IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

WRITING A ERROR FREE CODE FOR ESTABLISHING DATA BASE DRIVEN APPLICATION WITH MYSQL DATABASE AND JAVA USING NET BEANS

[1]Divya Peddapalyam,[2]Dr.Madhu Malleshappa
[1]Msc.Biotechnology,[2]Msc.Phd-Post-Doc
[1,2]Department Of Biotechnology
[1,2]Garden City University, Bengaluru,Karnataka,India

Abstract: **This study is our academic project.The main reason for me to develop this programme is to store the analysed numerical data securely by providing user name and password Because in our academic project we studied and analysed raw data of the minimum organisms with IRGM protein in .If I want to continue my study then I have to secure the previous data,so for that we used this strategy. This study is the initiator to create a simple program example with MySQL Database and Java using NetBeans With Source Code .For developing this entire application we need NetBeans IDE,XAMPP,PhpMyAdmin,MySQL Database.I had written only the code that too error free.As I have used biological terms ,I encountered many errors.This page also provides evidence for error rectification also.**

Index Terms **– MySQL,Java,NetBeans,Source-Code,XAMPP,PhpMyAdmin**

I. SOURCE CODE-ERROR FREE:

```
package java_project_1_2;

public class Product {

private int id;
private String Name;
private float SolventAccessibility;
private float Antigenicity;
private float SolubilityUponOverExpression;
private float PredictedDisorderedProbability;
private float PredictedOrderedProbability;
private int ExposedResidues;
private int BuriedResidues;
private float AlphaHelicalTransmembraneProtein;
private float BetaBarrelTransmembraneProtein;
private float CompositionOfAminoAcids;
private float AverageMassOfAminoAcids;
private float IsoElectricPoint;
private float  RateOfTransmittedLightAssumingCystine;
private float AbsorbanceForMediumAsumingCystineResidues;
private float RateOfTransmittedLightWithOutAssumingCystine;
private float AbsorbanceForMediumWithOutAssumingCystine;
```

```
private float NonReliabilityOfProtein;
private float RelativeVolumeOccupiedByAliphaticSideChains;
private float ProbabilityOfHyrophilicAndHydrophobicProperties;
private String addDate;
private byte[] picture;
```

```
public Product(int pid, String pName, float pSolventAccessibility, float pAntigenicity, float
pSolubilityUponOverExpression, float pPredictedDisorderedProbability, float pPredictedOrderedProbability, int
pExposedResidues, int pBuriedResidues, float pAlphaHelicalTransmembraneProtein, float pBetaBarrelTransmembraneProtein,
float pCompositionOfAminoAcids, float pAverageMassOfAminoacids, float pIsoElectricPoint,float
pRateOfTransmittedLightAssumingCystine, float pAbsorbanceForMediumAssumingCystineResidues, float
pRateOfTransmittedLightWithOutAssumingCystine, float pAbsorbanceForMediumWithOutAssumingCystine, float
pNonReliabilityOfProtein, float pRelativeVolumeOccupiedByAliphaticSideChains, float
pProbabilityOfHydroPhilicAndHydrophobicProperties,String pAddDate, byte[] pimg)
{
   this.id = pid;
   this.name = pName;
   this.SolventAccessibility = pSolventAccessibility;
   this.Antigenicity = pAntigencity;
   this.SolubilityUponOverExpression = pSolubilityUponOverExpression;
   this.PredictedDisorderedProbability = pPredictedDisorderedProbability;
   this.PredictedOrderedProbability = pPredictedOrderedProbability;
   this.ExposedResidues = pExposedResidues;
   this.BuriedResidues = pBuriedResidues;
   this.AlphaHelicalTransmembraneProtein = pAlphaHelicalTransmembraneProtein;
   this.BetaBarrelTransmembraneProtein = pBetaBarrelTransmembraneProtein;
   this.CompositionOfAminoacids = pCompositionOfAminoAcids;
   this.AverageMassOfAminoAcids = pAverageMassOfAminoAcids;
   this.IsoElectricPoint = pIsoElectricPoint;
   this.RateOfTransmittedLightAssumingCystine = pRateOfTransmittedLightAssumingCystine;
   this.AbsorbanceForMediumAssumingCystine = pAbsorbanceForMediumAssumingCystine;
   this.RateOfTransmittedLightWithOutAssumingCystine = pRateOfTransmittedLightWithOutAssumingCystine;
   this.AbsorbanceForMediumWithOutAssumingCystine = pAbsorbanceForMediumWithOutAssumingCystine;
   this.NonReliabilityOfProtein =  pNonReliabilityOfProtein;
   this.RelativeVolumeOccupiedByAliphaticSideChains =pRelativeVolumeOccupiedByAliphaticSideChains;
   this.ProbabilityOfHydrophilicAndHydrophobicProperties =pProbabilityOfHydrophilicAndHyrophobicProperties;
   this.addDate = pAddDate;
   this.picture = pimg;
}

public int getId()
{
   return id;
}

public String getName()
{
   return name;
}

public float getSolventAccessibility()
{
   return SolventAccessibility;
}
public float getAntigenicity( )
{
return Antigenicity;
}
public float getSolubilityUponOverExpression( )
{
 return SolubilityUponOverExpression;
```

```
}
public float getPredictedDisorderedProbability( )
{
return PredictedDisorderedProbability;
}
public float getPredictedOrderedProbability( )
{
return PredictedOrderedProbability;
}
public int getExposedResidues( )
{
return ExposedResidues;
}
public int getBuriedResidues( )
{
return BuriedResidues;
}
public float getAlphaHelicalTransmembraneProtein( )
{
return AlphaHelicalTransmembraneProtein;
}
public float getBetaBarrelTransmembraneProtein( )
{
return BetaBarrelTransmembraneProtein;
}
public float getCompositionOfAminoAcids( )
{
return CompositionOfAminoacids;
}
public float getAverageMassOfAminoacids( )
{
return AverageMassOfAminoAcids;
}
public float getIsoElectricPoint( )
{
return IsoElectricPoint;
}
public float getRateOfTransmittedLightAssumingCystine( )
{
return RateOfTransmittedLightAssumingCystine;
}
public float getAbsorbanceForMediumAssumingCystine( )
{
return AbsorbanceForMediumAssumingCystine;
}
public float getRateOfTransmittedLightWithOutAssumingCystine( )
{
return RateOfTransmittedLightWithOutAssumingCystine;
}
public float getAbsorbanceForMediumWithOutAssumingCystine( )
{
return AbsorbanceForMediumWithOutAssumingCystine;
}
public float getNonReliabilityOfProtein( )
{
return NonReliabilityOfProtein;
}
public float getRelativeVolumeOccupiedByAliphaticSideChains( )
{
return RelativeVolumeOccupiedByAiphaticSideChains;
}
public float getProbabilityOfHydrophilicAndHydrophobicProperties( )
{
```

```
return ProbabilityOfHydrophilicAndHydrophobicProperties;
}
 public String getAddDate()
{
 return addDate;
 }

 public byte[] getImage()
 {
 return picture;
 }
```

EndProduct Class

```
package java_project_1_2;

import java.awt.Image;
import java.io.File;
import java.io.FileInputStream;
import java.io.InputStream;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.imageio.ImageIO;
import javax.swing.ImageIcon;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
import javax.swing.filechooser.FileNameExtensionFilter;
import javax.swing.table.DefaultTableModel;




}

/**
 *
 * @DIVYA PEDDAPALYAM
 */
public class Main_Window extends javax.swing.JFrame {

 /**
 * Creates new form Main_Window
 */
 public Main_Window() {

 initComponents();
 Show_Products_In_JTable();
   }

 String ImgPath = null;
 int pos = 0;

 // Function To Connect To MySQL Database
```

```
public Connection getConnection()
{
 Connection con = null;

 try {
 con = DriverManager.getConnection("jdbc:mysql://localhost/products_db2","root","");
 return con;
 } catch (SQLException ex) {
        Logger.getLogger(Main_Window.class.getName()).log(Level.SEVERE, null, ex);
        return null;
    }
  }

  // Check Input Fields
  public boolean checkInputs()
  {
     if(
          txt_name.getText() == null
|| txt_SolventAccessibility.getText() == null
|| txt_Antigenicity.getText() == null
||   txt_SolubilityUponOverExpression.getText()    == null
||    txt_PredictedDisorderedProbability.getText()  ==null
|| txt_PredictedOrderedProbability.getText() == null
|| txt_ExposedResidues.getText() == null
|| txt_BuriedResidues.getText() == null
|| txt_AlphaHelicalTransmembraneProtein.getText() == null
|| txt_BetaBarrelTransmembraneProtein.getText() == null
|| txt_CompositionOfAminoAcids.getText()    ==null
||  txt_AverageMassOfAminoacids.getText()==null
|| txt_IsoElectricPoint.getText() == null
|| txt_RateOfTransmittedLightAssumingCystine.getText() == null
|| txt_AbsorbanceForMediumAssumingCystine.getText() == null
|| txt_RateOfTransmittedLightWithOutAssumingCystine.getText() == null
|| txt_NonReliabilityOfProtein.getText() == null
|| txt_RelativeVolumeOccupiedByAliphaticSideChains.getText() == null
|| txt_ProbabilityOfHydrophilicAndHydrophobicProperties.getText() == null
|| txt_AddDate.getDate() == null
        ){
          return false;
       }
       else{
          try{
             Float.parseFloat(txt_SolventAccessibility.getText());
Float.parseFloat(txt_Antigenicity.getText());
Float.parseFloat(txt_SolubilityUponOverExpression.getText());
Float.parseFloat(txt_PredictedDisorderedProbability.getText());
Float.parseFloat(txt_PredictedOrderedProbability.getText());
Int.parseInt(txt_ExposedResidues.getText());
Int.parseInt(txt_BuriedResidues.getText());
Float.parseFloat(txt_AlphaHelicalTransmembraneProtein.getText());
Float.parseFloat(txt_BetaBarrelTransmembraneProtein.getText());
Float.parseFloat(txt_CompositionOfAminoAcids.getText());
Float.parseFloat(txt_AverageMassOfAminoAcids.getText());
Float.parseFloat(txt_IsoElectricPoint.getText());
Float.parseFloat(txt_RateOfTransmittedLightAssumingCystine.getText());
Float.parseFloat(txt_AbsorbanceForMediumAssumingCystine.getText());
Float.parseFloat(txt_RateOfTransmittedLightWithOutAssumingCystine.getText());
Float.parseFloat(txt_AbsorbanceForMediumWithOutAssumingCystine.getText());
Float.parseFloat(txt_NonReliabilityOfProtein.getText());
Float.parseFloat(txt_RelativeVolumeOccupiedByAliphaticSideChains.getText());
Float.parseFloat(txt_ProbabilityOfHydrophilicAndHydrophobicProperties.getText());
```

```
            return true;
           }catch(Exception ex)
           {
              return false;
           }
        }
    }


     // Function To Resize The Image To Fit Into JLabel
      public ImageIcon ResizeImage(String imagePath, byte[] pic)
      {
       ImageIcon myImage = null;

       if(imagePath != null)
       {
          myImage = new ImageIcon(imagePath);
       }else{
          myImage = new ImageIcon(pic);
       }

       Imageimg = myImage.getImage();
       Image img2 = img.getScaledInstance(lbl_image.getWidth(), lbl_image.getHeight(), Image.SCALE_SMOOTH);
       ImageIcon image = new ImageIcon(img2);
       return image;

        }

    // Display Data In JTable:
    //      1 - Fill ArrayList With The Data

    public ArrayList<Product> getProductList()
    {
          ArrayList<Product> productList = new ArrayList<Product>();
          Connection con = getConnection();
          String query = "SELECT * FROM products";

          Statement st;
          ResultSet rs;

       try {

          st = con.createStatement();
          rs = st.executeQuery(query);
          Product product;

          while(rs.next())
          {

product=newProduct(rs.getInt("id"),rs.getString("name"),Float.parseFloat(rs.getString("price")),rs.Float.parseFloat(rs.getString("So
lventAccessibility")),rs.Float.parseFloat(rs.getString("Antigenicity")),Float.parseFloat(rs.getString("SolubilityUponOverExpression
")),Float.parseFloat(rs.getString("PredictedDisorderedProbability")),Float.parseFloat(rs.getString("PredictedOrderedProbability")),(
rs.getInt("ExposedResidues")),(rs.getint("buriedResidues")),Float.parseFloat(rs.getString("AlphaHelicalTransmembraneProtein")),
Float.parseFloat(rs.getString("BetaBarrelTransmembraneProtein")),Float.parseFloat(rs.getString("CompositionOfAminoAcids")),F
loat.parseFloat(rs.getString("AverageMassOfAminoAcids")),Float.parseFloat(rs.getString("IsoelectricPoint")),Float.parseFloat(rs.g
etString("RateOfTransmittedLightAssumingCystine")),Float.parseFloat(rs.getString("AbsorbanceForMediumAssumingCystine")),
Float.parseFloat(rs.getString("RateOfTransmittedLightWthOutAssumingCystine")),Float.parseFloat(rs.getString("AbsorbanceFor
MediumWithOutAssumingCystine")),Float.parseFloat(rs.getString("NonReliabilityOfProtein")),Float.parsefloat(rs.getString("Relat
iveVolumeOccupiedByAliphaticSideChains")),Float.parsefloat(rs.getString("ProbabilityOfHydrophilicAndHydrophobicProperties"
)),rs.getString("add_date"),rs.getBytes("image"));
                productList.add(product);
```

```
}

} catch (SQLException ex) {
Logger.getLogger(Main_Window.class.getName()).log(Level.SEVERE, null, ex);
}

return productList;

 }



  //     2 - Populate The JTable

  public void Show_Products_In_JTable()
  {
  ArrayList<Product> list = getProductList();
 DefaultTableModel model = (DefaultTableModel)JTable_Products.getModel();
 // clear jtable content
 model.setRowCount(0);
 Object[] row = new Object[4];
 for(int i = 0; i < list.size(); i++)
     {
        row[0] = list.get(i).getId();
        row[1] = list.get(i).getName();
        row[2] = list.get(i).getSolventAccessibility();
        row[3] = list.get(i).getAntigenicity();
        row[4] = list.get(i).getSolubilityUponOverExpression( );
        row[5] = list.get(i).getPredictedDisorderedProbability( );
        row[6] = list.get(i).getPredictedOrderedProbability( );
        row[7] = list.get(i).getExposedResidues( );
        row[8] = list.get(i).getBuriedResidues( );
        row[9] = list.get(i).getAlphaHelicalTransmembraneProtein( );
        row[10] = list.get(i).getBetaBarreltransmembraneProtein( );
        row[11] = list.get(i).getCompositionOfAminoAcids( );
        row[12] = list.get(i).getAverageMassOfAminoAcids( );
        row[13] = list.get(i).getIsoElectricPoint( );
        row[14] = list.get(i).getRateOfTransmittedLightAssumingCystine( );
        row[15] = list.get(i).getAbsorbanceForMediumAssumingCystine( );
        row[16] = list.get(i).getRateOfTransmittedLightWithOutAssumingCystine( );
        row[17] = list.get(i).getAbsorbanceForMediumWithOutAssumingCystine( );
        row[18] = list.get(i).getNonReliabilityOfProtein( );
        row[19] = list.get(i).getRelativeVolumeOccupiedByAliphaticSideChains( );
        row[20] = list.get(i).ProbabilityOfHydrophilicAndHydrophobicProperties( );

        model.addRow(row);
     }

  }

  // Show Data In Inputs
  public void ShowItem(int index)
  {
        txt_id.setText(Integer.toString(getProductList().get(index).getId()));
        txt_name.setText(getProductList().get(index).getName());
        txt_SolventAccessibility.setText(Float.toString(getProductList().get(index).getSolventAccessibility()));

        txt_Antigenicity.setText(Float.toString(getProductList().get(index).getAntigenicity()));

        txt_SolubilityUponOverExpression.setText(Float.toString(getProductList().get(index).getSolubilityUponOverExpression()));

        txt_PredictedDisorderedProbability.setText(Float.toString(getProductList().get(index).getPredictedDisorderedProbability())

        );
        txt_PredictedOrderedProbability.setText(Float.toString(getProductList().get(index).getPredictedOrderedProbability()));
```

```
txt_AlphaHelicalTransmembraneProtein.setText(Float.toString(getProductList().get(index).getAlphaHelicalTransmembraneProtein
()));txt_BetaBarrelTransmembraneProtein.setText(Float.toString(getProductList().get(index).getBetaBarrelTransmembraneProtein())
); txt_CompositionOfAminoAcids.setText(Float.toString(getProductList().get(index).getCompositionOfAminoAcids()));
txt_AverageMassOfAminoAcids.setText(Float.toString(getProductList().get(index).getAverageMassOfAminoAcids()));
txt_IsoElectricPoint.setText(Float.toString(getProductList().get(index).getIsoElectricPoint()));txt_RateOfTransmittedLightAssumingC
ystine.setText(Float.toString(getProductList().get(index).getRateOfTransmittedLightAssu mingCystine()));

txt_AbsorbanceForMediumAssumingCystine.setText(Float.toString(getProductList().get(index).getAbsorbanceForMediumAssumi
ngCystine()));txt_RateOfTransmittedLightWithOutAssumingCystine.setText(Float.toString(getProductList().get(index).getRateOfTra
nsmittedLi
ghtWithOutAssumingCystine()));txt_AbsorbanceForMediumWithOutAssumingCystine.setText(Float.toString(getProductList().get(ind
ex).getAbsorbanceForMediu mWithOutAssumingCystine()));
txt_NonReliabilityOfProtein.setText(Float.toString(getProductList().get(index).getNonReliabilityOfProtein()));txt_RealtiveVolumeOc
cupiedByAliphaticSideChains.setText(Float.toString(getProductList().get(index).getRelativeVolumeOccupi
edByAliphaticSideChains()));txt_ProbabilityOfHydrophilicAndHydrophobicProperties.setText(Float.toString(getProductList().get(in
dex).getProbabilityOfHydro philicAndHydrophobicProperties()));

            try {
            Date addDate = null;
            addDate = new SimpleDateFormat("yyyy-MM-dd").parse((String)getProductList().get(index).getAddDate());
            txt_AddDate.setDate(addDate);
             } catch (ParseException ex) {
            Logger.getLogger(Main_Window.class.getName()).log(Level.SEVERE, null, ex);
             }

            lbl_image.setIcon(ResizeImage(null, getProductList().get(index).getImage()));
            }

    /**
     * This method is called from within the constructor to initialize theform.
     * WARNING: Do NOT modify this code. The content of thismethodisalways
     * regenerated by the FormEditor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        jLabel5 = new javax.swing.JLabel();
        .
        jLabel22=new javax swing JLabel();
        txt_name = new javax.swing.JTextField();
        txt_id = new javax.swing.JTextField();
         txt_SolventAccessibility= new javax.swing.JTextField();
        txt_Antigenicity = new javax.swing.JTextField();
        txt_SolubilityUponOverExpression = new javax.swing.JTextField();
        txt_PredictedDisorderedProbability= new javax.swing.JTextField();
        txt_PredictedOrderedProbability = new javax.swing.JTextField();
        txt_ExposedResidues = new javax.swing.JTextField();
        txt_BuriedResidues = new javax.swing.JTextField();
        txt_AlphaHelicalTransmembraneProtein = new
        javax.swing.JTextField();
```

```
txt_BetaBarrelTransmembraneProtein = new javax.swing.JTextField();
txt_CompositionOfAminoAcids = new javax.swing.JTextField();
txt_AverageMassOfAminoAcids = new javax.swing.JTextField();
txt_IsoElectricPoint= new javax.swing.JTextField();
txt_RateOfTransmittedLightAssumingCystine= new javax.swing.JTextField();
txt_AbsorbanceForMediumAssumingCystine= new javax.swing.JTextField();
txt_RateOfTransmittedLightWithOutAssumingCystine= new javax.swing.JTextField();
txt_AbsorbanceForMediumWithOutAssumingCystine= new javax.swing.JTextField();
txt_NonReliabilityOfProtein= new javax.swing.JTextField();
        txt_RelativeVolumeOccupiedByAliphaticSideChains = new javax.swing.JTextField();
        txt_ProbabilityOfHydrophilicAndHydrophobicProperties = new javax.swing.JTextField();
        txt_AddDate = new com.toedter.calendar.JDateChooser();
        lbl_image = new javax.swing.JLabel();
        jScrollPane1 = new javax.swing.JScrollPane();
        JTable_Products = new javax.swing.JTable();
        Btn_Choose_Image = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
        jButton3 = new javax.swing.JButton();
        Btn_Insert = new javax.swing.JButton();
        Btn_First = new javax.swing.JButton();
        Btn_Previous = new javax.swing.JButton();
        Btn_Last = new javax.swing.JButton();
        Btn_Next = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jPanel1.setBackground(new java.awt.Color(255, 255, 204));

        jLabel1.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
        jLabel1.setText("id:");

        jLabel2.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
        jLabel2.setText("name:");

        jLabel3.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
        jLabel3.setText("SolventAccessibility:");

        jLabel4.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
        jLabel4.setText("Antigenicity:");

        jLabel5.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
        jLabel5.setText("SolubilityUponOverExpression:");

        jLabel5.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
        jLabel5.setText("PredictedDisorderedProbability:");
        jLabel5.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
        jLabel5.setText("PredictedOrderedProbability:");

        jLabel5.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
        jLabel5.setText("ExposedResidues:");
        jLabel5.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
        jLabel5.setText("BuriedResidues:");
        jLabel5.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
        jLabel5.setText("AlphaHelicalTransmembraneProtein:");

        jLabel5.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
        jLabel5.setText("BetaBarrelTransmembraneProtein:");


        jLabel5.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
        jLabel5.setText("CompositionOfAminoacids:");

        jLabel5.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
```

```
jLabel5.setText("AverageMassOfAminoAcids:");
jLabel5.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
jLabel5.setText("IsoElectricPoint:");

jLabel5.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
jLabel5.setText("RateOfTransmittedLightAssumingCystine:");

jLabel5.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
jLabel5.setText("AbsorbanceForMediumAssumingCystine:");
jLabel5.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
jLabel5.setText("RateOfTransmittedLightWithOutAssumingCystine:");

jLabel5.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
jLabel5.setText("AbsorbanceForMediumWithOutAssumingCystine:");


jLabel5.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
jLabel5.setText("NonReliabilityOfProtein:");

jLabel5.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
jLabel5.setText("RelativeVolumeOccupiedByAliphaticSideChains:");
jLabel5.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
jLabel5.setText("ProbabilityOfHydrophilicAndHydrophobicProperties:");


jLabel5.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
jLabel5.setText(“Add date:");


txt_name.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
txt_name.setPreferredSize(new java.awt.Dimension(59, 50));

txt_id.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
txt_id.setEnabled(false);
txt_id.setPreferredSize(new java.awt.Dimension(59, 50));

txt_price.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
txt_price.setPreferredSize(new java.awt.Dimension(59, 50));

txt_AddDate.setDateFormatString("yyyy-MM-dd");
txt_AddDate.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N

lbl_image.setBackground(new java.awt.Color(204, 255, 255));
lbl_image.setOpaque(true);

JTable_Products.setModel(new javax.swing.table.DefaultTableModel(
   new Object [][] {

   },
   new String [] {
      "ID", "Name",
"SolventAccessibility",”Antigenicity”,”SolubilityUponOverExpression”,”PredictedDisorderedProbability”,”PredictedOrderedProbability”,ExposedResidues”,”BuriedResidues”,”AlphaHelicalTransmembraneProtein”,”BetaBarrelTransmembraneProtein”,”CompositionOfAminoAcids”,”AverageMassOfAminoAcids”,”IsoElectricPoint”,”RateOfTransmittedLightAssumingCystine”,AbsorbanceForMediumAssumingCystineResidues”,”RateOfTransmittedLightWithOutAssumingCystine”,”AbsorbanceForMediumWithOutAssumingCystine”,”NonReliabilityOfProtein”,”RelativeVolumeOccupiedByAliphaticSideChains”,”ProbabilityOfHydrophilicAndHydrophobicProperties”, "Add Date"
   }
));
JTable_Products.addMouseListener(new java.awt.event.MouseAdapter() {
   public void mouseClicked(java.awt.event.MouseEvent evt) {
      JTable_ProductsMouseClicked(evt);
```

```
    }
});
jScrollPane1.setViewportView(JTable_Products);

Btn_Choose_Image.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
Btn_Choose_Image.setText("Choose Image");
Btn_Choose_Image.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        Btn_Choose_ImageActionPerformed(evt);
    }
});

jButton2.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jButton2.setIcon(new javax.swing.ImageIcon(getClass().getResource("@AUTHOR-DIVYA PEDDAPALYAM"))); //
NOI18N
jButton2.setText("Update");
jButton2.setIconTextGap(15);
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});

jButton3.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jButton3.setIcon(new javax.swing.ImageIcon(getClass().getResource("@AUTHOR-DIVYA PEDDAPALYAM"))); //
NOI18N
jButton3.setText("Delete");
jButton3.setIconTextGap(15);
jButton3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton3ActionPerformed(evt);
    }
});

Btn_Insert.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
Btn_Insert.setIcon(new javax.swing.ImageIcon(getClass().getResource("@AUTHOR-DIVYA PEDDAPALYAM"))); //
NOI18N
Btn_Insert.setText("Insert");
Btn_Insert.setIconTextGap(15);
Btn_Insert.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        Btn_InsertActionPerformed(evt);
    }
});

Btn_First.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
Btn_First.setIcon(new javax.swing.ImageIcon(getClass().getResource("@AUTHOR-DIVYA PEDDAPALYAM"))); //
NOI18N
Btn_First.setText("First");
Btn_First.setIconTextGap(15);
Btn_First.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        Btn_FirstActionPerformed(evt);
    }
});

Btn_Previous.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
Btn_Previous.setIcon(new javax.swing.ImageIcon(getClass().getResource("@AUTHOR-DIVYA PEDDAPALYAM")));
// NOI18N
Btn_Previous.setText("Previous");
Btn_Previous.setIconTextGap(15);
Btn_Previous.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
            Btn_PreviousActionPerformed(evt);
        }
    });

    Btn_Last.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
    Btn_Last.setIcon(new javax.swing.ImageIcon(getClass().getResource("@AUTHOR-DIVYA PEDDAPALYAM"))); //
    NOI18N
    Btn_Last.setText("Last");
    Btn_Last.setIconTextGap(15);
    Btn_Last.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            Btn_LastActionPerformed(evt);
        }
    });

    Btn_Next.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
    Btn_Next.setIcon(new javax.swing.ImageIcon(getClass().getResource("@AUTHOR-DIVYA PEDDAPALYAM"))); //
    NOI18N
    Btn_Next.setText("Next");
    Btn_Next.setIconTextGap(15);
    Btn_Next.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            Btn_NextActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
    jPanel1.setLayout(jPanel1Layout);
    jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
    .addGap(33, 33, 33)
    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                .addComponent(jLabel1)
                .addComponent(jLabel2)
                .addComponent(jLabel3)
                .addComponent(jLabel4)
                .addComponent(jLabel5)
                .
                .addComponent(jLabel22))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                .addComponent(Btn_Choose_Image, javax.swing.GroupLayout.DEFAULT_SIZE, 227, Short.MAX_VALUE)
                .addComponent(txt_AddDate, javax.swing.GroupLayout.PREFERRED_SIZE, 125,
                 javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(txt_id, javax.swing.GroupLayout.PREFERRED_SIZE, 109,
                 javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(txt_name, javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .
                .(TILL PROBABILITY OF HYDROPHILIC AND HYDROPHOBIC PROPERTIES)

                addComponent(lbl_image, javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
  javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap())
            .addGroup(jPanel1Layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(Btn_Insert)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(jButton2)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(jButton3)
```

```
                .addGap(45, 45, 45)
                .addComponent(Btn_First)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(Btn_Next)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(Btn_Previous)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(Btn_Last)
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            );
                jPanel1Layout.setVerticalGroup(
               jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
               .addGroup(jPanel1Layout.createSequentialGroup()
               .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
               .addGroup(jPanel1Layout.createSequentialGroup()
               .addGap(31, 31, 31)
               .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
               .addComponent(txt_id, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
                javax.swing.GroupLayout.PREFERRED_SIZE)
               .addComponent(jLabel1))
               .addGap(9, 9, 9)
               .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
               .addComponent(txt_name, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
                 javax.swing.GroupLayout.PREFERRED_SIZE)
               .addComponent(jLabel2))
               .addGap(14, 14, 14)
              .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                          .
                          .

(TILL 22 )
   .addComponent(jLabel4)) .addGroup(jPanel1Layout.createSequentialGroup()
   .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
   .addComponent(txt_AddDate, javax.swing.GroupLayout.PREFERRED_SIZE, 41,
    javax.swing.GroupLayout.PREFERRED_SIZE)))
                          .addGap(18, 18, 18)
                          .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                          .addComponent(jLabel5)
                          .addComponent(lbl_image, javax.swing.GroupLayout.PREFERRED_SIZE,
                           167, javax.swing.GroupLayout.PREFERRED_SIZE))
                           .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                          .addComponent(Btn_Choose_Image,
                           javax.swing.GroupLayout.PREFERRED_SIZE, 36,
                           javax.swing.GroupLayout.PREFERRED_SIZE))
                          .addGroup(jPanel1Layout.createSequentialGroup()
                           .addContainerGap()
                           .addComponent(jScrollPane1,
                            javax.swing.GroupLayout.PREFERRED_SIZE,
                            javax.swing.GroupLayout.DEFAULT_SIZE,
                            javax.swing.GroupLayout.PREFERRED_SIZE)))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 54, Short.MAX_VALUE)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
 .addComponent(Btn_Insert, javax.swing.GroupLayout.PREFERRED_SIZE, 40,

                javax.swing.GroupLayout.PREFERRED_SIZE)

                . addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
                  javax.swing.GroupLayout.PREFERRED_SIZE)
                  .addComponent(jButton3, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
                  javax.swing.GroupLayout.PREFERRED_SIZE)
                  .addComponent(Btn_First, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
                  javax.swing.GroupLayout.PREFERRED_SIZE)
                  .addComponent(Btn_Next, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
                  javax.swing.GroupLayout.PREFERRED_SIZE)
                  .addComponent(Btn_Previous, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
                  javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
                .addComponent(Btn_Last, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
                javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(31, 31, 31))
        );

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
            Short.MAX_VALUE)
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
             Short.MAX_VALUE)
        );

        pack();
    }// </editor-fold>
    // Button Browse Image From Your Computer
    private void Btn_Choose_ImageActionPerformed(java.awt.event.ActionEvent evt) {

        JFileChooser file = new JFileChooser();
        file.setCurrentDirectory(new File(System.getProperty("user.home")));

        FileNameExtensionFilter filter = new FileNameExtensionFilter("*.images", "jpg","png");
        file.addChoosableFileFilter(filter);
        int result = file.showSaveDialog(null);
        if(result == JFileChooser.APPROVE_OPTION)
        {
            File selectedFile = file.getSelectedFile();
            String path = selectedFile.getAbsolutePath();
            lbl_image.setIcon(ResizeImage(path, null));
            ImgPath = path;
        }

       else{
       System.out.println("No File Selected");
        }

        }



    // Button Insert Data Into MySQL Database
    // 1 - Check If The imgPath Is Not Null And The Inputs Are Not Empty
    // 2 - Insert The Data
    private void Btn_InsertActionPerformed(java.awt.event.ActionEvent evt)
        { if(checkInputs() && ImgPath != null)
        {
            try {
                Connection con = getConnection();
                PreparedStatement ps =con.prepareStatement("INSERT INTO
products(name,SolventAccessibility,Antigenicity,SolubilityUponOverExpression,PredictedDisorderedProbability,PredictedOrder
edProbability,AlphaHelicalTransMembraneProtein,BetaBarrelTransmembraneProtein,CompositionOfAminoAcids,AverageMass
OfAminoAcids,IsoElectricPoint,RateOfTransmittedLightAssumingCystine,AbsorbanceForMediumAssumingCystine,RateOfTra
nsmittedLightWithOutAssumingCystine,AbsorbanceForMediumWithOutAssumingCystine,NonReliabilityOfProtein,RelativeVol
umeOccupiedByAliphaticSideChains,ProbabilityOfHydrophilicAndHydrophobicProperties,add_date,image)"
                        + "values(?,?,?,?) ");
                ps.setString(1, txt_name.getText());
                ps.setString(2, txt_SolventAccessibility.getText());
                ps.setString(2, txt_Antigenicity.getText());
```

```
ps.setString(2, txt_SolubilityUponOverExpression.getText());
ps.setString(2, txt_PredictedDisorderedProbability.getText());
ps.setString(2, txt_PredictedOrderedProbability.getText());
ps.setString(2, txt_AlphaHelicalTransmembraneProtein.getText());
ps.setString(2, txt_BetaBarrelTransmembraneProtein.getText());
ps.setString(2, txt_CompositionOfAminoAcids.getText());
ps.setString(2, txt_AverageMassOfAminoAcids.getText());
ps.setString(2, txt_IsoElectricPoint.getText());
ps.setString(2, txt_RateOftransmittedLightAssumingCystine.getText());
ps.setString(2, txt_AbsorbanceForMediumAssumingCystine.getText());
ps.setString(2, txt_RateOfTransmittedLightWithOutAssumingCystine.getText());
ps.setString(2, txt_AbsorbanceForMediumWithOutAssumingCystine.getText());
ps.setString(2, txt_NonReliabilityOfProtein.getText());
ps.setString(2, txt_RelativeVolumeOccupiedByAliphaticSideChains.getText());
ps.setString(2, txt_ProbabilityOfHydrophilicAndHydrophobicProperties.getText());


SimpleDateFormat dateFormat = newSimpleDateFormat("yyyy-MM-dd");
String addDate =    dateFormat.format(txt_AddDate.getDate());
ps.setString(3, addDate);

InputStream img = new FileInputStream(new File(ImgPath));
ps.setBlob(4, img);
ps.executeUpdate();
Show_Products_In_JTable();

JOptionPane.showMessageDialog(null, "Data Inserted");
 } catch (Exception ex) {
 JOptionPane.showMessageDialog(null, ex.getMessage());
 }
 }else{
JOptionPane.showMessageDialog(null, "One Or More Field Are Empty");
}

// only for test
System.out.println("Name => "+txt_name.getText());
System.out.println("SolventAccessibility => "+txt_SolventAccessibility.getText());
 System.out.println("Antigenicity => "+txt_Antigenicity.getText());
System.out.println("SolubilityUponOverExpression => "+txt_SolubilityUponOverExpression.getText());
System.out.println("PredictedDisorderedProbability => "+txt_PredictedDisorderedProbability.getText());
System.out.println("PredictedOrderedProbability => "+txt_PredictedOrderedProbability.getText());
System.out.println("AlphaHelicalTransmembraneProtein => "+txt_AlphaHelicalTransmembraneProtein.getText());
System.out.println("BetaBarrelTransmembraneProtein => "+txt_BetaBarrelTransmembraneProtein.getText());
System.out.println("CompositionOfAminoAcids=> "+txt_CompositionOfAminoAcids.getText());
System.out.println("AverageMassOfAminoAcids => "+txt_AverageMassOfAminoAcids.getText());
System.out.println("IsoElectricPoint => "+txt_IsoElectricPoint.getText());
System.out.println("RateOfTransmittedLightAssumingCystine =>
"+txt_RateOfTransmittedLightAssumingCystine.getText());
System.out.println("AbsorbanceForMediumAssumingCystine =>
"+txt_AbsorbanceForMediumAssumingCystine.getText());
System.out.println("RateOfTransmittedLightWithOutAssumingCystine =>
"+txt_RateOfTransmittedLightWithOutAssumingCystine.getText());

System.out.println("AbsorbanceForMediumWithOutAssumingCystine=>"+txt_AbsorbanceForMediumWithOutAssumingCystine.
getText());System.out.println("NonReliabilityOfProtein =>
"+txt_NonReliabilityOfProtein.getText());System.out.println("RelativeVolumeOccupiedByAliphaticSideChains"=>+txt_RelativeVol
umeOccupiedByAliphaticSideChains.ge tText());
System.out.println("ProbabilityOfHydrophilicAndHydrophobicProperties=>"+txt_ProbabilityOfHydrophilicAndHydrophobicProp
erties.getText());System.out.println("Image => "+ImgPath);
```

```
    }


    // Button Update Data From MySQL Database
    // 1 - Check If Inputs Is Not Null
    //     If The imgPath Is Not Null Update Also The Image
    //     else don't update theImage
    // 2 - Update The Data
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

        if(checkInputs() && txt_id.getText() != null)
        {
        String UpdateQuery = null;
        PreparedStatement ps = null;
        Connection con = getConnection();

       // update without image
       if(ImgPath == null)
       {
        try {
       UpdateQuery = "UPDATE products SET name = ?, (ALL PROTEIN PROPERTIES) = ?"
       + ", add_date = ? WHERE id = ?";
                ps = con.prepareStatement(UpdateQuery);

                ps.setString(1, txt_name.getText());
                ps.setString(2, txt_SolventAccessibility.getText());
ps.setString(3, txt_Antigenicity.getText());
ps.setString(4, txt_SolubilityUponOverExpression.getText());
ps.setString(5, txt_PredictedDisorderedProbability.getText());
ps.setString(6, txt_PredictedOrderedProbability.getText());
ps.setString(7, txt_AlphaHelicalTransmembraneProtein.getText());
ps.setString(8, txt_BetaBarrelTransmembraneProtein.getText());
ps.setString(9, txt_CompositionOfAminoAcids.getText());
ps.setString(10, txt_AverageMassOfAminoAcids.getText());
ps.setString(11, txt_IsoElectricPoint.getText());
ps.setString(12, txt_RateOfTransmittedLightAssumingCystine.getText());
ps.setString(13, txt_AbsorbanceForMediumAssumingCystine.getText());
ps.setString(14, txt_RateOfTransmittedLightWithOutAssumingCystine.getText());
ps.setString(15, txt_AbsorbanceForMediumWithOutAssumingCystine.getText());
ps.setString(16, txt_NonReliabilityOfProtein.getText());
 ps.setString(17, txt_RelativeVolumeOccupiedByAliphaticSideChains.getText());
 ps.setString(18, txt_ProbabilityOfHydrophilicAndHydrophobicProperties.getText());


                SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
                String addDate = dateFormat.format(txt_AddDate.getDate());

                ps.setString(19, addDate);

                ps.setInt(20, Integer.parseInt(txt_id.getText()));

                ps.executeUpdate();
                Show_Products_In_JTable();
                JOptionPane.showMessageDialog(null, "Product Updated");

            } catch (SQLException ex) {
                Logger.getLogger(Main_Window.class.getName()).log(Level.SEVERE, null, ex);
            }

            }
            // update With Image
            else{
```

```
            try{
            InputStream img = new FileInputStream(new File(ImgPath));

             UpdateQuery = "UPDATE products SET name = ?, (ALL PROTEIN PROPERTIES)= ?"
                    + ", add_date = ?, image = ? WHERE id = ?";

             ps = con.prepareStatement(UpdateQuery);

             ps.setString(1, txt_name.getText());

ps.setString(2, txt_SolventAccessibility.getText());
ps.setString(3, txt_Antigenicity.getText());
ps.setString(4, txt_SolubilityUponOverExpression.getText());
ps.setString(5, txt_PredictedDisorderedProbability.getText());
ps.setString(6, txt_PredictedOrderedProbability.getText());
ps.setString(7, txt_AlphaHelicalTransmembraneProtein.getText());
ps.setString(8, txt_BetaBarrelTransmembraneProtein.getText());
ps.setString(9, txt_CompositionOfAminoAcids.getText());
ps.setString(10, txt_AverageMassOfAminoAcids.getText());
ps.setString(11, txt_IsoElectricPoint.getText());
ps.setString(12, txt_RateOfTransmittedLightAssumingCystine.getText());
ps.setString(13, txt_AbsorbanceForMediumAssumingCystine.getText());
ps.setString(14, txt_RateOfTransmittedLightWithOutAssumingCystine.getText());
ps.setString(15, txt_AbsorbanceForMediumWithOutAssumingCystine.getText());
ps.setString(16, txt_NonReliabilityOfProtein.getText());
ps.setString(17, txt_RelativeVolumeOccupiedByAliphaticSideChains.getText());
 ps.setString(18, txt_ProbabilityOfHydrophilicAndHydrophobicProperties.getText());

                SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
                String addDate = dateFormat.format(txt_AddDate.getDate());

                ps.setString(19, addDate);

                ps.setBlob(20, img);

                ps.setInt(21, Integer.parseInt(txt_id.getText()));

                ps.executeUpdate();
                Show_Products_In_JTable();
                JOptionPane.showMessageDialog(null, "Product Updated");

                 }catch(Exception ex)
                 {
                JOptionPane.showMessageDialog(null, ex.getMessage());
                 }
                 }
                 }else{
                JOptionPane.showMessageDialog(null, "One Or More Fields Are Empty Or Wrong");
                 }

                 }

// Button Delete The Data From MySQL Database
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

 if(!txt_id.getText().equals(""))
 {
 try {
Connection con = getConnection();
PreparedStatement ps = con.prepareStatement("DELETE FROM products WHERE id = ?");
 int id = Integer.parseInt(txt_id.getText());
 ps.setInt(1, id);
            ps.executeUpdate();
```

```
            Show_Products_In_JTable();
            JOptionPane.showMessageDialog(null, "Product Deleted");
    } catch (SQLException ex) {
        Logger.getLogger(Main_Window.class.getName()).log(Level.SEVERE, null, ex);
        JOptionPane.showMessageDialog(null, "Product Not Deleted");
    }

    }else{
    JOptionPane.showMessageDialog(null, "Product Not Deleted : No Id To Delete");
    }

    }

    // JTable Mouse Clicked
    // Display The Selected Row Data Into JTextFields
    // And The Image Into JLabel
      private void JTable_ProductsMouseClicked(java.awt.event.MouseEvent evt) {

        int index = JTable_Products.getSelectedRow();
        ShowItem(index);

        }

    // Button First Show The First Record
    private void Btn_FirstActionPerformed(java.awt.event.ActionEvent evt) {
    pos = 0;
    ShowItem(pos);
    }

    // Button Last Show The Last Record
    private void Btn_LastActionPerformed(java.awt.event.ActionEvent evt) {
    pos = getProductList().size()-1;
    ShowItem(pos);
    }


// Button Next Show The Next Record
private void Btn_NextActionPerformed(java.awt.event.ActionEvent evt) {

pos++;

if(pos >= getProductList().size())
{
pos = getProductList().size()-1;
}

ShowItem(pos);

}

// Button Previous Show The Previous Record
  private void Btn_PreviousActionPerformed(java.awt.event.ActionEvent evt) {

    pos--;

    if(pos < 0)
    {
    pos = 0;
    }

    ShowItem(pos);

    }
```

II. NOTES ON NET BEANS IDE:

The NetBeans Platform is a generic application framework for Java desktop applications. The NetBeans Platform provides the infrastuctural plumbing that, without it, every developer has to write themselves, such as solutions for persisting application state; connecting actions to menu items, toolbar items and keyboard shortcuts; window management, and much more. The NetBeans Platform provides all these out of the box so that you don't need to manually code these or other basic features yourself. Instead, you can focus on what your customers care about: domain-specific business logic. For example, developers of software for oil flow analysis can focus on their algorithms, while everything around it, from the architecture of the application to the display of windows to the user, is managed by the NetBeans Platform. The NetBeans Platform provides a reliable and flexible application architecture that can save you years of development time. There are many strategies and patterns available to help create applications that are robust and extensible, as its developers have many years of experience in creating flexible solutions. Many applications have been created on the NetBeans Platform.

III. CONCEPTS:

The modular nature of a NetBeans Platform application gives you the power to meet complex requirementswith relative ease. You're able to assemble small, simple, and easily tested modules that encapsulate coarselygrained application features.Versioning, available per module, helps give you confidence that your modules will work together, while strict control over the public APIs they expose help you create a flexible application that's easier to maintain.Since your application can use either standard NetBeans Platform modules or OSGi bundles,you're able tointegrate third-party modules or develop your own.Many applications, especially as they increase in size, tend to have multiple features performing similar tasks.The NetBeans Platform solves this problems through very heavy use of abstractions. For example, when youinteract with a file, you will be using a FileObject, not an instance of java.io.File. When you deal with menuitems and toolbar buttons, under the hood you will be using NetBeans Platform Action classes, instead ofdirectly interacting with menus and toolbars, though you can create custom components to use if you needthem. Another abstraction, on top of files and other business objects, is called Nodes.In general, the NetBeans Platform provides high-level abstractions to handle the common cases found insoftware development, while allowing the flexibility needed to do something more low-level if the need arises.Just as application servers, such as GlassFish or WildFly, provide lifecycle services to web applications,the NetBeans Platform aims to provide lifecycle services to Java desktop applications. Application serversunderstand how to compose web modules, EJB modules, and related artifacts, into a single web application.In a comparable manner, the NetBeans Platform understands how to compose NetBeans modules into a singleJava desktop application.Furthermore, there is no need to write a main method for your application because the NetBeans Platformalready contains one. Support is provided for persisting user settings across restarts of the application, such as, by default, the size and positions of the windows in the application.End users of the application benefit from pluggable applications because these enable them to install newfeatures, in the middle of release cycles, into their running applications. At runtime, NetBeans modules can be installed, uninstalled, activated, and deactivated. The NetBeans Platform provides an infrastructure for registering and retrieving service implementations,enabling you to minimize direct dependencies between individual modules and enabling a loosely coupled architecture, via built-in strategies supporting "high cohesion and low coupling".The NetBeans Platform provides a virtual file system, which is a hierarchical registry for storing user settings,comparable to the Windows Registry on Microsoft Windows systems. It also includes a unified API providing stream-oriented access to flat and hierarchical structures, such as disk-based files on local or remote servers,memory- based files, and even XML documents.Most serious applications need more than one window. Coding good interaction between multiple windowsis not a trivial task. The NetBeans window system lets you maximize/minimize, dock/undock, and drag-anddrop windows, without you providing any code at all.JavaFX and Swing are the standard UI toolkits on the Java desktop and can be used throughout the NetBeansPlatform. One of the most striking aspects of the design and codebase of the NetBeans Platform is its use of standards.Wherever a standard for doing something exists, the developers of the NetBeans Platform opt to use it, ratherthan reinvent the wheel. For example, module manifest files are based on the Java Versioning Specification,Nodes are conceptually based on the JavaBeans BeanContext specification, and so on. Wherever there wasan existing standard or a near match, it was used.What this adherence to standards achieves is extensibility. As other pieces of code that work with the same standards are created, it is much less difficult to get them to interoperate with the NetBeans Platform.It requires greater discipline to adhere to standards than to reinvent the wheel, but doing so gets you maintainability and interoperability, as standards are, by definition, documented, and if something is astandard, others are probably using it as well.

IV. MODULE SYSTEM:

Refactoring a monolithic application into its distinct parts makes so much sense that it almost feels superfluous to explain what a module is and what its benefits are. Nevertheless, a brief explanation of what and why is needed, for those who are new to the concept, and especially for those who are familiar with it, since their understanding of modularity may be different to the implementation of the concept in the NetBeans Platform. A NetBeans module is a JAR file, in other words, a Java archive containing a set of classes and other files that make up the module. What makes a JAR a module is comparable to what makes a JAR an OSGi bundle, that is, a set of special keys in the JAR manifest file that mark it as a module and tell the NetBeans Platform what to do to install it. The NetBeans module system provides an alternative to OSGi, one that has all the most important elements of OSGi, without its complexity. Conceptually, a NetBeans module is a wrapper around a JAR file, providing it with its own classloader and version number. By default, any classes within a module cannot be seen by any classes in any other module. The import statement cannot be used to import such classes. In fact, compilation fails for any class that imports classes from packages that have not been exposed by the modules in which they are found. The encapsulation provided by a NetBeans module protects its content from being misused, since the module needs to explicitly expose a package for other modules to be able to use the classes found within the package. Each distinct feature in a NetBeans Platform application is organized into a distinct module. What a feature is, and what a module's boundaries are, is something about which the NetBeans Platform is not opinionated. Development teams are free to use NetBeans modules in whatever way makes sense to their business needs. How many modules an application should have, and how large a module should be, are also decisions about which the NetBeans Platform has no opinion. Multiple small modules may provide a flexible structure, but may be difficult to maintain, while organizing the complete application into a single module, while possible, runs counter to the aims of modularity.

One point of potential terminology confusion is the use of the terms plugin and module. For most practical intents and purposes, there is no difference. A module is simply a unit of code that you can plug in to the NetBeans Platform. The term plugin has been popularized by various other environments and can easily be applied to modules created for the NetBeans Platform as well. Traditionally, the term module is used in the NetBeans Platform environment, since the NetBeans Platform itself is composed of modules. For example, you cannot say that the core NetBeans Platform is composed of plugins. On the other hand, if you are creating new features for the NetBeans Platform but your feature set is composed of multiple modules, you might prefer to refer to those modules collectively as a single plugin.

V. DEPLOYMENT -FORMAT:

To support the ideals of modularity, it should be possible to make a module available to users via a single deployment package. Comparable to a JAR achive, and building on top of the same specification and concept, it should be possible to bundle an arbitrary set of source files and other resources into an archive for distribution to users of the application. The NetBeans Platform equivalent of the JAR archive format is the NBM archive format. The letters NBM stand for "NetBeans Module". Together with module-specific metadata, such as instructions about the conditions under which the module should be activated within the application, the classes making up the module are packaged into an NBM file. In Ant-based NetBeans modules, a target is included for generating the NBM file, while a goal is available in Maven-based NetBeans modules for the same purpose. NetBeans IDE includes a "Create NBM" menu item for NetBeans module projects, so that NBM files can be created in the IDE. NBM files can be inspected in the same way as any other archive, such as via a ZIP utility

It should be possible to uniquely identify a module. For that purpose, a unique String as an identifier for the module is a mandatory part of manifest files in NetBeans modules, using the OpenIDE-Module key. Via tools in NetBeans IDE you can define the unique identifier and modify it as needed. Be aware that the OpenIDE-Module key, together with versioning details, is also required for providing automatic updates in the Plugin Manager. Versioning is a feature available to modules. Those responsible for the final application can then select modules and versions of modules that are most suitable for a specific business need. It is not necessarily advisable to select the latest version of a module. The latest version might be too buggy or not cooperate well with the rest of the application. It may be better to select an older version of a module or even to not include the given module in the final application at all. NetBeans modules differentiate between specification versions and implementation versions.The version indicating the generation of the module, that is, only increment this version when, for example, the architecture of the module is rewritten. The major release version is appended to the end of the OpenIDE-Module key in the manifest, such as org.carsales.carviewer/1, where /1 indicates that this is the first major release of the module.The version of the officially exposed interfaces, defined by the OpenIDE-Module-Specification-Version key in the manifest. The assumption is that the interfaces are a public API and that the new version is backward compatible with previous versions. Following convention, the first bug fix release of a module appends .1 after the previously defined specification version, such as from 4.0 to 4.0.1.The implementation state of a module, define by the OpenIDE-Module-Implementation-Version key in the manifest. The implementation version is not assumed to be backward compatible and can only be used within a specific version of the application. This is useful if a module explicitly depends on one specific implementation of another module, which should be avoided as far as possible.

Example to manifest file that could be included with the JAR tools –m option:

```
Manifest-Version: 1.0
OpenIDE-Module: com.modulemakers.clip_disp/2
OpenIDE-Module-Specification-Version: 2.0.1
OpenIDE-Module-Implementation-Version: 2.0-beta-rewrite
OpenIDE-Module-Build-Version: 2003-12-31 00:23 cron@buildhost.mycorp.com
OpenIDE-Module-Name: Clipboard Displayer
OpenIDE-Module-Name_cs: Prohlizec schranky
OpenIDE-Module-Install: com/modulemakers/clip_disp/Installer.class
OpenIDE-Module-Layer: com/modulemakers/clip_disp/Layer.xml
X-Comment-1: I am a comment (just a deliberately meaningless
X-Comment-2: header) - "Sealed" is a standard manifest attribute.
Sealed: true

Name: com/modulemakers/clip_disp/DisplayClipboardAction.class
OpenIDE-Module-Class: Action
```

Using custom module class:

```
 package com.modulemakers.clip_disp;
 import.org.openide.modules.ModuleInstall
 ; import org.openide.filesystems.FileUtil;
 import java.net.*;

public class ModuleHandler extends ModuleInstall {
 public void installed() {
 // This module has been installed for the first time! Notify authors.
HttpURLConnection conn = (HttpURLConnection)
  (new URL ("http://www.modulemakers.com/clip_disp/installed.cgi").openConnection ());
 conn.getResponseCode ();
 conn.disconnect ();
 // Handle setup within this session too:
 restored ();
  }

  // Nothing special required here.
  // public void restored() {
  // }

  // Do not need to do anything special on uninstall.
  // Tools action will be removed automatically.
  // public void uninstalled() {
  // }

  public boolean closing() {
 // Ask the user to save any open, modified clipboard contents.
 // If the user selects "Cancel" on one of these dialogs, don't exit yet!
  return DisplayClipboardAction.askAboutExiting ();
  }
  }
```

Specification and requesting all versions:

```
OpenIDE-Module: com.mycom.mymodule/1
OpenIDE-Module-Specification-Version:  1.0.1
OpenIDE-Module-Implementation-Version: 1.0-release-g
OpenIDE-Module-Module-Dependencies:
com.mycom.mysistermodule/1 = 1.0-release-g,
```

com.othercom.anothermodule > 2.1,
org.netbeans.modules.applet/1 > 1.0
OpenIDE-Module-Package-Dependencies: javax.television > 0.9
OpenIDE-Module-Java-Dependencies: Java = 1.2.1b4, VM > 1.0
OpenIDE-Module-Provides: javax.television.TunerProvider, javax.television.RemoteControl
OpenIDE-Module-Requires: org.netbeans.javahelp.api.Help

VI. NUMBERING SCHEME FOR UPDATES:

While developers have the responsibility to manage dependencies from their modules to both the Open APIs and other modules, and mark API changes of all sorts with changes in the module or API specification version, release engineers who publish modules also need to make version-number changes. Remember, it is never particularly harmful to increase the specification version (for example before cutting a release of a module), and either developers or release engineers may do so - such changes of course do not need any matching documentation as described above for API changes.It is recommended that API and module specification versions in the trunk follow a two-digit scheme such as 1.5, where the next in sequence would be 1.6. On a release branch, three-digit schemes should be used, such as 1.5.1, 1.5.2, and so on. Post- release patches could have four digits, and so on.If a number of API changes are made between releases, it may be annoying for the API specification version to be e.g. 1.133. Additionally, if specification versions of the APIs are to be used to distinguish the APIs available in each IDE release, they should be more mnemonic. So it may be useful to choose a new first digit after a release. For example, 1.20 may be branched for a release, forming 1.20.1 and so on, released as 1.20.4; meanwhile, the development builds become 2.1 rather than 1.21, so that everyone can remember that 1.x numbers mean one release, and 2.x numbers the next release.It is important that every published release of a module have a different specification version. Otherwise automated updates cannot work correctly. Of course, if a "new version" of a module is being published solely because it was included in some bugfix build, and in fact did not contain any user-noticeable changes from the last released version, release engineers may prefer to either avoid increasing its specification version, or withhold it from the update center altogether, so as to prevent users of the previous similar version from unwittingly wasting time downloading it; but this is difficult to manage and no one currently does .Please remember that implementation versions of modules are not intended to be ordered. Implementation versions need not actually be numeric at all, and the IDE's Modules API intentionally prevents inter-module dependencies from using them except as exact string comparisons. Specification versions, by contrast, must be numeric, and the only permitted comparisons in dependencies are of the form "version x.y.z or anything greater".As a practical policy for using implementation versions, it is helpful to make them integers if they are being used in implementation dependencies from other modules, and use the build property spec.version.base in both producers and consumers of implementation dependencies in place of a fixed specification version. This trick makes management of complex sets of modules with implementation dependencies much easier. From the NBM project GUI, just check the checkbox Append Implementation Versions Automatically in the Versioning panel.Release engineers should assume that module manifests provide complete information about which versions of what module may be run on which version of the IDE, via their major release versions, specification versions, and dependencies. Of course these assumptions should also be tested before actually publishing something on a public update server; but if any inconsistencies are found, these are P1/P2 bugs for the developer and it is better to resolve them properly in the code, than to use tricks in the update server to force certain configurations of modules to be loaded

VII. CONCLUSION:

The error Reached End of File While Parsing is a compiler error and almost always means that your curly parenthesis are not ending completely or maybe there could be extra parenthesis in the end. Every opening braces { needs one closing braces }. The only purpose of the extra braces is to provide scope-limit . If you put curly braces in the wrong places or omit curly braces where the braces should be, your program probably won't work at all. Moreover, If you don't indent lines of code in an informative manner , your program will still work correctly, but neither you nor any other programmer will be able to figure out what you were thinking when you wrote the code. Because this error is both common and easily avoided , using a code editor like **NETBEANS** or **ECLIPSE** . Using these IDE's, you can autoformat your code by pressing Alt+Shift+F . This will indent your code properly and align matching braces with the control structure (loop, if, method, class) that they belong to. This will make it easier for you to see where you're missing a matchingbrace.

VIII. REFERENCES:

[1] Custodio, E., & Castro, M. (2016). Advancing pre-enrollment procedure through online registration and grade evaluation system. International Journal of Signal Processing Systems, 4(5), 399-404, doi:10.18178/ijsps.4.5.399-404

[2] Singh, R., Singh, R., Kaur, H., & Gupta, O. (2016). Development of Online Student Course Registration System. Oriental Journal of Computer Science & Technology, 9(2), doi: 10.13005/ojcst/9.02.0

[3] Darunday, J., Melarpis, F., Monserrat, T., & Recario, R. (2016). Flash crowd online student enrollment system. doi: 10.13140/RG.2.2.21700.53125

[4] Kumar, B. (2011). Thin client web-based campus information systems for Fiji National University. International Journal of Software Engineering & Applications, 2(1), doi: 10.5121/ijsea.2011.2102

[5] Thompson, M., & Ahn, B. (2012). The development of an online grading system for distributed grading in a large first year project-based design course. In Proceedings of the 119th ASEE Annual Conference & Exposition.

[6] Frenzel, C. (1999). Management of Information Technology. Third Edition. Course Technology Press, Cambridge, MA.

[7] Marcial, D. (2012). Information technology resources in the higher education institutions in the Philippines. Philippine Information Technology Journal,
[8] Then, P. (2006). Online Student Enrollment System. Swinburne University of Technology (Sarawak Campus).
[9] Custinar, A., Cruz, C., Ecobiza, T., Piquero, M. (2015) Computerized Enrollment System for Mary Lourdes Academy. STI College – Global City.
[10] Bacala, M., Reano, E. (2009). Online Enrollment System For Cavite Maritime Institute Dasmarinas, Cavite. De La Salle University – Dasmariñas, College of Science
[11] The U.S National Archives and Record Administration, 8601, Adelphi, Record, College Park MD 20740-6001, 2001
[12] Arch intern Med-Abstract, Electronic Health Record Use and the Quality of Ambulatory Care in the U.S, Vol. 167
[13] Jeffery A. Linder, MD, MPH, Jun. Ma, MD, Rd, PhD, David W. Bates MD, MSC, Blackford Middleton MD, PhD, MSC, Randall Stafford, MD, PhD, Arch Intern Med. 2007, 67 (13), 1400-1405
[14] Jain A. K, Ross A, Prabhakar S, (2004), "An introduction to biometric recognition, " IEEE Transactions on Circuits Systems for Video Technology, vol. 14(1), pp. 4-20, 2004.
[15] He D, Wang D, (2014), "Robust biometrics-based authentication scheme for multi-server environment, " IEEE Systems Journal, pp. 1-8, 2014.
[16] Jing Dong, Tieniu Tan. (2008), "E ects of watermarking on iris recognition performance". Proceedings of the 10th International Conference on Control, Automation, Robotics and Vision, pp. 1156- 1161, 2008.
[17] J. Hammerle-Uhl, K. Raab, A. Uhl. (2010), "Experimental study on the impact of robust watermarking on iris recognition accuracy", Proceedings of the 25th ACM Symposium on Applied Computing, pp 1479-1484, 2010.
[18] A. Lang, J. Dittmann. (2007), "Digital watermarking of biometric speech references: impact to the peer system performance". In E. J. Delp and P. W. Wong, editors, Security, Steganography, and Watermarking of Multimedia Contents IX, number 6505 in Proceedings of SPIE, page 650513 , 2007.
[19] M. I. Rajibul, M. S Shohel, S. Andrews. (2008), "Biometric template protection using watermarking with hidden password encryption". Proceedings of the International Symposium on Information Technology 2008, pp 296-303, 2008.
[20] W. Stallings, (2010), "Cryptography and Network Security: Principles and Practices". Prentice-Hall, 5th edition, Upper Saddle River, NJ, USA, 2010.
[21] I.-E. Liao, C.-C. Lee, M.-S. Hwang, (2006), "A password authentication scheme over insecure networks, " Journal of Computer and System Sciences, vol. 72, pp. 727-740, 2006.
[22] M. Jakobsson, M. Dhiman, (2013), "The benefits of understanding passwords, " in Mobile Authentication, ser. Springer Briefs in Computer Science. Springer New York, 2013, pp. 5-24.
[23] M. Weir, S. Aggarwal, M. Collins, D. H. Stern, (2010), "Testing metrics for password creation policies by attacking large sets of revealed passwords, " in Proceedings of the 17th ACM