Table of Contents – Code

Page

1 - AddCategory.java

```
package JFrames;

import Java.Categorie;
import com.mysql.cj.jdbc.Driver;
import java.sql.Statement;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import java.util.Vector;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableModel;
import jdk.jfr.Category;

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */

/**
 *
 * @author joni
 */
public class AddCategory extends javax.swing.JFrame {

    /**
     * Creates new form Category
     */
    public AddCategory() { //Constructor
        initComponents();
        Connect();
        LoadTable();
        this.setLocationRelativeTo(null);
        addBtn.setEnabled(false);
        JTableCategory.setRowHeight(26);
    }

    Connection con = null;
    PreparedStatement pst = null;
    ResultSet rs = null;

    public void validateField(){
```

```
        String category = txtCategory.getText();
        if(!category.equals(""))
            addBtn.setEnabled(true);
        else
            addBtn.setEnabled(false);
    }

public class CategoryDao{  // This class deletes the category when the user clicks the table
    public static void delete(String ID) { // Deletes by ID
        String query = "delete from Category where CategoryID = '"+ID+"'";
        try {
            DbOperation.setDataOrDelete(query, "Category deleted successfully!");
        } catch (SQLException ex) {
            Logger.getLogger(Category.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

        public static ArrayList<Category> getAllCategories(){
        ArrayList<Category> categoryList = new ArrayList<Category>();
        Connection con = ConnectionProvider.getConnection();
        String query = "SELECT * FROM `Category`";

        Statement st;
        ResultSet rs;

        try {
            st = con.createStatement();
            rs = st.executeQuery(query);
            Category category = null;
            while (rs.next())
                category = (Category) new Categorie(
                        rs.getInt("ID"),
                        rs.getString("Name"));

              categoryList.add(category);
            }

        catch(Exception e){
            JOptionPane.showMessageDialog(null, e);
        }
        return categoryList;

    }

}

    public void Connect(){
     //This connect method was used in all JFrame's that had jTables in them
        try {
```

```
            Class.forName("com.mysql.cj.jdbc.Driver");
            con = DriverManager.getConnection("jdbc:mysql://localhost:3307/Restaurant", "root", "");

        } catch (ClassNotFoundException ex) { //This catch searches if the class if found, we can remove the
'cj' and it should still work the same way.
            Logger.getLogger(Driver.class.getName()).log(Level.SEVERE, null, ex);
        } catch (SQLException ex) { // This catch get the connection to my desired MySQL database,
"Restaurant".
            Logger.getLogger(Driver.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    public void LoadTable(){

        int c;

        try{
            pst = con.prepareStatement("select * from Category");
            rs = pst.executeQuery();

            com.mysql.cj.jdbc.result.ResultSetMetaData rsd = (com.mysql.cj.jdbc.result.ResultSetMetaData)
rs.getMetaData();
            c = rsd.getColumnCount();

            DefaultTableModel d = (DefaultTableModel)JTableCategory.getModel();
            d.setRowCount(0);

            while(rs.next()){
                Vector v2 = new Vector();

                for(int i = 1; i<=c; i++){
                    v2.add(rs.getString("CategoryID"));
                    v2.add(rs.getString("Category"));
                }
                d.addRow(v2);

            }

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

    }


 public class ConnectionProvider{  // This class provides the connection to the MySQL Server
     public static Connection getConnection() {
        Connection con = null;
        try {
```

```
            con = DriverManager.getConnection("jdbc:mysql://localhost:3307/Restaurant", "root", "");
            return con;
        } catch (SQLException ex) {
            Logger.getLogger(Employees.class.getName()).log(Level.SEVERE, null, ex);
            return null;
        }
    }

}

    public class DbOperation{
        public static void setDataOrDelete(String Query, String msg) throws SQLException{
            try{
                Class.forName("com.mysql.cj.jdbc.Driver");
                Connection con = ConnectionProvider.getConnection();
                Statement st = con.createStatement();
                st.executeUpdate(Query);
                if(!msg.equals(""))
                    JOptionPane.showMessageDialog(null, msg);

            } catch (Exception e) {
              JOptionPane.showMessageDialog(null, e, "Message", JOptionPane.ERROR_MESSAGE);
            }
        }
        public static ResultSet getData(String query){
        try{
            Connection con = ConnectionProvider.getConnection();
            Statement st = con.createStatement();
            ResultSet rs = st.executeQuery(query);
            return rs;
        }
        catch(Exception e){
            JOptionPane.showMessageDialog(null, e, "Message", JOptionPane.ERROR_MESSAGE);
            return null;
        }

    }
    }


    public static void delete(String ID) {
        String query = "delete from Category where CategoryID = '"+ID+"'";
        try {
            DbOperation.setDataOrDelete(query, "Category deleted successfully!");
        } catch (SQLException ex) {
            Logger.getLogger(Category.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
```

```
    private void txtCategoryKeyReleased(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
        validateField();
    }

    private void addBtnActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:

        String Name = txtCategory.getText();

        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            con = DriverManager.getConnection("jdbc:mysql://localhost:3307/Restaurant", "root", "");
            pst = con.prepareStatement("insert into Category (Category) values(?)");
            pst.setString(1, Name);
            pst.executeUpdate();
            JOptionPane.showMessageDialog(null, "Category Added!");
            LoadTable();

            txtCategory.setText("");


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

    }

    private void JTableCategoryMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        int index = JTableCategory.getSelectedRow();
        TableModel model = JTableCategory.getModel();
        String ID = model.getValueAt(index, 0).toString();
        String Name = model.getValueAt(index, 1).toString();
        int A = JOptionPane.showConfirmDialog(null, "Do you want to remove the " +Name+ " category?",
"Select", JOptionPane.YES_NO_OPTION);
        if(A==0){
            CategoryDao.delete(ID);
            setVisible(false);
            new AddCategory().setVisible(true);
        }
    }

    private void lblBackMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        AdminPage Admin = new AdminPage();
        Admin.setVisible(true);
```

```
        dispose();
    }

    // Variables declaration - do not modify
    private javax.swing.JTable JTableCategory;
    private javax.swing.JButton addBtn;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPanel jPanel3;
    private javax.swing.JPanel jPanel4;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JLabel lblBack;
    private javax.swing.JTextField txtCategory;
    // End of variables declaration

}
```

2 - *AddProduct.java*

```
package JFrames;

import Java.MyQuery;
import com.mysql.cj.jdbc.Driver;
import java.awt.Color;
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.util.HashMap;
import java.util.Vector;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */
/**
 *
```

```
 * @author joni
 */
public class AddProduct extends javax.swing.JFrame {

    Color EnterBack = new Color(115,51,13);
    Color ExitBack = new Color(129,69,34);

    /**
     * Creates new form AddProduct
     */
    public AddProduct() {
        initComponents();
        Combo();
        this.setLocationRelativeTo(null);
        addBtn.setEnabled(false);
        clearBtn.setEnabled(false);
        categoryLoad();
//        LoadTable();
    }

    Connection con = null;
    PreparedStatement pst = null;
    ResultSet rs = null;

    // Validation rules in order to be able to add and change information about a product
    // ! means not
    public void validateField() {
        String Name = txtName.getText();
        String Price = txtPrice.getText();
        String Qty = txtQty.getText();
        String Description = txtDescription.getText();
        if (!Name.equals("") && !Price.equals("") && !Qty.equals("")) { // And
            addBtn.setEnabled(true);
        } else {
            addBtn.setEnabled(false);
        }
    }
        public void validateFields() {
        String Name = txtName.getText();
        String Price = txtPrice.getText();
        String Qty = txtQty.getText();
        String Description = txtDescription.getText();
        if (!Name.equals("") || !Price.equals("") || !Qty.equals("")) { // Or
            clearBtn.setEnabled(true);
        } else {
            clearBtn.setEnabled(false);
        }
    }
            public Connection getConnection() {
```

```
        Connection con = null;
        try {
            con = DriverManager.getConnection("jdbc:mysql://localhost:3307/Restaurant", "root", "");
            return con;
        } catch (SQLException ex) {
            Logger.getLogger(Employees.class.getName()).log(Level.SEVERE, null, ex);
            return null;

        }
    }

        //Source: https://youtu.be/WAIcC3pl94o (modified)
        public void Combo(){
        MyQuery m = new MyQuery(); //import the 'MyQuery' class.
        Connection con = m.getConnection(); //Connect to the MySQL database.
        HashMap<String, Integer> map = m.comboBox(); //import the java.util.HashMap.
        for(String st : map.keySet()){ //the keyset method returns what was put in the map
                            //and the for loop iterates through those string items(the category names).
            cbCategory.addItem(st); //puts the location of the string items in the Category jcombobox.
        }
}

    private void txtNameKeyReleased(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
        validateField();
        validateFields();
    }

    private void txtPriceKeyReleased(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
        validateField();
        validateFields();
    }

    private void txtDescriptionKeyReleased(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
        validateField();
        validateFields();
    }

    private void formComponentShown(java.awt.event.ComponentEvent evt) {
        // TODO add your handling code here:
    }

    private void addBtnActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:

        String Name = txtName.getText();
        String Category = cbCategory.getSelectedItem().toString();
```

```
        String Price = txtPrice.getText();
        String Qty = txtQty.getText();
        String Description = txtDescription.getText();

        try {

            pst = con.prepareStatement("insert into Product (Name, Category, Price, Qty, Description)
values(?,?,?,?,?)");
            pst.setString(1, Name);
            pst.setString(2, Category);
            pst.setString(3, Price);
            pst.setString(4, Qty);
            pst.setString(5, Description);

            int k = pst.executeUpdate();

            if (k == 1) {
                JOptionPane.showMessageDialog(this, "Product Added Successfully!");

                txtName.setText("");
                cbCategory.setSelectedIndex(-1);
                txtPrice.setText("");
                txtQty.setText("");
                txtDescription.setText("");

                txtName.requestFocus();
            } else {
                JOptionPane.showMessageDialog(this, "Error! Product Not Added!");
            }

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

    }

    private void clearBtnActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        txtName.setText("");
        cbCategory.setSelectedIndex(-1);
        txtPrice.setText("");
        txtQty.setText("");
        txtDescription.setText("");
    }

    private void txtQtyKeyReleased(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
        validateField();
        validateFields();
```

```
}

private void lblBackMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    AdminPage Admin = new AdminPage();
    Admin.setVisible(true);
    dispose();
}

private void lblBackMouseEntered(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    panelBack.setBackground(EnterBack);

}

private void lblBackMouseExited(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    panelBack.setBackground(ExitBack);

}

private void txtPriceKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    // Validation so only integers can be typed in this field
        if (!Character.isDigit(evt.getKeyChar())) {
        evt.consume();
    }
}

private void txtQtyKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    // Validation which restricts typable elements to type Integer only
        if (!Character.isDigit(evt.getKeyChar())) {
        evt.consume();
    }
}

private void cbCategoryActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}
public class CategoryItem {  // Class to call jComboBox from category

    int ID;
    String Name;

    public CategoryItem(int ID, String Name) {
        this.ID = ID;
        this.Name = Name;
    }
```

```
        public String toString() {
            return Name;
        }
    }

    private void categoryLoad() { // Necessary because  we get categpry information from the Category SQL Database
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            con = DriverManager.getConnection("jdbc:mysql://localhost:3307/Restaurant", "root", "");
            pst = con.prepareStatement("select * from Category");
            rs = pst.executeQuery();
            cbCategory.removeAllItems();

            while (rs.next()) {
                cbCategory.addItem(new CategoryItem(rs.getInt(1), rs.getString(2)));
            }

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

    }

    // Variables declaration - do not modify
    private javax.swing.JButton addBtn;
    private javax.swing.JComboBox cbCategory;
    private javax.swing.JButton clearBtn;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPanel jPanel3;
    private javax.swing.JPanel jPanel4;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTextField jTextField1;
    private javax.swing.JLabel lblBack;
    private javax.swing.JPanel panelBack;
    private javax.swing.JTextArea txtDescription;
    private javax.swing.JTextField txtName;
    private javax.swing.JTextField txtPrice;
```

```
    private javax.swing.JTextField txtQty;
    // End of variables declaration
}
```

3 - *AdminPage.java*

```
package JFrames;

import java.util.Date;
import com.mysql.cj.jdbc.Driver;
import com.mysql.cj.jdbc.result.ResultSetMetaData;
import java.awt.BorderLayout;
import java.awt.Color;
import java.sql.Statement;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Connection;
import java.sql.SQLException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Vector;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartPanel;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.plot.PiePlot;
import org.jfree.data.general.DefaultPieDataset;

/**
 *
 * @author joni
 */
public class AdminPage extends javax.swing.JFrame implements Runnable{
                    //inheritence               //implements Runnable
                                            //to create the thread below
    int hour, minute, second;
    int day, month, year;
    String timestr, yearstr;

// Colors are used throughout the program to give the appearance that something is being navigated upon
before being clicked
Color EnterColor = new Color(0,0,0);
Color ExitColor = new Color(78,78,78);
Color LogoutEnterColor = new Color(86,85,167);
```

```
Color LogoutExitColor = new Color(129,129,243);
    /**
     * Creates new form AdminPage
     */
    public AdminPage() {  // Constructor

        initComponents();
        showPieChart();
        setDataToCards();
        CustomerLoad();
        SalesLoad();
        MessagesLoad();
        Connect();

        Thread t = new Thread(this); //multithreading to execute the start method below
        t.start(); //You may need to comment the clock because its making the laptop overheat because its
an overridden function!

        JTableSales.setRowHeight(20);
        JTableCustomer.setRowHeight(20);
        JTableMessage.setRowHeight(50);
        JTableMessage.getColumnModel().getColumn(0).setPreferredWidth(3);
        JTableMessage.getColumnModel().getColumn(1).setPreferredWidth(50);
    }

    Connection con;
    PreparedStatement pst;
    ResultSet rs;

    public void Connect(){

        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            con = DriverManager.getConnection("jdbc:mysql://localhost:3307/Restaurant", "root", "");
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(Driver.class.getName()).log(Level.SEVERE, null, ex);
        } catch (SQLException ex) {
            Logger.getLogger(Driver.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

      public void CustomerLoad(){

         int x;

         try{
             pst = con.prepareStatement("select * from CustomerOrder");
             rs = pst.executeQuery();
```

```
            ResultSetMetaData rsd = (ResultSetMetaData) rs.getMetaData(); //import
com.mysql.cj.jdbc.result.ResultSetMetaData
            x = rsd.getColumnCount();

            DefaultTableModel dtm = (DefaultTableModel)JTableCustomer.getModel(); //import
javax.swing.table.DefaultTableModel
            dtm.setRowCount(0);

            while(rs.next()){
                Vector vec = new Vector(); //import the java.util.Vector

                for(int i = 1; i<=1; i++){ //Now write all the respective fields in the database 'CustomerOrder' in
order and this iterates through 'i'.
                    vec.add(rs.getString("CustomerID"));
                    vec.add(rs.getString("Name"));
                    vec.add(rs.getString("Category"));
                    vec.add(rs.getString("Price"));
                    vec.add(rs.getString("Qty"));
                    vec.add(rs.getString("Total"));
                    }
                dtm.addRow(vec); //Delivers the rows in the 'vec' object above from the MySQL database
'CustomerOrder' to be visible in the jTable here.
            }
        } catch(SQLException ex){
            Logger.getLogger(Driver.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

     public void SalesLoad(){

        int x;

        try{
            pst = con.prepareStatement("select * from Sales");
            rs = pst.executeQuery();

            ResultSetMetaData rsd = (ResultSetMetaData) rs.getMetaData();
            x = rsd.getColumnCount();

            DefaultTableModel dtm = (DefaultTableModel)JTableSales.getModel();
            dtm.setRowCount(0);

            while(rs.next()){
                Vector vec = new Vector();

                for(int i = 1; i<=1; i++){
                    vec.add(rs.getString("CustomerID"));
                    vec.add(rs.getString("Date"));
                    vec.add(rs.getString("Total"));
```

```
                vec.add(rs.getString("Paid"));
                vec.add(rs.getString("Balance"));
                }
            dtm.addRow(vec);
        }
    } catch(SQLException ex){
        Logger.getLogger(Driver.class.getName()).log(Level.SEVERE, null, ex);
    }
}

  public void MessagesLoad(){

    int x;

    try{
        pst = con.prepareStatement("select * from CustomerReview");
        rs = pst.executeQuery();

        ResultSetMetaData rsd = (ResultSetMetaData) rs.getMetaData();
        x = rsd.getColumnCount();

        DefaultTableModel dtm = (DefaultTableModel)JTableMessage.getModel();
        dtm.setRowCount(0);

        while(rs.next()){
            Vector vec = new Vector();

            for(int i = 1; i<=1; i++){
                vec.add(rs.getString("Date"));
                vec.add(rs.getString("Message"));
                }
            dtm.addRow(vec);
        }
    } catch(SQLException ex){
        Logger.getLogger(Driver.class.getName()).log(Level.SEVERE, null, ex);
    }
}

 // Method to get the number of the last row from the tables to display on the labels
// Credits: UniqueDeveloper on YouTube: part 11 of library management system.
 public void setDataToCards(){

    Statement st = null;
    ResultSet rs = null;

    long l = System.currentTimeMillis();
    Date todaysDate = new Date(1);

    try{
```

```
        Class.forName("com.mysql.cj.jdbc.Driver");
         con = DriverManager.getConnection("jdbc:mysql://localhost:3307/Restaurant", "root", "");
          st = con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
          rs = st.executeQuery("select * from Employee");
          rs.last();
          lbl_Employees.setText(Integer.toString(rs.getRow()));

          rs = st.executeQuery("select * from Category");
          rs.last();
          lbl_Categories.setText(Integer.toString(rs.getRow()));

          rs = st.executeQuery("select * from Product");
          rs.last();
          lbl_Products.setText(Integer.toString(rs.getRow()));

       }catch (Exception e){
           e.printStackTrace();
       }
   }

   public void showPieChart(){

     // Create Dataset
     DefaultPieDataset barDataset = new DefaultPieDataset();

     try{
        con = DriverManager.getConnection("jdbc:mysql://localhost:3307/Restaurant", "root", "");
        String sql = "select Total, count(*) as count from Sales group by CustomerID";
        Statement st = con.createStatement();
        ResultSet rs = st.executeQuery(sql);
        while(rs.next()){
            barDataset.setValue(rs.getString("Total"), new Double(rs.getDouble("count")));
        }

     }catch (Exception e){
        e.printStackTrace();
     }

      // Create Pie Chart
      JFreeChart piechart = ChartFactory.createPieChart3D("Customers' Spending",barDataset,
false,true,false);//explain

       PiePlot piePlot =(PiePlot) piechart.getPlot();

       piePlot.setBackgroundPaint(Color.white);

       // Create chartPanel to display Pie Chart
       ChartPanel barChartPanel = new ChartPanel(piechart);
```

```
        panelPieChart.removeAll();
        panelPieChart.add(barChartPanel, BorderLayout.CENTER);
        panelPieChart.validate();
    }

    private void ExitMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
            JFrame frame = new JFrame("EXIT");
       if(JOptionPane.showConfirmDialog(frame,"Are you sure you want to Exit?", "Exit",
JOptionPane.YES_NO_OPTION)==JOptionPane.YES_NO_OPTION)
       {
          System.exit(0);
       }
    }

    private void lbl_ManageEmployeesMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        Employees employees = new Employees();
        employees.setVisible(true);
        dispose();
    }

    private void lbl_ManageEmployeesMouseEntered(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        PanelEmployees.setBackground(EnterColor);
    }

    private void lbl_ManageEmployeesMouseExited(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        PanelEmployees.setBackground(ExitColor);
    }

    private void lbl_ContactSuppliersMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        Supplier suppliers = new Supplier();
        suppliers.setVisible(true);
        dispose();
    }

    private void lbl_ContactSuppliersMouseEntered(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        PanelContactSuppliers.setBackground(EnterColor);
    }

    private void lbl_ContactSuppliersMouseExited(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        PanelContactSuppliers.setBackground(ExitColor);
    }
```

```
private void lbl_AddNewCategoryMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    AddCategory categories = new AddCategory();
    categories.setVisible(true);
    dispose();
}

private void lbl_AddNewCategoryMouseEntered(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    PanelAddNewCategory.setBackground(EnterColor);
}

private void lbl_AddNewCategoryMouseExited(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    PanelAddNewCategory.setBackground(ExitColor);
}

private void lbl_AddNewProductMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    AddProduct products = new AddProduct();
    products.setVisible(true);
    dispose();
}

private void lbl_AddNewProductMouseEntered(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    PanelAddNewProduct.setBackground(EnterColor);
}

private void lbl_AddNewProductMouseExited(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    PanelAddNewProduct.setBackground(ExitColor);
}

private void lbl_ManageProductsMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    ManageProduct manageproducts = new ManageProduct();
    manageproducts.setVisible(true);
    dispose();
}

private void lbl_ManageProductsMouseEntered(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    PanelManageProducts.setBackground(EnterColor);
}

private void lbl_ManageProductsMouseExited(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    PanelManageProducts.setBackground(ExitColor);
```

```
}

private void lbl_LogoutMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    HomePage homepage = new HomePage();
    homepage.setVisible(true);
    dispose();
}

private void lbl_LogoutMouseEntered(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    PanelLogout.setBackground(LogoutEnterColor);
}

private void lbl_LogoutMouseExited(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    PanelLogout.setBackground(LogoutExitColor);
}

// Variables declaration - do not modify
private javax.swing.JLabel Exit;
private javax.swing.JTable JTableCustomer;
private javax.swing.JTable JTableMessage;
private javax.swing.JTable JTableSales;
private javax.swing.JPanel PanelAddNewCategory;
private javax.swing.JPanel PanelAddNewProduct;
private javax.swing.JPanel PanelContactSuppliers;
private javax.swing.JPanel PanelEmployees;
private javax.swing.JPanel PanelLogout;
private javax.swing.JPanel PanelManageEmployees;
private javax.swing.JPanel PanelManageProducts;
private javax.swing.JLabel date;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel17;
private javax.swing.JLabel jLabel19;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel21;
private javax.swing.JLabel jLabel23;
private javax.swing.JLabel jLabel24;
private javax.swing.JLabel jLabel25;
private javax.swing.JLabel jLabel26;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
```

```
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel10;
private javax.swing.JPanel jPanel11;
private javax.swing.JPanel jPanel12;
private javax.swing.JPanel jPanel13;
private javax.swing.JPanel jPanel14;
private javax.swing.JPanel jPanel15;
private javax.swing.JPanel jPanel17;
private javax.swing.JPanel jPanel18;
private javax.swing.JPanel jPanel19;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel22;
private javax.swing.JPanel jPanel23;
private javax.swing.JPanel jPanel3;
private javax.swing.JPanel jPanel4;
private javax.swing.JPanel jPanel5;
private javax.swing.JPanel jPanel6;
private javax.swing.JPanel jPanel7;
private javax.swing.JPanel jPanel8;
private javax.swing.JPanel jPanel9;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JScrollPane jScrollPane3;
private javax.swing.JSeparator jSeparator1;
private javax.swing.JSeparator jSeparator2;
private javax.swing.JLabel lbl_AddNewCategory;
private javax.swing.JLabel lbl_AddNewProduct;
private javax.swing.JLabel lbl_Categories;
private javax.swing.JLabel lbl_ContactSuppliers;
private javax.swing.JLabel lbl_Employees;
private javax.swing.JLabel lbl_Logout;
private javax.swing.JLabel lbl_ManageEmployees;
private javax.swing.JLabel lbl_ManageProducts;
private javax.swing.JLabel lbl_Products;
private javax.swing.JPanel panelLineChart;
private javax.swing.JPanel panelPieChart;
private javax.swing.JLabel time;
// End of variables declaration

@Override
public void run() { // Clock
  while(true){
     try{
        Calendar c = Calendar.getInstance();
        minute = c.get(Calendar.MINUTE);
        second = c.get(Calendar.SECOND);
        hour = c.get(Calendar.HOUR_OF_DAY);

        if(hour>12){
```

```
                hour = hour-12;
            }

            year = c.get(Calendar.YEAR);
            month = c.get(Calendar.MONTH);
            day = c.get(Calendar.DAY_OF_MONTH);

            SimpleDateFormat sdf = new SimpleDateFormat("hh:mm:ss a");
            SimpleDateFormat df = new SimpleDateFormat("dd/MM/yyyy");

            Date dat = c.getTime();
            timestr = sdf.format(dat);
            yearstr = df.format(dat);
            date.setText(yearstr);
            time.setText(timestr);

        }catch(Exception e){
           // e.printStackTrace();
        }
     }
   }
}
```

4 - *CustomerPage.java*

```
 package JFrames;

import java.awt.Color;
import javax.swing.JFrame;
import javax.swing.JOptionPane;

/**
 *
 * @author joni
 */
public class CustomerPage extends javax.swing.JFrame {
                                //inheritence

Color EnterColor = new Color(0,0,0);
Color ExitColor = new Color(78,78,78);
Color LogoutEnterColor = new Color(86,85,167);
Color LogoutExitColor = new Color(129,129,243);
    /**
     * Creates new form AdminPage
     */
    public CustomerPage() {
        initComponents();
    }
```

```
    private void exitBtnMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
            JFrame frame = new JFrame("EXIT");
       if(JOptionPane.showConfirmDialog(frame,"Are you sure you want to Exit?", "Exit",
JOptionPane.YES_NO_OPTION)==JOptionPane.YES_NO_OPTION)
        {
            System.exit(0);
        }
    }

    private void lbl_FoodMenuMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        Food food = new Food();
        food.setVisible(true);
        dispose();
    }

    private void lbl_FoodMenuMouseEntered(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        PanelEmployees.setBackground(EnterColor);
    }

    private void lbl_FoodMenuMouseExited(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        PanelEmployees.setBackground(ExitColor);
    }

    private void lbl_BeverageMenuMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        Beverages bev = new Beverages();
        bev.setVisible(true);
        dispose();
    }

    private void lbl_BeverageMenuMouseEntered(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        PanelContactSuppliers.setBackground(EnterColor);
    }

    private void lbl_BeverageMenuMouseExited(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        PanelContactSuppliers.setBackground(ExitColor);
    }

    private void lbl_PlaceOrderMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        Orders ord = new Orders();
        ord.setVisible(true);
        dispose();
```

```
    }

    private void lbl_PlaceOrderMouseEntered(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        PanelAddNewCategory.setBackground(EnterColor);
    }

    private void lbl_PlaceOrderMouseExited(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        PanelAddNewCategory.setBackground(ExitColor);
    }

    private void lbl_LeaveReviewMouseExited(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        PanelAddNewProduct.setBackground(ExitColor);
    }

    private void lbl_LeaveReviewMouseEntered(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        PanelAddNewProduct.setBackground(EnterColor);
    }

    private void lbl_LeaveReviewMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        Reviews rev = new Reviews();
        rev.setVisible(true);
        dispose();
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
```

```
java.util.logging.Logger.getLogger(CustomerPage.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(CustomerPage.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(CustomerPage.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(CustomerPage.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        }
        //</editor-fold>
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new CustomerPage().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JPanel PanelAddNewCategory;
    private javax.swing.JPanel PanelAddNewProduct;
    private javax.swing.JPanel PanelContactSuppliers;
    private javax.swing.JPanel PanelEmployees;
    private javax.swing.JPanel PanelManageEmployees;
    private javax.swing.JLabel exitBtn;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel11;
    private javax.swing.JLabel jLabel12;
    private javax.swing.JLabel jLabel13;
    private javax.swing.JLabel jLabel14;
    private javax.swing.JLabel jLabel15;
    private javax.swing.JLabel jLabel16;
    private javax.swing.JLabel jLabel17;
    private javax.swing.JLabel jLabel18;
    private javax.swing.JLabel jLabel19;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel20;
    private javax.swing.JLabel jLabel21;
    private javax.swing.JLabel jLabel22;
```

```
    private javax.swing.JLabel jLabel23;
    private javax.swing.JLabel jLabel24;
    private javax.swing.JLabel jLabel25;
    private javax.swing.JLabel jLabel26;
    private javax.swing.JLabel jLabel27;
    private javax.swing.JLabel jLabel28;
    private javax.swing.JLabel jLabel29;
    private javax.swing.JLabel jLabel30;
    private javax.swing.JLabel jLabel31;
    private javax.swing.JLabel jLabel32;
    private javax.swing.JLabel jLabel33;
    private javax.swing.JLabel jLabel34;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel12;
    private javax.swing.JPanel jPanel17;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPanel jPanel3;
    private javax.swing.JPanel jPanel4;
    private javax.swing.JPanel jPanel5;
    private javax.swing.JPanel jPanel6;
    private javax.swing.JSeparator jSeparator1;
    private javax.swing.JSeparator jSeparator2;
    private javax.swing.JSeparator jSeparator3;
    private javax.swing.JLabel lbl_BeverageMenu;
    private javax.swing.JLabel lbl_FoodMenu;
    private javax.swing.JLabel lbl_LeaveReview;
    private javax.swing.JLabel lbl_PlaceOrder;
    private javax.swing.JPanel panelLineChart;
    private javax.swing.JPanel panelPieChart;
    // End of variables declaration
}
```

5 - Employees.java

```
package JFrames;

import Java.Employee;
import java.sql.Statement;
import com.mysql.jdbc.Driver;
import java.awt.Color;
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.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.swing.ImageIcon;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
import javax.swing.RowFilter;
import javax.swing.filechooser.FileNameExtensionFilter;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableModel;
import javax.swing.table.TableRowSorter;

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */
/**
 *
 * @author joni
 */
public class Employees extends javax.swing.JFrame {

    Color ExitBack = new Color(74, 27, 1);
    Color EnterBack = new Color(92, 46, 1);

    Connection con = null;
    ResultSet rs = null;
    PreparedStatement pst = null;

    /**
     * Creates new form Employees
     */
    public Employees() {
        initComponents();
        this.setLocationRelativeTo(null);
        ShowEmployeeInJTable();
        addBtn.setEnabled(false);
        updateBtn.setEnabled(false);
```

```
        removeBtn.setEnabled(false);
        clearBtn.setEnabled(false);
        // Column sizes for the jTable
        JTableEmployee.getColumnModel().getColumn(0).setPreferredWidth(25);
        JTableEmployee.getColumnModel().getColumn(1).setPreferredWidth(60);
        JTableEmployee.getColumnModel().getColumn(2).setPreferredWidth(100);
        JTableEmployee.getColumnModel().getColumn(3).setPreferredWidth(68);
        JTableEmployee.getColumnModel().getColumn(4).setPreferredWidth(60);
        JTableEmployee.getColumnModel().getColumn(5).setPreferredWidth(30);
        JTableEmployee.getColumnModel().getColumn(6).setPreferredWidth(50);
        JTableEmployee.getColumnModel().getColumn(7).setPreferredWidth(30);
        JTableEmployee.getColumnModel().getColumn(8).setPreferredWidth(150);

    }

    String ImgPath = null;

    int pos = 0; // Position of record in table declaration
                 // This will be used for the navigation buttons to go from one employee to another.
    public Connection getConnection() {
        Connection con = null;
        try {
            con = DriverManager.getConnection("jdbc:mysql://localhost:3307/Restaurant", "root", "");
            return con;
        } catch (SQLException ex) {
            Logger.getLogger(Employees.class.getName()).log(Level.SEVERE, null, ex);
            return null;

        }
    }

    // Validation rules that must be satisfied in order to be able to add and update an empolyee
    public void validateField() {
        String Name = txtName.getText();
        String Email = txtEmail.getText();
        String Number = txtPhoneNumber.getText();
        String Salary = txtSalary.getText();
        String Address = txtAddress.getText();

        if (!Name.equals("") && !Email.equals("") && !Number.equals("") && !Salary.equals("") &&
!Address.equals("")) {
            addBtn.setEnabled(true);
        } else {
            addBtn.setEnabled(false);
        }
    }

    public void validateFields() {
        String Name = txtName.getText();
```

```
        String Email = txtEmail.getText();
        String Number = txtPhoneNumber.getText();
        String Salary = txtSalary.getText();
        String Address = txtAddress.getText();

        if (!Name.equals("") && !Email.equals("") && !Number.equals("") && !Salary.equals("") &&
!Address.equals("")) {
            updateBtn.setEnabled(true);
        } else {
            updateBtn.setEnabled(false);
        }
    }

    public void validateFieldss() {
        String Name = txtName.getText();
        String Email = txtEmail.getText();
        String Number = txtPhoneNumber.getText();
        String Salary = txtSalary.getText();
        String Address = txtAddress.getText();

        if (!Name.equals("") && !Email.equals("") && !Number.equals("") && !Salary.equals("") &&
!Address.equals("")) {
            removeBtn.setEnabled(true);
        } else {
            removeBtn.setEnabled(false);
        }
    }

    public void validateFieldsss() {
        String Name = txtName.getText();
        String Email = txtEmail.getText();
        String Number = txtPhoneNumber.getText();
        String Salary = txtSalary.getText();
        String Address = txtAddress.getText();

        if (!Name.equals("") || !Email.equals("") || !Number.equals("") || !Salary.equals("") ||
!Address.equals("")) {
            clearBtn.setEnabled(true);
        } else {
            clearBtn.setEnabled(false);
        }
    }

//  Credits: Maurice Muteti on YouTube whose code I modified from this video
https://youtu.be/33jZhuO3yeE
//  Using an ArrayList to store the data from the MySQL database 'Employee' into our jTable here.
//  We are also using the getters from our class 'Employee' to perform this action.
    public ArrayList<Employee> getEmployeeList() {
```

```
        ArrayList<Employee> employeeList = new ArrayList<Employee>(); //import the java.util.ArrayList;
        Connection con = getConnection();
        String query = "SELECT * FROM `Employee`"; //the * saves us the trouble of having to type all the
column names.

        Statement st; //import the java.sql.Statement
        ResultSet rs; //import the java.sql.ResultSet

        try {
            st = con.createStatement();
            rs = st.executeQuery(query);
            Employee employee;
            while (rs.next()) { //this method loops through results so provide the getters inside the 'Employee'
object.

                employee = new Employee(
                        rs.getInt("WorkerID"),
                        rs.getString("Name"),
                        rs.getString("Email"),
                        rs.getString("DOB"),
                        rs.getInt("PhoneNumber"),
                        rs.getString("Gender"),
                        rs.getString("Position"),
                        Float.parseFloat(rs.getString("Salary")), //needs to be written this way due to the data type.
                        rs.getString("Address"),
                        rs.getBytes("Photo")
                );
                employeeList.add(employee); //add to the list the employee.
            }
        } catch (SQLException ex) {
            Logger.getLogger(Employees.class.getName()).log(Level.SEVERE, null, ex);
        }
        return employeeList;
    }

    //Populating the jtable using the ArrayList
    public void ShowEmployeeInJTable() {

        ArrayList<Employee> list = getEmployeeList(); //Calling the method for the ArrayList above.

        DefaultTableModel model = (DefaultTableModel) JTableEmployee.getModel(); //import the
java.util.DefaultTableModel.

        model.setRowCount(0); //So that it doesn't add all the previous jTable data again but only the current
data being entered.
        Object[] row = new Object[9]; //The number '9' represents the number of rows.

        for (int i = 0; i < list.size(); i-=-1) { //for loop iterates through the object's rows when less than 9. 'i' is
the index.
```

```
            row[0] = list.get(i).getWorkerID();
            row[1] = list.get(i).getName();
            row[2] = list.get(i).getEmail();
            row[3] = list.get(i).getDOB();
            row[4] = list.get(i).getPhoneNumber();
            row[5] = list.get(i).getGender();
            row[6] = list.get(i).getPosition();
            row[7] = list.get(i).getSalary();
            row[8] = list.get(i).getAddress();
            model.addRow(row);
        }
         JTableEmployee.setRowHeight(30);
    }

    public void showEmployee(int index) {

        txtWorkerID.setText(Integer.toString(getEmployeeList().get(index).getWorkerID()));
        txtName.setText(getEmployeeList().get(index).getName());
        txtEmail.setText(getEmployeeList().get(index).getEmail());

        try {
            Date DOB = null;
            DOB = new SimpleDateFormat("yyyy-MM-dd").parse((String)
getEmployeeList().get(index).getDOB());
            txtDOB.setDate(DOB);
        } catch (ParseException ex) {
            Logger.getLogger(Employees.class.getName()).log(Level.SEVERE, null, ex);
        }

        txtPhoneNumber.setText(Integer.toString(getEmployeeList().get(index).getPhoneNumber()));

        String Gender = getEmployeeList().get(index).getGender();
        if (Gender.equals("Male")) {
            male.setSelected(true);
            female.setSelected(false);
        } else if (Gender.equals("Female")) {
            female.setSelected(true);
            male.setSelected(false);
        }

        txtPosition.setSelectedItem(getEmployeeList().get(index).getPosition());
        txtSalary.setText(Float.toString(getEmployeeList().get(index).getSalary()));
        txtAddress.setText(getEmployeeList().get(index).getAddress());
        lbl_img.setIcon(ResizeImage(null, getEmployeeList().get(index).getPhoto()));
    }

    public void Connect() {

        try {
```

```
        Class.forName("com.mysql.cj.jdbc.Driver");
        con = DriverManager.getConnection("jdbc:mysql://localhost:3307/Restaurant", "root", "");

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

// Check input fields verification
public boolean checkInputs() {
    if (txtName.getText() == null || txtEmail.getText() == null
            || txtDOB.getDate() == null || txtPhoneNumber.getText() == null
            || txtPosition.getSelectedItem().toString() == null
            || txtSalary.getText() == null || txtAddress.getText() == null) {
        return false;
    } else {
        try {
            Float.parseFloat(txtSalary.getText());
            return true;
        } catch (Exception ex) {
            return false;
        }
    }
}

private void maleActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    gender = "Male";
    male.setSelected(true);
    female.setSelected(false);
}

private void femaleActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    gender = "Male";
    male.setSelected(false);
    female.setSelected(true);
}

private void addBtnActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    if (checkInputs() && ImgPath != null) {

        try {
            Connection con = getConnection();
```

```
            PreparedStatement ps = con.prepareStatement("insert into Employee (Name, Email, DOB,
PhoneNumber, Gender, Position, Salary, Address, Photo)"
                    + "values(?,?,?,?,?,?,?,?,?)");
            ps.setString(1, txtName.getText());
            ps.setString(2, txtEmail.getText());

            SimpleDateFormat SDF = new SimpleDateFormat("yyyy-MM-dd");
            String DOB = SDF.format(txtDOB.getDate());
            ps.setString(3, DOB);

            ps.setString(4, txtPhoneNumber.getText());

            if (male.isSelected()) {
                gender = "Male";
            } else if (female.isSelected()) {
                gender = "Female";
            }

            ps.setString(5, gender);

            ps.setString(6, txtPosition.getSelectedItem().toString());
            ps.setString(7, txtSalary.getText());
            ps.setString(8, txtAddress.getText());

            try {
                InputStream img = new FileInputStream(new File(ImgPath));
                ps.setBlob(9, img);
                ps.executeUpdate();
                ShowEmployeeInJTable();

                JOptionPane.showMessageDialog(null, "Employee Added!");

            } catch (Exception ex) {
                JOptionPane.showMessageDialog(null, ex.getMessage());
                Logger.getLogger(Employees.class.getName()).log(Level.SEVERE, null, ex);
            }
        } catch (SQLException ex) {
            Logger.getLogger(Employees.class.getName()).log(Level.SEVERE, null, ex);
            JOptionPane.showMessageDialog(null, "Employee Not Added!");

        }
    } else {
        JOptionPane.showMessageDialog(null, "Employee already exists!");
    }

    // Display the inputted results as a HashMap would
    System.out.println("Name =>" + txtName.getText());
    System.out.println("Email =>" + txtEmail.getText());
    System.out.println("PhoneNumber =>" + txtPhoneNumber.getText());
```

```
        System.out.println("Gender =>" + gender);
        System.out.println("Position =>" + txtPosition.getSelectedItem().toString());
        System.out.println("Salary =>" + txtSalary.getText());
        System.out.println("Address =>" + txtAddress.getText());
        System.out.println("Photo url =>" + ImgPath);

        // Reset/empty feilds after an employee is added
        txtWorkerID.setText("");
        txtName.setText("");
        txtEmail.setText("");
        txtDOB.setDate(null);
        txtPhoneNumber.setText("");
        male.setSelected(false);
        female.setSelected(false);
        txtPosition.setSelectedIndex(0);
        txtSalary.setText("");
        lbl_img.setIcon(null);
        txtAddress.setText("");
        txtSearch.setText("");

    }

    private void txtSearchKeyReleased(java.awt.event.KeyEvent evt) {
        //Searches (Filters) for an Employee using the 'Search' jtextfield.
        //Credit Noureddine on Youtube. Video link: https://youtu.be/DJEXpgLyAtQ
        DefaultTableModel dtm = (DefaultTableModel) JTableEmployee.getModel(); //import the
javax.swing.table.DefaultTableModel.
        TableRowSorter<DefaultTableModel> tr = new TableRowSorter<DefaultTableModel>(dtm); //import
javax.swing.table.TableRowSorter.
        String search = txtSearch.getText();
        JTableEmployee.setRowSorter(tr);
        tr.setRowFilter(RowFilter.regexFilter(search)); //import javax.swing.RowFilter.

    }

    private void removeBtnActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        if (!txtWorkerID.getText().equals("")) {
            Connection con = getConnection();
            try {
                PreparedStatement ps = con.prepareStatement("delete FROM Employee where WorkerID =
?");
                int WorkerID = Integer.parseInt(txtWorkerID.getText());
                ps.setInt(1, WorkerID);
                ps.executeUpdate();
                ShowEmployeeInJTable();

                JOptionPane.showMessageDialog(null, "Employee Removed!");
```

```
            } catch (SQLException ex) {
                Logger.getLogger(Employees.class.getName()).log(Level.SEVERE, null, ex);
                JOptionPane.showMessageDialog(null, "Employee Not Removed!");

            }

        }

    }

    private void txtPhoneNumberKeyTyped(java.awt.event.KeyEvent evt) {
        // Validation: Only numbers
        if (!Character.isDigit(evt.getKeyChar())) {
            evt.consume();
        }
    }
    //Source: https://youtu.be/mdPNwrQ8kxU 1BestCsharp blog on YouTube.
    //Resize image to fit any dimention into the jLabel without distorting its quality
    public ImageIcon ResizeImage(String imagePath, byte[] pic) {
        ImageIcon myImage = null; //import javax.swing.ImageIcon

        if (imagePath != null) { //We initialized the imagePath String to be "null"
            myImage = new ImageIcon(imagePath); //If null show the image's filepath
        } else {
            myImage = new ImageIcon(pic);//Otherwise show the picture
        }
        Image img = myImage.getImage();
        Image img2 = img.getScaledInstance(lbl_img.getWidth(), lbl_img.getHeight(),
Image.SCALE_SMOOTH);
        ImageIcon image = new ImageIcon(img2);
        return image; //Show the output of the image on the jLabel "lbl_img".
    }

    private void ImageBtnActionPerformed(java.awt.event.ActionEvent evt) {
        //When the image button is pressed.
        //import javax.swing.JFileChooser to allow importing a file onto the jLabel
        JFileChooser file = new JFileChooser();// JFileChooser method allows the user to choose any file
they want.
        file.setCurrentDirectory(new File(System.getProperty("user.home")));// Redirects the user to their
home page.
                                                    // Can also be written as "user.dir"
        //Filter that restricts the users choices only for image files to be selected.
        FileNameExtensionFilter filter = new FileNameExtensionFilter("*.image", "jpg", "png");
        file.addChoosableFileFilter(filter); //import javax.swing.filechooser.FileNameExtensionFilter
        int result = file.showSaveDialog(null);//The method showSaveDialog() returns "0" if is successfull.
        if (result == JFileChooser.APPROVE_OPTION) { //If the user selects an image file, their option will
be approved.
            File selectedFile = file.getSelectedFile();
            String path = selectedFile.getAbsolutePath(); //The filepath of the image file
```

```
            lbl_img.setIcon(ResizeImage(path, null));
            ImgPath = path;
        } else { //Error message if user tries selecting a file that is not an image file.
            System.out.println("Please select an image file!");
        }
    }

    private void updateBtnActionPerformed(java.awt.event.ActionEvent evt) {
        // Updates an existing users information
        // Source: 1BestCsharp blog https://youtu.be/e-hAIY9QtqA
        if (checkInputs() && txtWorkerID.getText() != null) {
            String UpdateQuery = null;
            PreparedStatement ps = null;
            Connection con = getConnection();

            if (ImgPath == null) {
                try {
                    UpdateQuery = "update Employee set Name = ?, Email = ?, DOB = ?, PhoneNumber = ?,
Gender = ?, Position = ?, Salary = ?, Address = ?, Photo = ? where WorkerID = ?";

                    ps = con.prepareStatement(UpdateQuery);

                    ps.setString(1, txtName.getText());
                    ps.setString(2, txtEmail.getText());

                    SimpleDateFormat SDF = new SimpleDateFormat("yyyy-MM-dd");
                    String DOB = SDF.format(txtDOB.getDate());

                    ps.setString(3, DOB);

                    ps.setString(4, txtPhoneNumber.getText());

                    if (male.isSelected()) {
                        gender = "Male";
                    } else if (female.isSelected()) {
                        gender = "Female";
                    }
                    ps.setString(5, gender);

                    ps.setString(6, txtPosition.getSelectedItem().toString());
                    ps.setString(7, txtSalary.getText());
                    ps.setString(8, txtAddress.getText());

                    ps.setInt(9, Integer.parseInt(txtWorkerID.getText()));

                    ps.executeUpdate();
                    ShowEmployeeInJTable();

                    JOptionPane.showMessageDialog(this, "Employee Information Updated!");
```

```
            } catch (SQLException ex) {
                Logger.getLogger(Employees.class.getName()).log(Level.SEVERE, null, ex);
                JOptionPane.showMessageDialog(null, "Employee Information Not Updated!");

            }
        }//for updating image
        else {

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

                UpdateQuery = "update Employee set Name = ?, Email = ?, DOB = ?, PhoneNumber = ?,
Gender = ?, Position = ?, Salary = ?, Address = ?, Photo = ? where WorkerID = ?";

                ps = con.prepareStatement(UpdateQuery);

                ps.setString(1, txtName.getText());
                ps.setString(2, txtEmail.getText());

                SimpleDateFormat SDF = new SimpleDateFormat("yyyy-MM-dd");
                String DOB = SDF.format(txtDOB.getDate());

                ps.setString(3, DOB);

                ps.setString(4, txtPhoneNumber.getText());

                if (male.isSelected()) {
                    gender = "Male";
                } else if (female.isSelected()) {
                    gender = "Female";
                }
                ps.setString(5, gender);

                ps.setString(6, txtPosition.getSelectedItem().toString());
                ps.setString(7, txtSalary.getText());
                ps.setString(8, txtAddress.getText());

                ps.setBlob(9, img);

                ps.setInt(10, Integer.parseInt(txtWorkerID.getText()));

                ps.executeUpdate();
                ShowEmployeeInJTable();

                JOptionPane.showMessageDialog(this, "Employee Information Updated!");

            } catch (Exception ex) {
                JOptionPane.showMessageDialog(null, ex.getMessage());
```

```
                Logger.getLogger(Employees.class.getName()).log(Level.SEVERE, null, ex);
                JOptionPane.showMessageDialog(null, "Employee Information Not Updated!");

            }
        }
    } else {
        JOptionPane.showMessageDialog(null, "Empty field(s) present!");
    }

}

private void JTableEmployeeMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:

    int index = JTableEmployee.getSelectedRow();
    showEmployee(index);

    addBtn.setEnabled(true);
    updateBtn.setEnabled(true);
    removeBtn.setEnabled(true);
    clearBtn.setEnabled(true);
}

private void txtSalaryActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtSalaryKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    if (!Character.isDigit(evt.getKeyChar())) {
        evt.consume();
    }
}

private void firstBtnActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    pos = 0;
    showEmployee(pos);
}

private void lastBtnActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    pos = getEmployeeList().size() - 1;
    showEmployee(pos);
}

private void nextBtnActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    pos++;
```

```
    if (pos >= getEmployeeList().size()) {
        pos = getEmployeeList().size() - 1;
    }
    showEmployee(pos);
}

private void previousBtnActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    pos--;

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

private void txtSearchMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    txtSearch.setText("");
}

private void clearBtnActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    txtWorkerID.setText("");
    txtName.setText("");
    txtEmail.setText("");
    txtDOB.setDate(null);
    txtPhoneNumber.setText("");
    male.setSelected(false);
    female.setSelected(false);
    txtPosition.setSelectedIndex(0);
    txtSalary.setText("");
    lbl_img.setIcon(null);
    txtAddress.setText("");
    txtSearch.setText("");


}

private void lblBackMouseEntered(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    panelBack.setBackground(ExitBack);
}

private void lblBackMouseExited(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    panelBack.setBackground(EnterBack);
}
```

```
private void lblBackMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    AdminPage Admin = new AdminPage();
    Admin.setVisible(true);
    dispose();
}

private void txtNameKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    validateField();
    validateFields();
    validateFieldss();
    validateFieldsss();
}

private void txtEmailKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    validateField();
    validateFields();
    validateFieldss();
    validateFieldsss();
}

private void txtDOBKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    validateField();
    validateFields();
    validateFieldss();
    validateFieldsss();
}

private void txtPhoneNumberKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    validateField();
    validateFields();
    validateFieldss();
    validateFieldsss();
}

private void txtSalaryKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    validateField();
    validateFields();
    validateFieldss();
    validateFieldsss();
}

private void txtAddressKeyReleased(java.awt.event.KeyEvent evt) {
```

```
    // TODO add your handling code here:
    validateField();
    validateFields();
    validateFieldss();
    validateFieldsss();
}

// Variables declaration - do not modify
private javax.swing.JButton ImageBtn;
private javax.swing.JTable JTableEmployee;
private javax.swing.JButton addBtn;
private javax.swing.ButtonGroup buttonGroup1;
private javax.swing.ButtonGroup buttonGroup2;
private javax.swing.ButtonGroup buttonGroup3;
private javax.swing.JButton clearBtn;
private javax.swing.JRadioButton female;
private javax.swing.JButton firstBtn;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel14;
private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel16;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JButton lastBtn;
private javax.swing.JLabel lblBack;
private javax.swing.JLabel lbl_img;
private javax.swing.JRadioButton male;
private javax.swing.JButton nextBtn;
private javax.swing.JPanel panelBack;
private javax.swing.JButton previousBtn;
private javax.swing.JButton removeBtn;
private javax.swing.JTextArea txtAddress;
private com.toedter.calendar.JDateChooser txtDOB;
private javax.swing.JTextField txtEmail;
private javax.swing.JTextField txtName;
private javax.swing.JTextField txtPhoneNumber;
private javax.swing.JComboBox<String> txtPosition;
```

```
    private javax.swing.JTextField txtSalary;
    private javax.swing.JTextField txtSearch;
    private javax.swing.JTextField txtWorkerID;
    private javax.swing.JButton updateBtn;
    // End of variables declaration

    // Declaration of image (BLOB) variables
    private ImageIcon format = null;

    String filename = null;

    byte[] person_image = null;

    private String gender;

    //private Connection getConnection() {
    //    throw new UnsupportedOperationException("Not supported yet."); // Generated from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
    //}
    private static class DbUtils {

        private static TableModel resultSetToTableMModel(ResultSet rs) {
            throw new UnsupportedOperationException("Not supported yet."); // Generated from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
        }

        private static TableModel resultSetToTableModel(ResultSet rs) {
            throw new UnsupportedOperationException("Not supported yet."); // Generated from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
        }

        public DbUtils() {
        }
    }
}
```

6 - HomePage.java

```
package JFrames;
/**
 *
 * @author joni
 */
public class HomePage extends javax.swing.JFrame {

    /**
     * Creates new form HomePage
```

```
     */
    public HomePage() {
        initComponents();
        this.setLocationRelativeTo(null);
    }

    private void customerBtnActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        CustomerPage cust = new CustomerPage();
        cust.setVisible(true);
        dispose();
    }

    private void adminBtnActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        Login login = new Login();
        login.setVisible(true);
        dispose();
    }

    // Variables declaration - do not modify
    private javax.swing.JButton adminBtn;
    private javax.swing.JButton customerBtn;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPanel jPanel3;
    private javax.swing.JPanel jPanel4;
    // End of variables declaration
}
```

7 - Login.java

```
package JFrames;

import com.sun.jdi.connect.spi.Connection;
import java.awt.Color;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.swing.JFrame;
import javax.swing.JOptionPane;

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
```

```
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */

/**
 *
 * @author joni
 */
public class Login extends javax.swing.JFrame {
 Connection con = null;
 PreparedStatement pst = null;
 ResultSet rs = null;
    /**
     * Creates new form Login
     */
    public Login() {
        initComponents();
    }

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

        try{
            Class.forName("com.mysql.cj.jdbc.Driver");

            java.sql.Connection
connect=DriverManager.getConnection("jdbc:mysql://localhost:3307/Restaurant","root","");
            String Sql= "Select * from login where Username=? and Password=?";
            PreparedStatement pst= connect.prepareStatement(Sql);
            pst.setString(1, Username.getText());
            pst.setString(2, Password.getText());
            ResultSet Rs= pst.executeQuery();

            if(Rs.next()){ // If username and password matched, redirect user to AdminPage
                 JOptionPane.showMessageDialog(null, "Welcome!");
                 AdminPage admin = new AdminPage();
                 admin.setVisible(true);
                 setVisible(false);
            } else{ // Otherwise error message below will appear, preventing the user from proceeding
                JOptionPane.showMessageDialog(null, "Username or Password is incorrect!");
            }
        }
        catch(Exception e){
            JOptionPane.showMessageDialog(null,e);
        }

    }

    private void resetActionPerformed(java.awt.event.ActionEvent evt) {
        Username.setText("");
        Password.setText("");
```

```
    }

    private void showPassActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        // Hides the characters being typed in the password field unless 'show password' is enabled
        if(showPass.isSelected()){
            Password.setEchoChar((char)0);
        } else{
            Password.setEchoChar('*');
        }
    }

    private void ExitActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        // Exits frame and the JOption Pane prompts the confirmation to exit
        JFrame frame = new JFrame("EXIT");
       if(JOptionPane.showConfirmDialog(frame,"Are you sure you want to Exit?", "Exit",
JOptionPane.YES_NO_OPTION)==JOptionPane.YES_NO_OPTION)
       {
          System.exit(0);
       }
    }

    private void UsernameFocusGained(java.awt.event.FocusEvent evt) {
        // TODO add your handling code here:
        String usernameVal = Username.getText().trim().toLowerCase();
        if (usernameVal.equals("username")){
            Username.setText("");
            Username.setForeground(Color.black);
        }
    }

    private void UsernameFocusLost(java.awt.event.FocusEvent evt) {
        // TODO add your handling code here:
        String usernameVal = Username.getText().trim().toLowerCase();
        if (usernameVal.equals("username") || usernameVal.equals("")){
            Username.setText("username");
            Username.setForeground(new Color(204,204,255));
        }
    }

    private void PasswordFocusGained(java.awt.event.FocusEvent evt) {
        // TODO add your handling code here:
        String passwordVal = String.valueOf(Password.getPassword()).trim().toLowerCase();
        if (passwordVal.equals("password")){
            Password.setText("");
            Password.setForeground(Color.black);
        }
    }
```

```
    private void PasswordFocusLost(java.awt.event.FocusEvent evt) {
        // TODO add your handling code here:
        String passwordVal = String.valueOf(Password.getPassword()).trim().toLowerCase();
        if (passwordVal.equals("password") || passwordVal.equals("")){
            Password.setText("password");
            Password.setForeground(new Color(204,204,255));
        }
    }

    // Variables declaration - do not modify
    private javax.swing.JButton Exit;
    private javax.swing.JPasswordField Password;
    private javax.swing.JTextField Username;
    private javax.swing.ButtonGroup buttonGroup1;
    private javax.swing.JInternalFrame jInternalFrame1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPanel jPanel3;
    private javax.swing.JSeparator jSeparator1;
    private javax.swing.JSeparator jSeparator2;
    private com.toedter.components.JSpinField jSpinField1;
    private javax.swing.JTextField jTextField3;
    private javax.swing.JButton login;
    private javax.swing.JButton reset;
    private javax.swing.JRadioButton showPass;
    // End of variables declaration
}
```

8 - ManageProduct.java

```
package JFrames;

import com.mysql.cj.jdbc.Driver;
import java.awt.Color;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Vector;
import java.util.logging.Level;
```

```
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */
/**
 *
 * @author joni
 */
public class ManageProduct extends javax.swing.JFrame {

    Color EnterBack = new Color(153,153,153);
    Color ExitBack = new Color(197,195,195);

    /**
     * Creates new form ManageProduct
     */
    public ManageProduct() {
        initComponents();
        categoryLoad();
        LoadTable();
        this.setLocationRelativeTo(null);
        updateBtn.setEnabled(false);
        removeBtn.setEnabled(false);
        clearBtn.setEnabled(false);
        // Column sizes for the jTable
        JTableProduct.getColumnModel().getColumn(0).setPreferredWidth(39);
        JTableProduct.getColumnModel().getColumn(1).setPreferredWidth(131);
        JTableProduct.getColumnModel().getColumn(2).setPreferredWidth(63);
        JTableProduct.getColumnModel().getColumn(3).setPreferredWidth(45);
        JTableProduct.getColumnModel().getColumn(4).setPreferredWidth(37);
        JTableProduct.getColumnModel().getColumn(5).setPreferredWidth(180);
        JTableProduct.setRowHeight(28);
    }

    Connection con = null;
    PreparedStatement pst = null;
    ResultSet rs = null;

    public void validateField() {
        String Name = txtName.getText();
        String Price = txtPrice.getText();
        String Qty = txtQty.getText();
        String Description = txtDescription.getText();
        if (!Name.equals("") && !Price.equals("") && !Qty.equals("") && !Description.equals("")) {
            updateBtn.setEnabled(true);
```

```
    } else {
        updateBtn.setEnabled(false);
    }
}

public void validateFields() {
    String Name = txtName.getText();
    String Price = txtPrice.getText();
    String Qty = txtQty.getText();
    String Description = txtDescription.getText();
    if (!Name.equals("") && !Price.equals("") && !Qty.equals("") && !Description.equals("")) {
        removeBtn.setEnabled(true);
    } else {
        removeBtn.setEnabled(false);
    }
}
   public void validateFieldss() {
    String Name = txtName.getText();
    String Price = txtPrice.getText();
    String Qty = txtQty.getText();
    String Description = txtDescription.getText();
    if (!Name.equals("") || !Price.equals("") || !Qty.equals("") || !Description.equals("")) {
        clearBtn.setEnabled(true);
    } else {
        clearBtn.setEnabled(false);
    }
}


public class CategoryItem {  //to call jComboBox from Category SQL DB

    int ID;
    String Name;

    public CategoryItem(int ID, String Name) {
        this.ID = ID;
        this.Name = Name;
    }

    public String toString() {
        return Name;
    }
}

private void categoryLoad() {
    try {
        Class.forName("com.mysql.cj.jdbc.Driver");
        con = DriverManager.getConnection("jdbc:mysql://localhost:3307/Restaurant", "root", "");
        pst = con.prepareStatement("select * from Category");
```

```
            rs = pst.executeQuery();
            txtCategory.removeAllItems();

            while (rs.next()) {
                txtCategory.addItem(new CategoryItem(rs.getInt(1), rs.getString(2)));
            }

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

    }

    public void LoadTable() {

        int c;

        try {
            pst = con.prepareStatement("select * from Product");
            rs = pst.executeQuery();

            com.mysql.cj.jdbc.result.ResultSetMetaData rsd = (com.mysql.cj.jdbc.result.ResultSetMetaData)
rs.getMetaData();
            c = rsd.getColumnCount();

            DefaultTableModel d = (DefaultTableModel) JTableProduct.getModel();
            d.setRowCount(0);

            while (rs.next()) {
                Vector v2 = new Vector();

                for (int i = 1; i <= c; i++) {
                    v2.add(rs.getString("ProductID"));
                    v2.add(rs.getString("Name"));
                    v2.add(rs.getString("Category"));
                    v2.add(rs.getString("Price"));
                    v2.add(rs.getString("Qty"));
                    v2.add(rs.getString("Description"));
                }
                d.addRow(v2);

            }

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

```
    }

    private void updateBtnActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:

        DefaultTableModel dtm = (DefaultTableModel) JTableProduct.getModel();

        int i = JTableProduct.getSelectedRow();

        int ProductID = Integer.parseInt(dtm.getValueAt(i, 0).toString());

        String Name = txtName.getText();
        String Category = txtCategory.getSelectedItem().toString();
        String Price = txtPrice.getText();
        String Qty = txtQty.getText();
        String Description = txtDescription.getText();

        try {

            pst = con.prepareStatement("update Product set  Name = ?, Category = ?, Price = ?, Qty = ?,
Description = ? where ProductID = ?");
            pst.setString(1, Name);
            pst.setString(2, Category);
            pst.setString(3, Price);
            pst.setString(4, Qty);
            pst.setString(5, Description);
            pst.setInt(6, ProductID);

            int k = pst.executeUpdate();

            if (k == 1) {
                JOptionPane.showMessageDialog(this, "Product Updated!");

                txtName.setText("");
                txtCategory.setSelectedIndex(-1);
                txtPrice.setText("");
                txtQty.setText("");
                txtDescription.setText("");

                txtName.requestFocus();

                LoadTable();

            } else {
                JOptionPane.showMessageDialog(this, "Error! Product Not Updated!");
            }

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

```
    }


}

private void JTableProductMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    DefaultTableModel dtm = (DefaultTableModel) JTableProduct.getModel();
    int i = JTableProduct.getSelectedRow();

    int ProductID = Integer.parseInt(dtm.getValueAt(i, 0).toString());
    txtName.setText(dtm.getValueAt(i, 1).toString());
    txtCategory.setSelectedItem(dtm.getValueAt(i, 2).toString());
    txtPrice.setText(dtm.getValueAt(i, 3).toString());
    txtQty.setText(dtm.getValueAt(i, 4).toString());
    txtDescription.setText(dtm.getValueAt(i, 5).toString());

    String Category = dtm.getValueAt(i, 2).toString();
    for(int j=0;j<=txtCategory.getItemCount();j++)
    {
        if(txtCategory.getItemAt(j).toString().equals(Category))
        {
            txtCategory.setSelectedIndex(j);
            break;
        }
    }

    updateBtn.setEnabled(true);
    removeBtn.setEnabled(true);
    clearBtn.setEnabled(true);
}

private void clearBtnActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    txtName.setText("");
    txtCategory.setSelectedIndex(-1);
    txtPrice.setText("");
    txtQty.setText("");
    txtDescription.setText("");
}

private void removeBtnActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    DefaultTableModel dtm = (DefaultTableModel)JTableProduct.getModel();
    int i = JTableProduct.getSelectedRow();

    int ProductID = Integer.parseInt(dtm.getValueAt(i, 0).toString());
```

```
        int dialogResult = JOptionPane.showConfirmDialog(null, "Do you want to remove this
product?","Warning",JOptionPane.YES_OPTION);

        if(dialogResult == JOptionPane.YES_OPTION){

        try {

            pst = con.prepareStatement("delete from Product where ProductID = ?");

            pst.setInt(1, ProductID);

            int k = pst.executeUpdate();

            if(k==1){
                JOptionPane.showMessageDialog(this, "Product Deleted!");

                txtName.setText("");
                txtCategory.setSelectedIndex(-1);
                txtPrice.setText("");
                txtQty.setText("");
                txtDescription.setText("");

                txtName.requestFocus();

                LoadTable();

            }
            else {
                JOptionPane.showMessageDialog(this, "Error! Product Not Deleted!");
            }

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

    private void txtNameKeyReleased(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
        validateField();
        validateFields();
        validateFieldss();
    }

    private void txtPriceKeyReleased(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
        validateField();
        validateFields();
        validateFieldss();
```

```
    }

    private void txtDescriptionKeyReleased(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
        validateField();
        validateFields();
        validateFieldss();
    }
//
    private void JTableProductKeyReleased(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
    }

    private void txtQtyKeyReleased(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
        validateField();
        validateFields();
        validateFieldss();
    }

    private void lblBackMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        AdminPage Admin = new AdminPage();
        Admin.setVisible(true);
        dispose();
    }

    private void lblBackMouseEntered(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
    }

    private void lblBackMouseExited(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
    }

    private void txtPriceKeyTyped(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
            if (!Character.isDigit(evt.getKeyChar())) {
            evt.consume();
        }
    }

    private void txtQtyKeyTyped(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
            if (!Character.isDigit(evt.getKeyChar())) {
            evt.consume();
        }
    }
```

```
    // Variables declaration - do not modify
    private javax.swing.JTable JTableProduct;
    private javax.swing.JButton clearBtn;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPanel jPanel3;
    private javax.swing.JPanel jPanel4;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JScrollPane jScrollPane2;
    private javax.swing.JLabel lblBack;
    private javax.swing.JPanel panelBack;
    private javax.swing.JButton removeBtn;
    private javax.swing.JComboBox txtCategory;
    private javax.swing.JTextArea txtDescription;
    private javax.swing.JTextField txtName;
    private javax.swing.JTextField txtPrice;
    private javax.swing.JTextField txtQty;
    private javax.swing.JButton updateBtn;
    // End of variables declaration
}
```

9 - Orders.java

```
package JFrames;

import Java.MyQuery;
import java.awt.Color;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
import java.sql.SQLException;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
```

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */
/**
 *
 * @author joni
 */
public class Orders extends javax.swing.JFrame {

    Color EnterBack = new Color(78,30,1);
    Color ExitBack = new Color(88,50,32);

    /**
     * Creates new form Orders
     */
    public Orders() { // Constructor
        initComponents();
        this.setLocationRelativeTo(null);
        BindCombo();
        JTableOrder.setRowHeight(25);
    }

    Connection con = null;
    PreparedStatement pst = null;
    ResultSet rs = null;

    //Declare variables
    double Total;
    double Tax;
    double Subtotal;

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */

    public void BindCombo() { // Combobox setup
        MyQuery mq = new MyQuery();
        Connection con = mq.getConnection();
        Statement st;
        ResultSet rs;

        try {
            st = con.createStatement();
            rs = st.executeQuery("SELECT `CategoryID`, `Category` FROM `Category`");
            while (rs.next()) {
                txtCategory.addItem(rs.getString(2));
```

```
            }

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

    private void sales(){//Thank you TUTUSFUNNY @YouTube for this code which I used and modified to fit my project
                        //Link to video: https://youtu.be/KxLam4q2dF0 at 3 hours 30 minutes into the video.
        DateTimeFormatter dtf = DateTimeFormatter.ofPattern("yyyy/MM/dd");
        LocalDateTime now = LocalDateTime.now();

        String Date = dtf.format(now);
        String Totall = txtTotal.getText();
        String Pay = txtPay.getText();
        String Change = txtChange.getText();
        int LastInsertID = 0;

        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            con = DriverManager.getConnection("jdbc:mysql://localhost:3307/Restaurant", "root", "");
            String query = "insert into Sales (Date, Total, Paid, Balance) values (?,?,?,?) ";
            pst = con.prepareStatement(query, Statement.RETURN_GENERATED_KEYS);
                                        //Passing the query through the prepared statement them
                                        //Calling the 'RETURN_GENERATED_KEYS function
                                        //To get the value of the auto generated primary key 'CustomerID'
                                        //from the inserted query using the mysql jdbc driver

            pst.setString(1, Date);
            pst.setString(2, Totall);
            pst.setString(3, Pay);
            pst.setString(4, Change);
            pst.executeUpdate();

     //The function below generates a Last Insert ID for two tables. One keeps track of the purchased items and the other, the money spent.
    //This is helpful for when we want to add data simultaneously. It ensures multiple orders by a customer must all bellong to the same ID.

            ResultSet generatedKeyResult = pst.getGeneratedKeys();

            //Checking if key was generated successfully by displaying the customer's ordering ID through a JOptionPane

            if(generatedKeyResult.next()){
                LastInsertID = generatedKeyResult.getInt(1);
            }
            JOptionPane.showMessageDialog(this, LastInsertID);
```

```
        int rows = JTableOrder.getRowCount();

    String queryy = "insert into CustomerOrder (CustomerID, Name, Category, Price, Qty, Total) values
(?,?,?,?,?,?)";
    pst = con.prepareStatement(queryy);
    // Initialize
    String Category = "";
    String Name = "";
    int Price = 0;
    int Qty = 0;
    double Total = 0;

    for(int i = 0; i < JTableOrder.getRowCount(); i++){

        Category = (String)JTableOrder.getValueAt(i, 0);
        Name = (String)JTableOrder.getValueAt(i, 1);
        Price = (int)JTableOrder.getValueAt(i, 2);
        Qty = (int)JTableOrder.getValueAt(i, 3);
        Total = (double)JTableOrder.getValueAt(i, 4);

        pst.setInt(1,LastInsertID); //Getting the CustomersID from Sales table to CustomerOrder table
using the same ID.

        pst.setString(2, Name);
        pst.setString(3, Category);
        pst.setInt(4, Price);
        pst.setInt(5, Qty);
        pst.setDouble(6, Total);
        pst.executeUpdate();

    }
      // Reduces the item left in stock because of the updating requirement after purchasing
      String queryyy = "update Product set Qty = Qty-? where Name = ?";
       pst = con.prepareStatement(queryyy);

       for (int i = 0; i < JTableOrder.getRowCount(); i++) {

           Name = (String) JTableOrder.getValueAt(i, 1);
           Qty = (int) JTableOrder.getValueAt(i, 3);

           pst.setInt(1, Qty);
           pst.setString(2, Name);
           pst.execute();
       }

    pst.addBatch(); //Every insert statement is being addad as a batch
    JOptionPane.showMessageDialog(this, "Order Saved And Remaining Quantity Reduced!");
```

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

    public void PrintReceipt(){
    String SubTotal = txtSubtotal.getText();
    String Total = txtTotal.getText();
    String Pay = txtPay.getText();
    String Tax = txtTax.getText();
    String Change = txtChange.getText();
    new  PrintReceipt (SubTotal, Tax, Total, Pay, Change, JTableOrder.getModel()).setVisible(true);
    }


    private void calculateBtnActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        Change();
        sales();
    }

    private void pos() {
        String Name = txtName.getText();
        String Category = txtCategory.getSelectedItem().toString();
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            try {
                con = DriverManager.getConnection("jdbc:mysql://localhost:3307/Restaurant", "root", "");
                pst = con.prepareStatement("select * from Product where Name = ?");
                pst.setString(1, Name);
                ResultSet rs = pst.executeQuery();

                while (rs.next()) {

                    int Qty;
                    Qty = rs.getInt("Qty");

                    int Price = Integer.parseInt(txtPrice.getText());
                    int newQty = Integer.parseInt(txtQuantity.getText());

                    if (newQty >= Qty) { //Show the following error message and give the customer the option of
ordering less
                        JOptionPane.showMessageDialog(this, "You cannot order this amount of " + Name + " " +
Category + "s");
                        JOptionPane.showMessageDialog(this, "We currently have " + Qty + " " + Name + " " +
Category + "s that can be ordered!");
                    }else{ // Perform the following arithmetic
```

```
    Subtotal = (newQty * Price);
    Total = Total - Tax;
    Tax = Tax + Subtotal * 0.15; // 15% tax
    Total = Total + Subtotal + Tax;
    // Call the results on their respective fields
    txtSubtotal.setText(""+Subtotal);
    txtTax.setText(""+Tax);
    txtTotal.setText(""+Total);

    DefaultTableModel dtm = (DefaultTableModel) JTableOrder.getModel();

    dtm = (DefaultTableModel)JTableOrder.getModel();
    dtm.addRow(new Object[]{ // Adds the objects below in the order presented
        Category,
        Name,
        Price,
        newQty,
        Subtotal,
        Tax,
        Total,
    });

    double sum = 0; // Initialization of the total

    for(int i=0; i<JTableOrder.getRowCount(); i++){ // Loops through the 5 columns in the table
        sum = sum + Double.parseDouble(JTableOrder.getValueAt(i, 4).toString());
    }

    txtSubtotal.setText(Double.toString(sum));
                }}
        } catch (SQLException ex) {
            Logger.getLogger(Orders.class.getName()).log(Level.SEVERE, null, ex);
        }
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(Orders.class.getName()).log(Level.SEVERE, null, ex);
    }
}

public void Change(){ // The customer's return after payment for the bill

double Pay = Double.parseDouble(txtPay.getText());

double Change = Pay - Total;

txtChange.setText(String.valueOf(Change));

}
```

```
    private void bill() { // This would be an alternative way to making a receipt in a pdf format.
                          // I chose to forego this method and code for simplicity purposes.
        String path = "E:\\";
        com.itextpdf.text.Document doc = new com.itextpdf.text.Document();
    }

    private void txtNameKeyReleased(java.awt.event.KeyEvent evt) {
 //Credit: TutusFunny on YouTube: https://youtu.be/-MeWRuSVtDE 25 minutes and 45 seconds into the video.
//I wrote this code in the 'key released' event rather than 'key pressed' because it searches automatically as opposed to having to press 'enter'.
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            try {
                con = DriverManager.getConnection("jdbc:mysql://localhost:3307/Restaurant", "root", "");
                String Name = txtName.getText(); //
                            //The 'Name' field in the MySQL database 'Product' is what's being searched for to come in the jtextfield 'Name'.
                pst = con.prepareStatement("select * from Product where Name = ?");
                pst.setString(1, Name); //The number '1' is the position of the Name field in the table.
                ResultSet rs = pst.executeQuery();

                if (rs.next() == true) { //boolean condition check for the ResultSet
                    String Category = rs.getString(2); //'2' is the position of the string field 'Category'
                    String Price = rs.getString(4); //'4' is the position of the string field 'Price'
                    txtCategory.setSelectedItem(Category); //Returns the corresponding category when the name jtextfield is typed.
                    txtPrice.setText(Price); //Returns the corresponding price of the product when the name jtextfield is typed.

                } else { //otherwise show nothing.
                    txtCategory.setSelectedIndex(0); //clears the jcombobox.
                    txtPrice.setText(""); //clears the jtextfield.
                }

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

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

    }

    private void addBtnActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        pos();
        txtCategory.setSelectedIndex(0);
```

```
    txtName.setText("");
    txtPrice.setText("");
    txtQuantity.setText("");
    txtPayment.setSelectedIndex(0);
}

private void clearBtnActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    txtCategory.setSelectedIndex(0);
    txtName.setText("");
    txtPrice.setText("");
    txtQuantity.setText("");
    txtPayment.setSelectedIndex(0);
    txtSubtotal.setText("");
    txtTotal.setText("");
    txtTax.setText("");
    txtPay.setText("");
    txtChange.setText("");
}

private void printBtnActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    PrintReceipt();
}

private void lblBackMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    CustomerPage Cust = new CustomerPage();
    Cust.setVisible(true);
    dispose();
}

private void lblBackMouseEntered(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    panelBack.setBackground(EnterBack);
}

private void lblBackMouseExited(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    panelBack.setBackground(ExitBack);
}

private void txtQuantityKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
        if (!Character.isDigit(evt.getKeyChar())) {
        evt.consume();
    }
}
```

```
    // Variables declaration - do not modify
    private javax.swing.JTable JTableOrder;
    private javax.swing.JButton addBtn;
    private javax.swing.JButton calculateBtn;
    private javax.swing.JButton clearBtn;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel11;
    private javax.swing.JLabel jLabel12;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPanel jPanel3;
    private javax.swing.JPanel jPanel5;
    private javax.swing.JScrollPane jScrollPane2;
    private javax.swing.JLabel lblBack;
    private javax.swing.JPanel panelBack;
    private javax.swing.JButton printBtn;
    private javax.swing.JComboBox txtCategory;
    private javax.swing.JTextField txtChange;
    private javax.swing.JTextField txtName;
    private javax.swing.JTextField txtPay;
    private javax.swing.JComboBox<String> txtPayment;
    private javax.swing.JTextField txtPrice;
    private javax.swing.JTextField txtQuantity;
    private javax.swing.JTextField txtSubtotal;
    private javax.swing.JTextField txtTax;
    private javax.swing.JTextField txtTotal;
    // End of variables declaration
}
```

10 - PrintReceipt.java

```
package JFrames;

import java.awt.Font;
import java.awt.print.PrinterException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.logging.Level;
```

```
import java.util.logging.Logger;
import javax.swing.table.TableModel;

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */
/**
 *
 * @author joni
 */
public class PrintReceipt extends javax.swing.JFrame {

    /**
     * Creates new form PrintReceipt
     */
    public PrintReceipt() {
        initComponents();
    }

    String SubTotall;
    String Taxx;
    String Totall;
    String Payy;
    String Changee;

    public PrintReceipt(String SubTotal, String Tax, String Total, String Pay, String Change, TableModel
model) {

        initComponents();

        Calendar timer = Calendar.getInstance();
        timer.getTime();
        SimpleDateFormat Time = new SimpleDateFormat("HH:mm");
        Time.format(timer.getTime());
        SimpleDateFormat Date = new SimpleDateFormat("dd/MM/yyyy");

        this.SubTotall = SubTotal;
        this.Taxx = Tax;
        this.Totall = Total;
        this.Payy = Pay;
        this.Changee = Change;

        txtPrint.setText(txtPrint.getText() + " ---------------------------------------------\n");
        txtPrint.setText(txtPrint.getText() + "Date: " + Date.format(timer.getTime()) + "\t" + "                Time:
" + Time.format(timer.getTime()) + "\n");
        txtPrint.setText(txtPrint.getText() + "                                                \n");
        txtPrint.setText(txtPrint.getText() + "                          KOMARR PLC                  \n");
        txtPrint.setText(txtPrint.getText() + "                      A.A S/C ARADA W.07                  \n");
```

```
        txtPrint.setText(txtPrint.getText() + "                    H.NO NEW AROUND                    \n");
        txtPrint.setText(txtPrint.getText() + "                        SANDFORD                    \n");
        txtPrint.setText(txtPrint.getText() + "                       Amount due                   \n");
        txtPrint.setText(txtPrint.getText().formatted(Font.BOLD, 12f));
        txtPrint.setText(txtPrint.getText() + " ---------------------------------------------\n");
        txtPrint.setText(txtPrint.getText() + "Product" + "\t" +"Category"+ "\t" + "Price" + "\t" + "Subtotal" +
"\n");

        for (int i = 0; i < model.getRowCount(); i++) {
            String Category = (String) model.getValueAt(i, 0);
            String Name = (String) model.getValueAt(i, 1);
            int Price = (Integer) model.getValueAt(i, 2);
            double Amount = (Double) model.getValueAt(i, 4);
            txtPrint.setText(txtPrint.getText() + Name + "\t" + Category+ "\t" + Price + "\t" + Amount + "\n");
        }
        txtPrint.setText(txtPrint.getText() + "\n");
        txtPrint.setText(txtPrint.getText() + "\t" + "\t" + "Subtotal: " + SubTotal + "\n");
        txtPrint.setText(txtPrint.getText() + "\t" + "\t" + "Tax (15%): " + Tax + "\n");
        txtPrint.setText(txtPrint.getText() + "\t" + "\t" + "Taxed total: " + Total + "\n");
        txtPrint.setText(txtPrint.getText() + "\t" + "\t" + "Amount paid: " + Pay + "\n");
        txtPrint.setText(txtPrint.getText() + "\t" + "\t" + "Change: " + Change + "\n");
        txtPrint.setText(txtPrint.getText() + "\n");
        txtPrint.setText(txtPrint.getText() + " ---------------------------------------------\n");
        txtPrint.setText(txtPrint.getText() + "            Thank You For Dining At Olive & Bread!            \n");
        txtPrint.setText(txtPrint.getText() + "\n");
        txtPrint.setText(txtPrint.getText() + "                                ERCA                    \n");

        try {
            txtPrint.print();
        } catch (PrinterException ex) {
            Logger.getLogger(PrintReceipt.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    // Variables declaration - do not modify
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTextArea txtPrint;
    // End of variables declaration
```

11 - Reveiws.java

```
package JFrames;

import java.awt.Color;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
```

```
import java.sql.ResultSet;
import java.sql.SQLException;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;

/**
 *
 * @author joni
 */
public class Reviews extends javax.swing.JFrame {

    /**
     * Creates new form Reviews
     */
    public Reviews() {
        initComponents();
        this.setLocationRelativeTo(null);
    }

    Connection con = null;
    PreparedStatement pst = null;
    ResultSet rs = null;

    Color ExitBack = new Color(99,68,32);
    Color EnterBack = new Color(77,53,25);

        private void Message(){
        DateTimeFormatter dtf = DateTimeFormatter.ofPattern("yyyy/MM/dd"); // Current date
        LocalDateTime now = LocalDateTime.now(); // Current time

        String Date = dtf.format(now);
        String Message = txtMessage.getText();

        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            con = DriverManager.getConnection("jdbc:mysql://localhost:3307/Restaurant", "root", "");
            pst = con.prepareStatement("insert into CustomerReview (Date, Message) values (?,?)");

            pst.setString(1, Date);
            pst.setString(2, Message);

            int k = pst.executeUpdate();

            if(k==1){
                JOptionPane.showMessageDialog(null, "Message Posted!");
                txtMessage.setText("");
```

```
            }
            else{
                JOptionPane.showMessageDialog(this, "Error! Message Not Sent!");
            }
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(Orders.class.getName()).log(Level.SEVERE, null, ex);
        } catch (SQLException ex) {
            Logger.getLogger(Orders.class.getName()).log(Level.SEVERE, null, ex);
        }
            }

    private void messageBtnActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        Message();
    }

    private void backBtnMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        CustomerPage cust = new CustomerPage();
        cust.setVisible(true);
        dispose();
    }

    private void backBtnMouseEntered(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        panelBack.setBackground(ExitBack);
    }

    private void backBtnMouseExited(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        panelBack.setBackground(ExitBack);
    }

    // Variables declaration - do not modify
    private javax.swing.JLabel backBtn;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPanel jPanel3;
    private javax.swing.JPanel jPanel4;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JButton messageBtn;
    private javax.swing.JPanel panelBack;
    private javax.swing.JTextArea txtMessage;
    // End of variables declaration
}
```

12 - Supplier.java

```
package JFrames;

import com.mysql.cj.jdbc.Driver;
import java.awt.Color;
import java.awt.Font;
import javax.swing.table.DefaultTableModel;
import java.sql.PreparedStatement;
import java.sql.Connection;
import java.sql.ResultSet;
import java.util.logging.Level;
import java.util.logging.Logger;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.text.SimpleDateFormat;
import java.util.Calendar;

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */

/**
 *
 * @author joni
 */
public class Supplier extends javax.swing.JFrame {

    Color EnterBack = new Color(171,190,213);
    Color ExitBack = new Color(202,217,236);

    /**
     * Creates new form Supplier
     */
    public Supplier() {
        initComponents();
        this.setLocationRelativeTo(null);
        JTableSupplier.setRowHeight(25);
    }
    Connection con;
    PreparedStatement pst;
    ResultSet rs;
    DefaultTableModel model;
    double Total, Tax;
    int Subtotal;
```

```
    public void bill(){
    String Total = txtTotal.getText();
    String Pay = txtPay.getText();
    String Balance = txtBalance.getText();
    // Displays the exact date and time that the payment was made
    Calendar timer = Calendar.getInstance();
    timer.getTime();
    SimpleDateFormat Time = new SimpleDateFormat("HH:mm");
    Time.format(timer.getTime());
    SimpleDateFormat Date = new SimpleDateFormat("dd/MM/yyyy");
    // Appearance of the bill on the jTextarea
    DefaultTableModel model = new DefaultTableModel();
    model = (DefaultTableModel)JTableSupplier.getModel();
    txtBill.setText(txtBill.getText() + "*********************************************************\n");
    txtBill.setText(txtBill.getText() + "                              Amount due                  \n");
    txtBill.setText(txtBill.getText().formatted(Font.BOLD,12f));
    txtBill.setText(txtBill.getText() + "*********************************************************\n");
    txtBill.setText(txtBill.getText() + "Product" + "\t" + "\t" + "Price" + "\t" + "Amount" + "\n");
        // Loops through the ordered items to finally display the billing details (Product, Price and Amount)
        for (int i = 0; i < model.getRowCount(); i++){
            String ProductName = (String)model.getValueAt(i, 0);
            int Price = (Integer)model.getValueAt(i, 1);
            int Amount = (Integer)model.getValueAt(i, 3);
            txtBill.setText(txtBill.getText() + ProductName + "\t" + "\t" + Price  + "\t" + Amount + "\n");
        }
      txtBill.setText(txtBill.getText() + "\n");
      txtBill.setText(txtBill.getText() + "\t" + "\t" + "Taxed total: " + Total + "\n");
      txtBill.setText(txtBill.getText() + "\t" + "\t" + "Amount paid: " + Pay + "\n");
      txtBill.setText(txtBill.getText() + "\t" + "\t" + "Balance: " + Balance + "\n");
      txtBill.setText(txtBill.getText() + "\n");
      txtBill.setText(txtBill.getText() + "Date: " +Date.format(timer.getTime()) + "\t" + "                Time: "
+Time.format(timer.getTime())+ "\n");
        txtBill.setText(txtBill.getText() + "*********************************************************\n");
      txtBill.setText(txtBill.getText() + "            THANK YOU FOR CHOOSING SUPPLIER!             \n");
      txtBill.setText(txtBill.getText() + "\n");
      txtBill.setText(txtBill.getText() + "                                ERCA                  \n");
    }

    public void Return(){

    double Pay = Integer.parseInt(txtPay.getText());
    double Balance = Pay - Total;
    txtBalance.setText(String.valueOf(Balance));
    }

    private void bought(){
        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            con = DriverManager.getConnection("jdbc:mysql://localhost:3307/Restaurant", "root", "");
```

```
            String Name = "";
            int Qty = 0;

            String query3 = "update Product set Qty = Qty+? where Name = ?";
            pst = con.prepareStatement(query3);

            for (int i = 0; i < JTableSupplier.getRowCount(); i++) {

                Name = (String) JTableSupplier.getValueAt(i, 0);
                Qty = (int) JTableSupplier.getValueAt(i, 2);

                pst.setInt(1, Qty);
                pst.setString(2, Name);
                pst.execute();
            }

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

        String ProductName = (txtPname.getText());
        int Price = Integer.parseInt(txtPrice.getText());
        int Quantity = Integer.parseInt(txtQuantity.getText());
        //Perform the following calculations
        Subtotal = (Quantity * Price);
        Total = Total - Tax;
        Tax = Tax + Subtotal * 0.15; //15% is the taxrate.
        Total = Total + Subtotal + Tax;
        // Call the results on their respective fields
        txtSubtotal.setText(""+Subtotal);
        txtTax.setText(""+Tax);
        txtTotal.setText(""+Total);

        model = (DefaultTableModel)JTableSupplier.getModel();
        model.addRow(new Object[]{ //Adds the objects below in the order presented
            ProductName,
            Price,
            Quantity,
            Subtotal,
        });

        int sum = 0; // Initialize the total charge

        for(int i=0; i<JTableSupplier.getRowCount(); i++){ // Loop through the 4 columns in the jTable
            sum = sum + Integer.parseInt(JTableSupplier.getValueAt(i, 3).toString());
```

```
    }
    // Clear fields in case a new item was to be added for purchase
    txtSubtotal.setText(Integer.toString(sum));
    txtID.setText("");
    txtPname.setText("");
    txtPrice.setText("");
    txtQuantity.setText("");
    txtID.requestFocus();
}

private void printBtnActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    Return();
    bill();
    bought();

    DefaultTableModel model = (DefaultTableModel)JTableSupplier.getModel();
    model.setRowCount(0);

}

private void JTableSupplierMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
     DefaultTableModel d1 = (DefaultTableModel)JTableSupplier.getModel();
     int selectIndex = JTableSupplier.getSelectedRow();

     txtPname.setText(d1.getValueAt(selectIndex, 0).toString());
     txtPrice.setText(d1.getValueAt(selectIndex, 1).toString());
     txtQuantity.setText(d1.getValueAt(selectIndex, 2).toString());
     txtTotal.setText(d1.getValueAt(selectIndex, 3).toString());

}

private void txtQuantityKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    if(!Character.isDigit(evt.getKeyChar())){
        evt.consume();}
}

private void txtPayKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    if(!Character.isDigit(evt.getKeyChar())){
        evt.consume();}
}

private void txtIDKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    try {
        Class.forName("com.mysql.cj.jdbc.Driver");
```

```
            try {
                con = DriverManager.getConnection("jdbc:mysql://localhost:3307/Restaurant","root", "");
                String ID = txtID.getText();

                pst = con.prepareStatement("select ID, Barcode, ProductName, Price from Supplier where
barcode = ?");
                pst.setString(1, ID);
                ResultSet rs = pst.executeQuery();

                if(rs.next()==true){
                    String ProductName = rs.getString(3);
                    String Price = rs.getString(4);
                    txtPname.setText(ProductName);
                    txtPrice.setText(Price);

                }else{
                    txtPname.setText("");
                    txtPrice.setText("");
                }

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

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

    private void lblBackMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        AdminPage Admin = new AdminPage();
        Admin.setVisible(true);
        dispose();
    }

    private void lblBackMouseEntered(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        panelBack.setBackground(EnterBack);
    }

    private void lblBackMouseExited(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        panelBack.setBackground(ExitBack);
    }

    // Variables declaration - do not modify
    private javax.swing.JTable JTableSupplier;
    private javax.swing.JButton addBtn;
```

```
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel11;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPanel jPanel3;
    private javax.swing.JPanel jPanel4;
    private javax.swing.JPanel jPanel5;
    private javax.swing.JPanel jPanel6;
    private javax.swing.JPanel jPanel7;
    private javax.swing.JPanel jPanel8;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JScrollPane jScrollPane2;
    private javax.swing.JLabel lblBack;
    private javax.swing.JPanel panelBack;
    private javax.swing.JButton printBtn;
    private javax.swing.JTextField txtBalance;
    private javax.swing.JTextArea txtBill;
    private javax.swing.JTextField txtID;
    private javax.swing.JTextField txtPay;
    private javax.swing.JTextField txtPname;
    private javax.swing.JTextField txtPrice;
    private javax.swing.JTextField txtQuantity;
    private javax.swing.JTextField txtSubtotal;
    private javax.swing.JTextField txtTax;
    private javax.swing.JTextField txtTotal;
    // End of variables declaration
}
```

13 - Categorie

```
package Java;

public class Categorie {
    private int ID;
    private String Name;

    public Categorie(int pID, String pName){
      this.ID = pID;
```

```
        this.Name = pName;
    }
     public int getID() {
         return ID;
       }
       public void setID(int ID) {
          this.ID = ID;
       }
       public String getName() {
          return Name;
       }
       public void setName(String Name) {
          this.Name = Name;
       }

   Categorie(int aInt) {
       throw new UnsupportedOperationException("Not supported yet."); // Generated from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
   }
```

14 - Employee.java

```
package Java;

public class Employee {

   private int WorkerID;
   private String Name;
   private String Email;
   private String DOB;
   private int PhoneNumber;
   private String Gender;
   private String Position;
   private float Salary;
   private String Address;
   private byte[] Photo;

   //We will be making an object(an Employee) out of this constructor.
   public Employee(int EWorkerID, String EName, String EEmail, String EDOB,
                int EPhoneNumber, String EGender, String EPosition,
                float ESalary, String EAddress, byte[] EPhoto)
   {
      this.WorkerID = EWorkerID;
      this.Name = EName;
      this.Email = EEmail;
      this.DOB = EDOB;
      this.PhoneNumber = EPhoneNumber;
```

```
        this.Gender = EGender;
        this.Position = EPosition;
        this.Salary = ESalary;
        this.Address = EAddress;
        this.Photo = EPhoto;
    }

    Employee(int aInt, String string, String string0, String string1, int aInt0, String string2, float aFloat,
String string3, byte[] bytes) {
        throw new UnsupportedOperationException("Not supported yet."); // Generated from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
    }
    Employee() {
        throw new UnsupportedOperationException("Not supported yet."); // Generated from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
    }

    //Autogenerated setters and getters
    //by encapsulating the private variables.

    public int getWorkerID(){
        return WorkerID;
    }
    public String getName(){
        return Name;
    }
    public String getEmail(){
        return Email;
    }
    public String getDOB(){
        return DOB;
    }
    public int getPhoneNumber(){
        return PhoneNumber;
    }
    public String getGender(){
        return Gender;
    }
    public String getPosition(){
        return Position;
    }
    public float getSalary(){
        return Salary;
    }
    public String getAddress(){
        return Address;
    }
    public byte[] getPhoto(){
        return Photo;
```

```
}

/**
 * @param WorkerID the WorkerID to set
 */
public void setWorkerID(int WorkerID) {
    this.WorkerID = WorkerID;
}

/**
 * @param Name the Name to set
 */
public void setName(String Name) {
    this.Name = Name;
}

/**
 * @param Email the Email to set
 */
public void setEmail(String Email) {
    this.Email = Email;
}

/**
 * @param DOB the DOB to set
 */
public void setDOB(String DOB) {
    this.DOB = DOB;
}

/**
 * @param PhoneNumber the PhoneNumber to set
 */
public void setPhoneNumber(int PhoneNumber) {
    this.PhoneNumber = PhoneNumber;
}

/**
 * @param Gender the Gender to set
 */
public void setGender(String Gender) {
    this.Gender = Gender;
}

/**
 * @param Position the Position to set
 */
public void setPosition(String Position) {
    this.Position = Position;
```

```
    }

    /**
     * @param Salary the Salary to set
     */
    public void setSalary(float Salary) {
        this.Salary = Salary;
    }

    /**
     * @param Address the Address to set
     */
    public void setAddress(String Address) {
        this.Address = Address;
    }

    /**
     * @param Photo the Photo to set
     */
    public void setPhoto(byte[] Photo) {
        this.Photo = Photo;
    }

}
```

15 - MyQuery.java

```
package Java;

import java.sql.Statement;
import java.sql.ResultSet;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.HashMap;
import java.util.logging.Level;
import java.util.logging.Logger;


public class MyQuery {

    public Connection getConnection() {
        Connection con = null;
        try {
            con = DriverManager.getConnection("jdbc:mysql://localhost:3307/Restaurant", "root", "");
        } catch (SQLException ex) {
            Logger.getLogger(Categorie.class.getName()).log(Level.SEVERE, null, ex);
```

```
        }
            return con;

    }

    //Using this class to populate the name of the category into the jcombobox.
    public HashMap<String, Integer> comboBox(){ //Create the HashMap and import it.
          HashMap<String, Integer> map = new HashMap<String, Integer>();
        Connection con = getConnection(); //Connect to the MySQL database.
        Statement st = null;
        ResultSet rs = null;
        String query = "SELECT * FROM `Category`"; //Database table name.

        try {
            st = con.createStatement();
            rs = st.executeQuery(query); //Call the query
            Categorie cat;

            while (rs.next()){
                cat = new Categorie(rs.getInt(1), rs.getString(2));
                map.put(cat.getName(), cat.getID()); //The map method takes the two parameters
                                                     //from our 'Categorie' class initialization.
            }
        } catch (SQLException ex) { //autogenerated sql catch clause
            Logger.getLogger(MyQuery.class.getName()).log(Level.SEVERE, null, ex);
        }
        return map;
    }
}
```