizable(true);
      this.setSize(636, 740);
   }
-----------------------------------------------------------------------------------------------------
private void JumpHomemenuActionPerformed(java.awt.event.ActionEvent evt) {
      new Home().setVisible(true);
      this.setVisible(false);
   }
-----------------------------------------------------------------------------------------------------
private void JumpAppmenuActionPerformed(java.awt.event.ActionEvent evt) {
      new Apps().setVisible(true);
      this.setVisible(false);
   }
-----------------------------------------------------------------------------------------------------
private void ExitOSmenuActionPerformed(java.awt.event.ActionEvent evt) {
     if(JOptionPane.showConfirmDialog(this, "Are you sure you want to exit?", "Confirmation",
JOptionPane.YES_NO_CANCEL_OPTION)==JOptionPane.YES_OPTION)
         {   System.exit(0);   }   }
```

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    this.setVisible(false);
  }
```

BACKEND TABLES

TABLE BACKGROUNDS

```
mysql> SELECT * from backgrounds;
+------------+-------------+
| Homescreen | Loginscreen |
+------------+-------------+
| back1.jpg  | back1.jpg   |
+------------+-------------+
1 row in set (0.00 sec)
```

DESCRIPTION OF BACKGROUNDS TABLE

```
mysql> DESC backgrounds;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| Homescreen  | varchar(50) | YES  |     | NULL    |       |
| Loginscreen | varchar(50) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
2 rows in set (0.14 sec)
```

TABLE BILL

```
mysql> SELECT * from bill;
Empty set (0.00 sec)
```

DESCRIPTION OF BILL TABLE

```
mysql> DESC bill;
+-----------+----------+------+-----+---------+-------+
| Field     | Type     | Null | Key | Default | Extra |
+-----------+----------+------+-----+---------+-------+
| Food_Name | char(25) | YES  |     | NULL    |       |
| Price     | int(10)  | YES  |     | NULL    |       |
| Quantity  | int(5)   | YES  |     | NULL    |       |
| Total     | int(25)  | YES  |     | NULL    |       |
+-----------+----------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

TABLE CALLING

```
mysql> SELECT * from calling;
+----------------+------------+
| Name           | Number     |
+----------------+------------+
| Mandeep Dalavi | 8669525368 |
+----------------+------------+
1 row in set (0.00 sec)
```

DESCRIPTION OF CALLING TABLE

```
mysql> DESC calling;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| Name   | varchar(30) | YES  |     | NULL    |       |
| Number | bigint(10)  | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

TABLE CONTACTS

```
mysql> SELECT * from contacts;
+-------+------------+-----------+------------+
| Sr_No | First_Name | Last_Name | Number     |
+-------+------------+-----------+------------+
|     1 | Mandeep    | Dalavi    | 8769525368 |
|     2 | Ajay       | Babu      | 7645458926 |
+-------+------------+-----------+------------+
2 rows in set (0.00 sec)
```

DESCRIPTION OF CONTACTS TABLE

```
mysql> DESC contacts;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| Sr_No      | int(3)      | NO   | PRI | NULL    |       |
| First_Name | varchar(30) | YES  |     | NULL    |       |
| Last_Name  | varchar(30) | YES  |     | NULL    |       |
| Number     | bigint(10)  | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
4 rows in set (0.09 sec)
```

TABLE SEARCH

```
mysql> SELECT * from search;
Empty set (0.00 sec)
```

DESCRIPTION OF SEARCH TABLE

```
mysql> DESC search;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| Name  | varchar(30) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
1 row in set (0.00 sec)
```

TABLE USERAVATAR

```
mysql> SELECT * from useravatar;
+-----------+
| Avatar    |
+-----------+
| user1.png |
+-----------+
1 row in set (0.00 sec)
```

DESCRIPTION OF USERAVATAR TABLE

```
mysql> DESC useravatar;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| Avatar | varchar(50) | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
1 row in set (0.00 sec)
```

TABLE USERPASS

```
mysql> SELECT * from userpass;
+-------------+----------+
| username    | password |
+-------------+----------+
| Legionnaire | password |
+-------------+----------+
1 row in set (0.00 sec)
```

DESCRIPTION OF USERPASS TABLE

```
mysql> DESC userpass;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| username | varchar(50) | YES  |     | NULL    |       |
| password | varchar(50) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
2 rows in set (0.01 sec)
```

TABLE WORDS

```
mysql> SELECT * from words;
+------+------+
| Srno | Word |
+------+------+
|   34 | bags |
|   35 | bake |
|   16 | bore |
|   29 | calm |
|   40 | cape |
|   26 | cows |
|   38 | crop |
|   22 | ears |
|   17 | flaw |
|    7 | grin |
|   21 | hare |
|   14 | have |
|   12 | home |
|   25 | into |
|   36 | jail |
|   32 | kite |
|   37 | labs |
|   18 | leak |
|    4 | lift |
|   27 | lock |
|   11 | main |
|   15 | mice |
|   20 | nail |
|   19 | node |
|    3 | plus |
|    5 | pure |
|   23 | rate |
|    6 | sail |
|   31 | sake |
|    8 | same |
|   33 | show |
|   13 | some |
|    2 | star |
|   39 | take |
|   28 | tame |
|    9 | wait |
|   24 | weak |
|   10 | wolf |
|    1 | work |
|   30 | year |
+------+------+
40 rows in set (0.08 sec)
```

DESCRIPTION OF WORDS TABLE

```
mysql> DESC words;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| Srno  | int(3)      | NO   | PRI | NULL    |       |
| Word  | varchar(20) | YES  | UNI | NULL    |       |
+-------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

Sr. No.	Frame	Variable Name	Component Property
01	Splash	jLabel1	Font: Century Gothic 36 Bold
			Foreground: Color-White
		loadingnum	Font: Cambria Math 24 Bold
			Foreground: Color-White
		loadingbar	Foreground: Color-Blue
		jLabel2	Icon: giphy (6).gif
		jLabel3	Icon: giphy (5).gif
		Splashlabel	Icon: Picture2.jpg
02	Login	Hellolabel	Font: Andalus 48 Bold
			Foreground: Color-White
		Userlabel	Icon: user12.jpg
		Usernamelabel	Font: Andalus 24 Bold
			Foreground: Color-White
		Passwordlabel	Font: Andalus 24 Bold
			Foreground: Color-White
		UserTF	Font: Andalus 24 Bold
		PassTF	Font: Andalus 24 Bold
			Foreground: Color-Red
		Submitbutton	Icon: thick_red (1).png
		Cancelbutton	Icon: x-mark-3-xxl (1).png
		Loginlabel	back1.jpg
03	Home	LegionnaireOSlabel	Font: Andalus 36 Bold
			Foreground: Color-White
		Menubutton	Icon: menu--1-.png
		GrizzlyBearbutton	Icon: icon_billing--1-.png
		Calculatorbutton	Icon: calculator-icon--2-.png
		Contactsbutton	Icon: contacts icon.png
		BullsandCowsbutton	Icon: game icon.png
		BullsandCowslabel	Font: Andalus 24 Bold
			Foreground: Color-White
		Contactslabel	Font: Andalus 24 Bold
			Foreground: Color-White
		GrizzlyBearlabel	Font: Andalus 24 Bold
			Foreground: Color-White

		Calculatorlabel	Font: Andalus 24 Bold
			Foreground: Color-White
		Homelabel	Icon: back3.jpg
		Homebutton	Icon: home-home-icon (2).png
		Backbutton	Icon: android-systemback(2).png
		Powerbutton	Icon: power-xxl (1).png
04	Apps	Calculatorapp	Icon: calculator-icon--2-.png
		Contactsapp	Icon: contacts icon.png
		BullsandCowsapp	Icon: game icon.png
		GrizzlyBearapp	Icon: icon_billing--1-.png
		NextPrimeNoapp	Icon: nextprime.png
		AboutUsapp	Icon: about-us-icon-15 (1).png
		Settingapp	Icon: settings-l-icon--1-.png
		BullsandCowslabel	Font: Andalus 24 Bold
			Foreground: Color-White
		Contactslabel	Font: Andalus 24 Bold
			Foreground: Color-White
		GrizzlyBearlabel	Font: Andalus 24 Bold
			Foreground: Color-White
		Calculatorlabel	Font: Andalus 24 Bold
			Foreground: Color-White
		NextPrimeNolabel	Font: Andalus 24 Bold
			Foreground: Color-White
		AboutUslabel	Font: Andalus 24 Bold
			Foreground: Color-White
		Settinglabel	Font: Andalus 24 Bold
			Foreground: Color-White
		MoreAppalabel	Font: Andalus 24 Bold
			Foreground: Color-White
		Appslabel	Icon: back3.jpg
		Homebutton	Icon: home-home-icon (2).png
		Backbutton	Icon: android-systemback(2).png
		Powerbutton	Icon: power-xxl (1).png
05	Bulls and Cows	jLabel1	Font: Century Gothic 36 Bold
			Foreground: Color-White
		JLabel(2-16)	Font: Century Gothic 18 Bold
			Foreground: Color-White
		jLabel17	Font: Tahoma 14 Bold

			Foreground: Color-Red
		inputTF	Font: Century Gothic 18 Bold
		jButton1	Font: Tahoma 18 Bold
		jButton2	Font: Century Gothic 24 Bold
		inputlabel(1-10)	Font: Century Gothic 18 Bold
			Foreground: Color-White
		bullslabel(1-10)	Font: Century Gothic 18 Bold
			Foreground: Color-White
		cowslabel(1-10)	Font: Century Gothic 18 Bold
			Foreground: Color-White
		Bullslabel	Icon: Game back2.jpg
		Homebutton	Icon: home-home-icon (2).png
		Backbutton	Icon: android-systemback(2).png
		Powerbutton	Icon: power-xxl (1).png
06	Enter No.	JLabel1	Font: Century Gothic 24 Bold
		numTF	Font: Tahoma 24 Bold
		jButton1	Font: Tahoma 24 Bold
07	How to Play	jLabel2	Font: Century Gothic 36 Bold
		JTextArea1	Font: Viner Hand ITC 24 Bold
		Gotit	Font: Matura MT Script 24 Bold
		Playlabel	Icon: contact back3.jpg
08	Contacts App	Searchbutton	Font: Andalus 24 Plain
			Foreground: Color-Blue
		Recent	Font: Andalus 24 Plain
			Foreground: Color-Blue
		Contacticon	Icon: contact user icon.png
		Contactnumber	Font: Andalus 24 Plain
			Foreground: Color-Blue
		Contactname	Font: Andalus 24 Plain
			Foreground: Color-Blue
		Callbackbutton	Font: Andalus 24 Bold
			Foreground: Color-White
		Searchtext	Font: Andalus 24 Plain
			Foreground: Color-Blue
		Addcontactlabel	Font: Andalus 30 Plain
		firstlabel	Font: Andalus 24 Plain
		lastlabel	Font: Andalus 24 Plain
		numberlabel	Font: Andalus 24 Plain

		Addcontactbutton	Font: Andalus 24 Plain
			Foreground: Color-Purple
		firsttext	Font: Andalus 24 Plain
		lasttext	Font: Andalus 24 Plain
		numbertext	Font: Andalus 24 Plain
		srlabel	Font: Andalus 18 Plain
		first	Font: Andalus 18 Plain
		last	Font: Andalus 18 Plain
		number	Font: Andalus 18 Plain
		firstTF	Font: Andalus 18 Plain
		lastTF	Font: Andalus 18 Plain
		numberTF	Font: Andalus 18 Plain
		Deletebutton	Font: Andalus 18 Plain
		Updatebutton	Font: Andalus 18 Plain
		Resetbutton	Font: Andalus 18 Plain
		Showdetailbutton	Font: Andalus 18 Plain
		Srenter	Font: Andalus 18 Plain
		Powerbutton	Icon: power-xxl (1).png
		Homebutton	Icon: home-home-icon (2).png
		Backbutton	Icon: android-systemback(2).png
		Contactbacklabel	Icon: contact back2.jpg
09	Contact Detail	Call	Font: Tahoma 24 Bold
		Delete	Font: Tahoma 24 Bold
		Close	Font: Tahoma 24 Bold
		Contacticon	Icon: contact user icon.png
		Contactname	Font: Andalus 24 Plain
			Foreground: Color-Blue
		Contactnumber	Font: Andalus 24 Plain
			Foreground: Color-Blue
10	Grizzly Bear Login	GrizzlyBearLogo	Icon: grizzy bear.png
		GrizzlyBear	Font: Andalus 36 Bold
			Foreground: Color-White
		Usernamelabel	Font: Andalus 24 Plain
		Passwordlabel	Font: Andalus 24 Plain
		Usernametext	Font: Andalus 24 Plain
		Loginbutton	Font: Andalus 24 Bold
		Backbutton	Font: Andalus 24 Bold
		Password	Font: Andalus 24 Plain

			Foreground: Color-Red
		GrizzlyBearlabel	Icon: app back1.jpg
		Homebutton	Icon: home-home-icon (2).png
		Backbutton 1	Icon: android-systemback(2).png
		Powerbutton	Icon: power-xxl (1).png
11	Grizzly Bear Menu	FoodNamelabel	Font: Century Gothic 24 Bold
		FoodTypelabel	Font: Century Gothic 24 Bold
		Pricelabel	Font: Century Gothic 24 Bold
		Quantitylabel	Font: Century Gothic 24 Bold
		Amountlabel	Font: Century Gothic 24 Bold
		FoodName	Font: Century Gothic 24 Bold
		FooType	Font: Century Gothic 24 Bold
		Price	Font: Century Gothic 24 Bold
		Quantity	Font: Century Gothic 24 Bold
		Amount	Font: Century Gothic 24 Bold
		Menultemlebel	Font: Century Gothic 24 Bold
		Drinks	Font: Century Gothic 15 Bold
		Sandwich	Font: Century Gothic 15 Bold
		Chapatti	Font: Century Gothic 15 Bold
		Rice	Font: Century Gothic 15 Bold
		SouthIndian	Font: Century Gothic 15 Bold
		ShakahariKhazana	Font: Century Gothic 15 Bold
		Soup	Font: Century Gothic 15 Bold
		Dal	Font: Century Gothic 15 Bold
		Western	Font: Century Gothic 15 Bold
		Breakfast	Font: Century Gothic 15 Bold
		AddBillbutton	Font: Century Gothic 24 Bold
		ShowBillbutton	Font: Century Gothic 24 Bold
		GrizzlyBearMenu	Font: Century Gothic 30 Bold
		menulabel	Icon: Wallpaper-white-21.jpg
		Homebutton	Icon: home-home-icon (2).png
		Backbutton	Icon: android-systemback(2).png
		Powerbutton	Icon: power-xxl (1).png
12	Grizzly Bear Bill	YourBill	Font: Century Gothic 30 Bold
		TotalAmountlabel	Font: Century Gothic 24 Bold
		TotalAmount	Font: Century Gothic 24 Bold
		PlaceOrderbutton	Font: Century Gothic 24 Bold
		GrizzlyBearbilllabel	Icon: Wallpaper-white-21.jpg

		Homebutton	Icon: home-home-icon (2).png
		Backbutton	Icon: android-systemback(2).png
		Powerbutton	Icon: power-xxl (1).png
13	Calculator App	Onebutton	Font: Tahoma 20 Bold
		Twobutton	Font: Tahoma 20 Bold
		Threebutton	Font: Tahoma 20 Bold
		Fourbutton	Font: Tahoma 20 Bold
		Fivebutton	Font: Tahoma 20 Bold
		Sixbutton	Font: Tahoma 20 Bold
		Sevenbutton	Font: Tahoma 20 Bold
		Eightbutton	Font: Tahoma 20 Bold
		Ninebutton	Font: Tahoma 20 Bold
		Zerobutton	Font: Tahoma 20 Bold
		x2button	Font: Tahoma 20 Bold
		Dividebutton	Font: Tahoma 24 Bold
		Multiplybutton	Font: Tahoma 24 Bold
		DotButton	Font: Tahoma 20 Bold
		ACbutton	Font: Tahoma 20 Bold
		Deletebutton	Cambria Math 48 Bold
		Subtractbutton	Font: Tahoma 36 Bold
		Addbutton	Font: Tahoma 24 Bold
		Equalbutton	Font: Tahoma 24 Bold
		Input	Font: Tahoma 18 Bold
		Logbutton	Font: Tahoma 20 Bold
		Cosbutton	Font: Tahoma 20 Bold
		Sinhbutton	Font: Tahoma 20 Bold
		Coshbutton	Font: Tahoma 20 Bold
		Tanbutton	Font: Tahoma 20 Bold
		xybutton	Font: Tahoma 20 Bold
		Sinbutton	Font: Tahoma 20 Bold
		Sqrtbutton	Font: Tahoma 20 Bold
		Tanhbutton	Font: Tahoma 20 Bold
		x3button	Font: Tahoma 20 Bold
		piebutton	Font: Felix Titling 20 Bold
		Plusminusbutton	Font: Cambria Math 18 Bold
		Cbrbutton	Font: Tahoma 20 Bold
		Roundbutton	Font: Tahoma 20 Bold
		Hexbutton	Font: Tahoma 20 Bold

		Binbutton	Font: Tahoma 20 Bold
		Modbutton	Font: Tahoma 20 Bold
		Inxbutton	Font: Tahoma 20 Bold
		Degbutton	Font: Tahoma 20 Bold
		Radbutton	Font: Tahoma 20 Bold
		Octalbutton	Font: Tahoma 20 Bold
		Calculatorlabel	Icon: back26.jpg
		Homebutton	Icon: home-home-icon (2).png
		Backbutton	Icon: android-systemback(2).png
14	Next Prime No.	Enternolabel	Font: Century Gothic 24 Bold
		Nextnolabel	Font: Century Gothic 24 Bold
		Nextno	Font: Century Gothic 24 Bold
		Submit	Font: Century Gothic 24 Bold
		Clear	Font: Century Gothic 24 Bold
		Enterno	Font: Century Gothic 24 Bold
		NextPrimeNolabel	Icon: Wallpaper-white-21.jpg
		Homebutton	Icon: home-home-icon (2).png
		Backbutton	Icon: android-systemback(2).png
		Powerbutton	Icon: power-xxl (1).png
15	Setting App	ChBackbutton	Font: Century Gothic 24 Bold
			Icon: displaysetting.png
		ChAvatarbutton	Font: Century Gothic 24 Bold
			Icon: Changeuser.png
		SecurSetbutton	Font: Century Gothic 24 Bold
			Icon: passicon.png
		Settinglabel	Icon: back20.jpg
		Homebutton	Icon: home-home-icon (2).png
		Backbutton	Icon: android-systemback(2).png
		Powerbutton	Icon: power-xxl (1).png
16	Change Background	homeback(1-18)	Icon: back(1-18) icon.jpg
		loginback(1-18)	Icon: back(1-18) icon.jpg
		Loginbacklabel	Font: Century Gothic 24 Bold
			Foreground: Color-Blue
		Homebacklabel	Font: Century Gothic 24 Bold
			Foreground: Color-Blue
		changebacklabel	Icon: back19.jpg
		Homelabel	Icon: home-home-icon (2).png
		Backlabel	Icon: android-systemback(2).png

		Powerlabel	Icon: power-xxl (1).png
17	Change Avatar	User(1-15)button	Icon: user(1-15).png
		changeavatarlabel	Icon: back19.jpg
		Homelabel	Icon: home-home-icon (2).png
		Backlabel	Icon: android-systemback(2).png
		Powerlabel	Icon: power-xxl (1).png
18	Security Setting	Securitylabel	Font: Century Gothic 48 Bold
		ChangeUserlabel	Font: Century Gothic 24 Bold
		ChangePasslabel	Font: Century Gothic 24 Bold
		CurUserlabel	Font: Century Gothic 24 Bold
		CurPasslabel	Font: Century Gothic 24 Bold
		NewUserlabel	Font: Century Gothic 24 Bold
		Curuserlabel	Font: Century Gothic 24 Bold
		Curpasslabel	Font: Century Gothic 24 Bold
		Newpasslabel	Font: Century Gothic 24 Bold
		CurUserleft	Font: Century Gothic 24 Bold
		NewUserleft	Font: Century Gothic 24 Bold
		NewPassright	Font: Century Gothic 24 Bold
		CurUserright	Font: Century Gothic 24 Bold
		Submitright	Font: Century Gothic 24 Bold
		Submitleft	Font: Century Gothic 24 Bold
		CurPassleft	Font: Century Gothic 24 Bold
			Foreground: Color-Red
		CurPassright	Font: Century Gothic 24 Bold
			Foreground: Color-Red
		SecurSetlabel	Icon: settingback3.png
		Homebutton	Icon: home-home-icon (2).png
		Backbutton	Icon: android-systemback(2).png
		Powerbutton	Icon: power-xxl (1).png
19	About Us	AboutUs	Font: Century Gothic 36 Bold
		AboutContent	Font: Andalus 24 Bold
		Createdbylabel	Font: Century Gothic 24 Bold
		Closebutton	Font: Century Gothic 24 Bold
		AboutUslabel	Icon: contactback4.jpg

BIBLIOGRAPHY

- Class XII NCERT Informatics Practices
- Class XII Informatics Practices-Sumita Arora
- Image Courtesy:
  - ✓Google Images
  - ✓Pexels
  - ✓Giphy
  - ✓Unsplash
  - ✓Zedge
- Youtube:
  - ✓http://youtu.be/1NbTjFx3ti8
  - ✓http://youtu.be/ZXToPivlR84
  - ✓http://youtu.be/oxT2G4gxsxc
- Hackerank, Code Chef & Udemy
- Help from Teachers and Friends