

PREFACE

I am Pleased to Present this Report on the Application, named “**Fusion Computer Warehouse**” Developed at the Computer Department – Bhakta Kavi Narsinh Mehta University as a part of our Fifth Semester Curriculums.

The Application Report is Submitted as a Part of The Course Leading to Bachelor of Computer Application Dr. V. R Godhaniya Collage of I.T, Porbandar.

This Report Contains overview of the whole Application anyone who is technical person can understand the contents of this report very easily. The implementation of the system is shown with figures and screens readers can find it easy to understand.

Acknowledgement

The successful and final outcome of this Application required lots of guidance and assistance from many people and I am extremely fortunate to have completed my Application work. Whatever I have done is only due to such guidance and assistance and I should not forget to thank them. I respect and thanks Mr. Savan Raval at first place for giving me an opportunity to do this Application and providing me support and guidance through which I could complete the Application in time. I am extremely grateful to his for providing such co-operation.

“Fusion Computer Warehouse” and which we conceived in the 5th semester of BCA affiliated with BKNMU (Bhakta Kavi Narsinh Mehta University).

We are also thankful to the BKNMU (Bhakta Kavi Narsinh Mehta University) for including this Application development subject in our syllabus. We got a golden opportunity to test and implement our creativity simultaneously. Lastly, we would like to extend our sincere thanks to our advisors, classmates as well as all the books and websites.

Index

Sr. No.	Title	Page no.
1	Preface	1
2	Acknowledgement	2
3	Project Profile	4
4	Abstract of Project	5
5	Feasibility Study(preliminary investigation)	6
6	Requirement analysis	7
7	Project Life Cycle(E.R. Diagram, Usecase Diagram)	8
8	Data Dictionary	11
9	Software Testing	15
10	User Interface Designs(Snapshots) & Source Code	16
11	Appendices (References, Bibliography)	85

Project Profile

Project Title : - Fusion Computer Warehouse

Front End : - Java

Back End : - MySQL

College Name : - Dr. V.R. Godhaniya IT College

Project Guide :- Mr. Savan Raval

Project By :- Nadeemahmed Palkhiwala

Abstract of Project

- The title of the project is “Fusion Computer Warehouse”. This project is aimed to fulfill all the Computer product requirements.
- A large variety of PC related devices such as Monitors, Motherboards and other Input / Output devices.

Feasibility Study

- A feasibility study is a short, focused study, which aim to answer a number of questions:

Does the system contribute to the overall objectives of the organization.

Different type of feasibility study is as follow:

- 1) Operational Feasibility
- 2) Technical Feasibility
- 3) Economical Feasibility

Operational Feasibility :

- The operational feasibility is a measure of how well a proposed system solves the problems.
- And takes advantages of the opportunities identification during scope definition and how it satisfy the requirement.
- Identification in the requirement analysis phase of system development.

Technical Feasibility :

- The technical feasibility assessment is focused on gaining and understanding of present technical resources of the organization and their applicability to the expected needs of the system.
- It is an evaluation of the hardware and software and how it meets the need of the proposed system.

Economical Feasibility :

- The purpose of economical feasibility assessment is to determine the positive economic benefits to the organization that proposed system will provide.
 - It includes qualification and identification of all benefits expected.
 - The assessment typically involves a cost/ benefits analysis.

Requirement Analysis

○ Minimum Hardware Requirement

<u>Component</u>	<u>Minimum Requirement</u>
Processor	: - i3 Processor or Above
Supported Architecture	: - 32 Bit : - 64 Bit
Memory	: - 2 GB or Above
Hard Disk	: - 250 GB Minimum
Ram	: - 2 Gb or Above

○ Minimum Software Requirement

<u>Component</u>	<u>Minimum Requirement</u>
Operating system	: -Windows : -7 or above

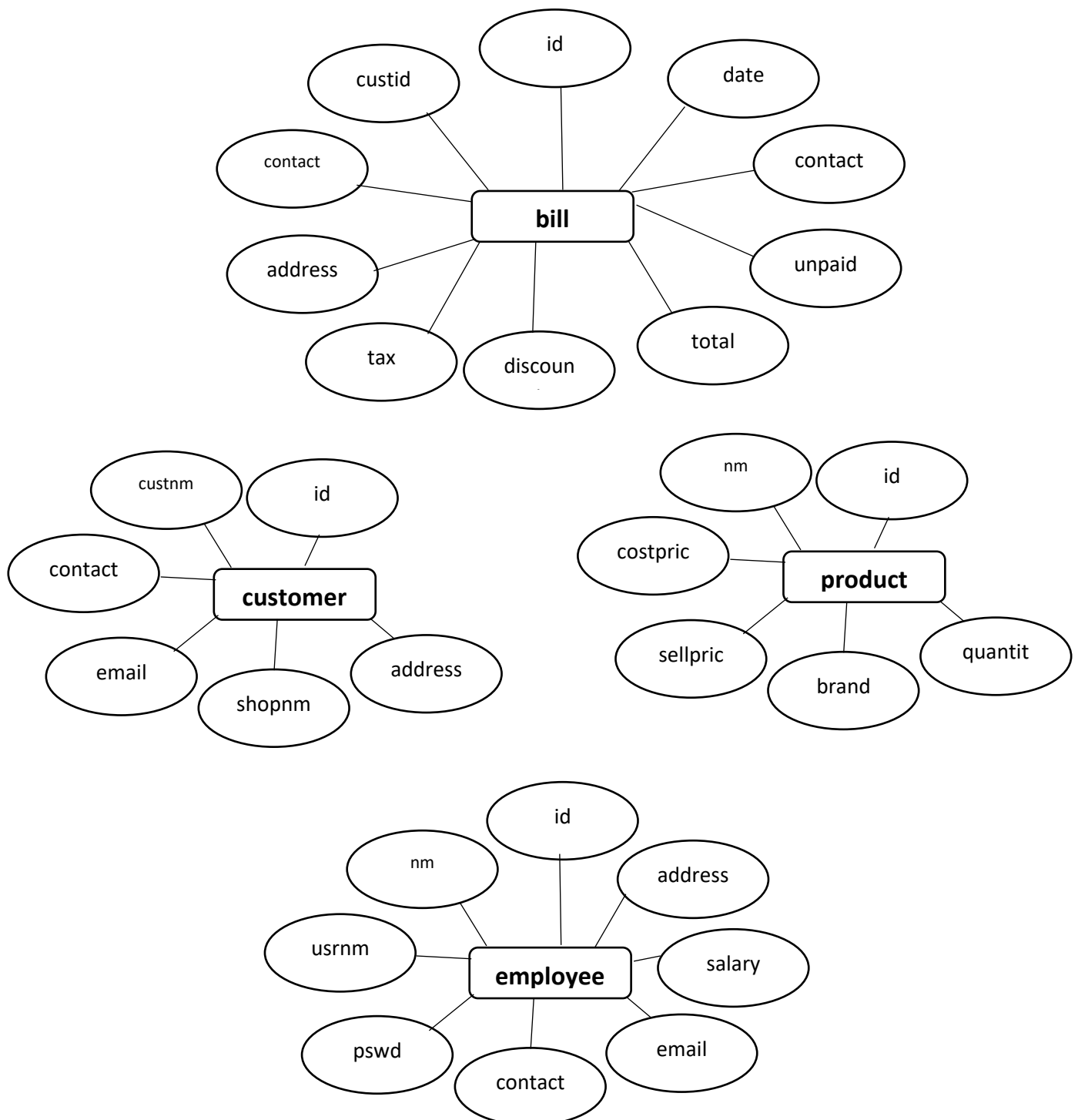
WAMP

: - Php 5.6, MYSQL 10.4

Project Develop Software

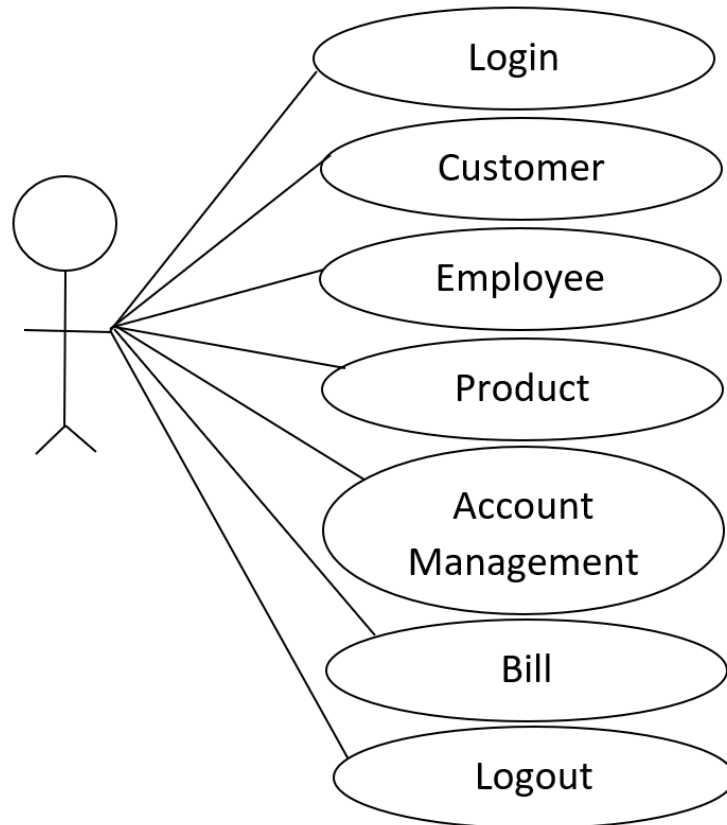
: - Netbeans IDE

ER DIAGRAM

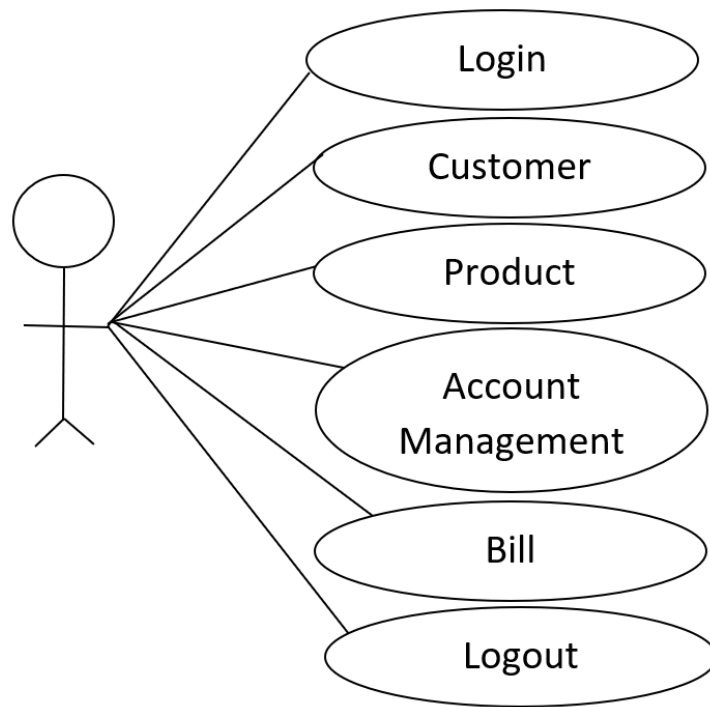


Use Case Diagram

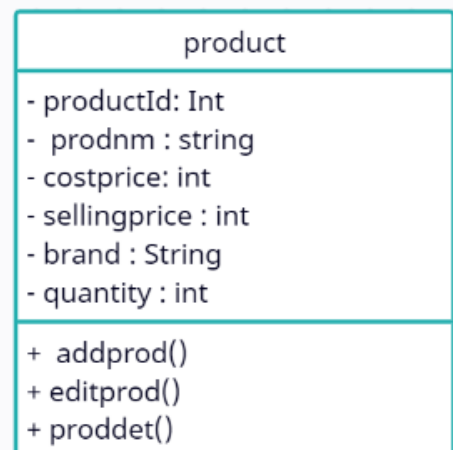
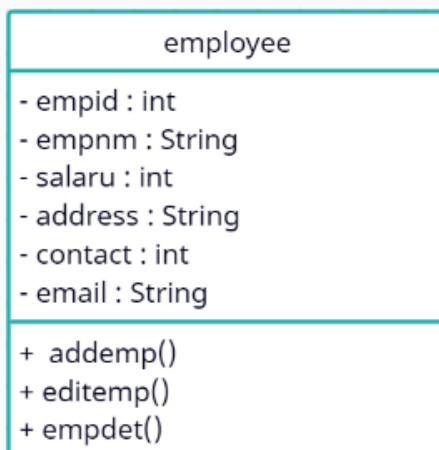
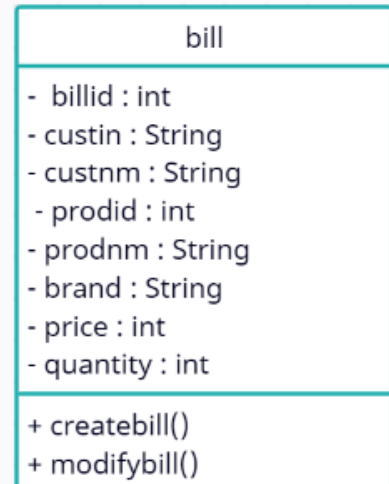
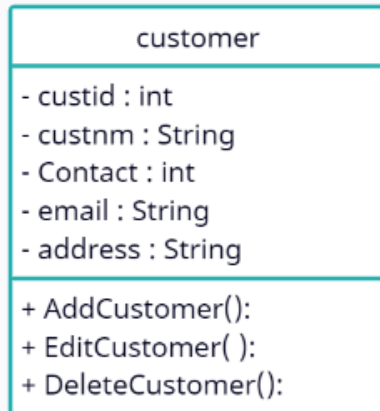
Admin



Employee

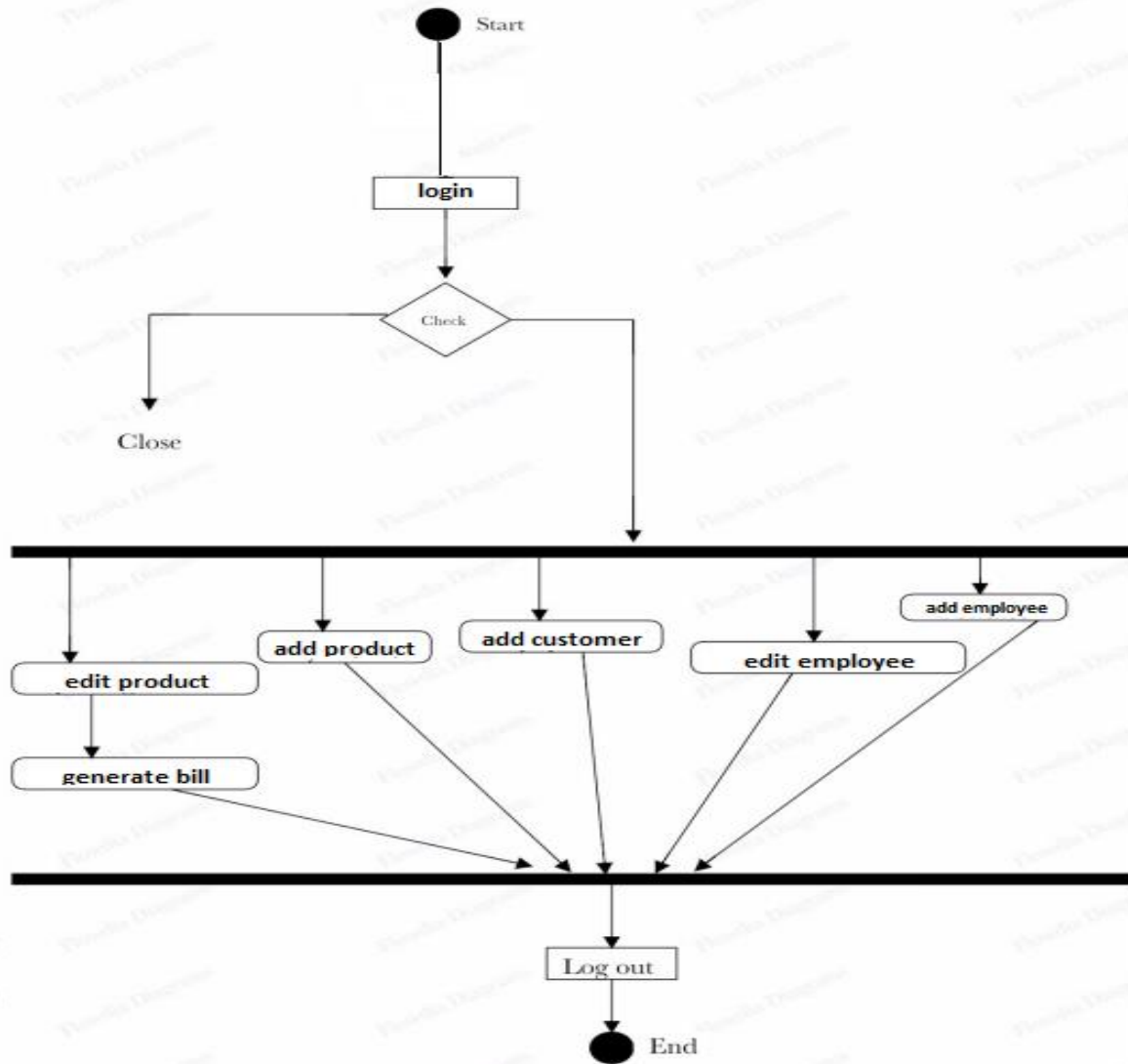


CLASS DIAGRAM



DATA FLOW DIAGRAM

ACTIVITY DIAGRAM



Data Dictionary

TABLES:

1. customer :

Sr no.	Filed Name	Data type	Size
1	id	varchar	5
2	custnm	varchar	30
3	contact	varchar	10
4	email	varchar	50
5	shopnm	varchar	30
6	address	varchar	50

2. employee :

Sr. No	Field Name	Data Type	Size
1	id	varchar	5
2	nm	varchar	25
3	usrnm	varchar	30
4	pswd	varchar	20
5	contact	varchar	30
6	email	varchar	10
7	salary	varchar	30
8	address	varchar	30

3. product :

Sr. No	Field Name	Data Type	Size
1	id	varchar	5
2	nm	varchar	20
3	costprice	varchar	20
4	sellingprice	varchar	20
5	brand	varchar	20
6	quantity	varchar	20

4. bill :

Sr. No	Field Name	Data Type	Size
1	id	varchar	5
2	custid	varchar	30
3	custnm	Varchar	25
4	contact	Varchar	10
5	address	Varchar	50
6	tax	Varchar	10
7	discount	Varchar	10
8	tax	Varchar	10
9	discount	Varchar	10
10	total	Varchar	10
11	unpaid	Varchar	10
12	date	Varchar	10

Test cases Preparation

•Purpose:-

- The purpose of the test cases is to test the various input and see the output produce any error or not. A correct system must accomplish the following:
- Computer correct results. Operate safely, and cause the server to operate safely.
- Perform the tasks required by the system containing the software, as explained in the software applications.
- Achieve these goals for all inputs.
- Recognize inputs outside its domain.
- We shall see that satisfying these pre-requisites depends on a variety of things. One of these things is to provide clear and correct software specifications.
- Required Input and expected result.

Test data Preparation

- When an extensive range or volume of data is needed for testing then using the test data preparation tool is of great help.
- They are very useful for performance and reliability testing, where a large amount of realistic data is needed. They may be used by developers and may also be used during system or acceptance testing.
- Test data preparation tools allow (enable) data to be selected from an existing database or created, generated, manipulated and edited for use in tests. The most sophisticated tools can deal with a range of files and database formats.
- Features or characteristics of test data preparation tools are as follows:
- To extract selected data records from files or databases.
- To 'massage' data records to make them anonymous or not able to be identified with real people (for data protection);
- To enable records to be sorted or arranged in a different order;
- To generate new records populated with pseudo-random data, or data set up according to some guidelines, e.g. an operational profile;

- To construct a large number of similar records from a template, for example to give a large set of records for volume tests.

Software Testing

- **Manual (Issues & Solutions) with data**

Manual testing is a method used by software developers to run tests manually. There are many manual testing types which are carried out manually as well as automatically. These are,

- **Black Box Testing:** It is a testing method to test functionalities and requirements of the system. It does not test the internal part of the system.
- **White Box Testing:** It is a testing method based on information of the internal logic of an application's code and also known as Glass box Testing. It works on Internal working code of the system.
- **Integration Testing:** Integrated modules testing method to verify joint functionality after integration. Modules are typically code modules, individual applications, client and server applications on a network, etc. This type of testing is especially applicable to client/server and distributed systems.
- **System Testing:** It is a technique to test whole system.
- **Unit Testing:** Testing method to test specific component of software or module. It is specially done by programmers and not by testers, because it needs thorough knowledge of the internal programming design and code.
- **Acceptance Testing:** This type of testing verifies that the system meets the customer specified requirements or not. User or a customer does this testing to decide whether to accept application.
- **Automated (Issues & Solutions)** ○ The method takes automation tool support to execute the test cases is known as Automation Testing. Automated testing is good for large project. It is a method which uses automation tools to run tests that repeat predefined actions, matches the developed program's probable and real results. If the project prospects and results align, your project is behaving as it should, and you are likely bug free. If the two don't align, still, there is a problem that requires to be addressed. You'll have to take a look at your code, alter it, and continue to run tests until the actual and expected outcomes align.

1. Splash Screen :



```
import java.awt.geom.RoundRectangle2D;
import javax.swing.JOptionPane;

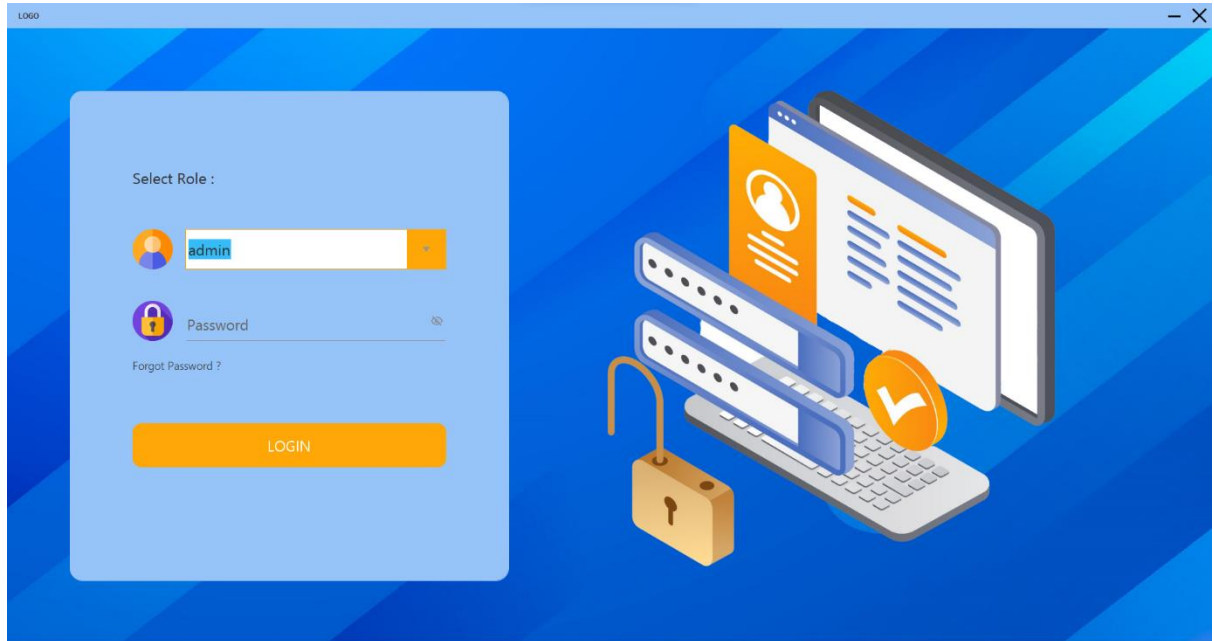
public class Splash extends javax.swing.JFrame
{
    public Splash()
    {
        initComponents();
        setShape(new RoundRectangle2D.Double(0, 0, getWidth(), getHeight(), 40, 40))
    }
    {
        Splash s = new Splash();
```

```
java.awt.EventQueue.invokeLater(new Runnable()
{
    public void run()
    {
        s.setVisible(true);
    }
});
s.splashh();
}
```

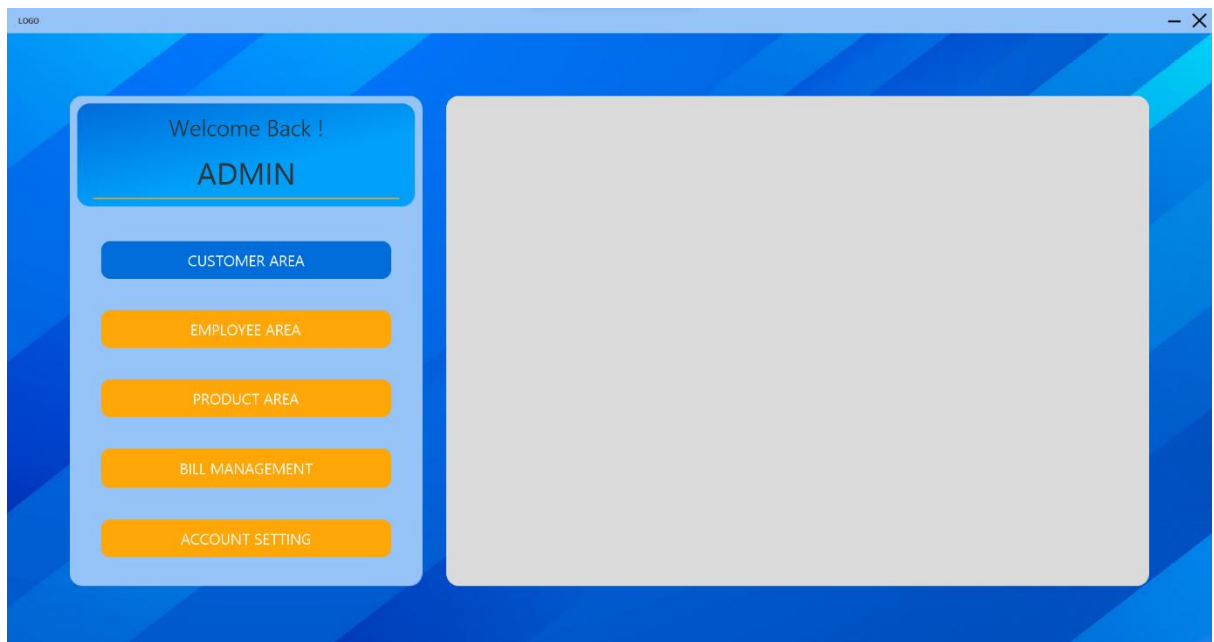
```
public void splashh()
{
    try
    {
        Thread.sleep(1000);
        for (int i = 0; i <= 4; i++)
        {
            if (i == 0)
            {
                lbl_txt.setText("Connecting to Database...");
                Thread.sleep(1500);
            }
            else if (i == 1)
            {
                lbl_txt.setText("Importing Files...");
                Thread.sleep(1500);
            }
            else if (i == 2)
            {
                lbl_txt.setText("Almost there...");
                Thread.sleep(1500);
            }
        }
    }
}
```

```
        else if (i == 3)
        {
            lbl_txt.setText("Opening Application..");
            Thread.sleep(1500);
        }
        else
        {
            Main m = new Main();
            m.setVisible(true);
            this.dispose();
        }
    }
}
catch (Exception e)
{
    JOptionPane.showMessageDialog(rootPane, e.getMessage());
}
}
```

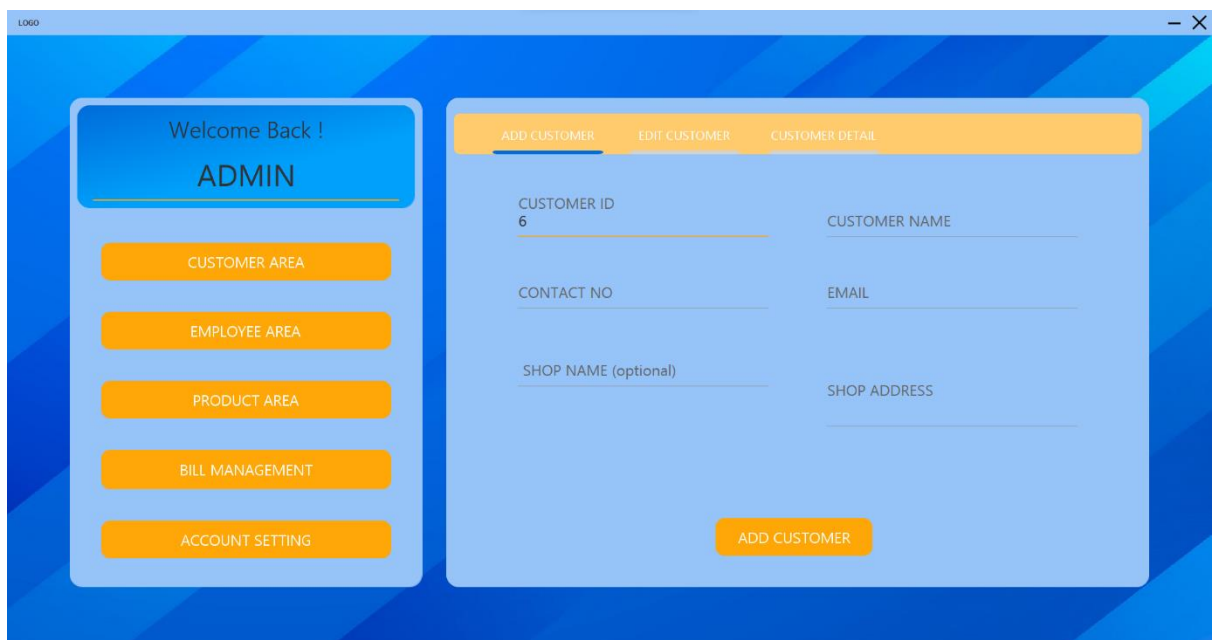
2. Login Screen :



3. Dashboard



4. Add Customer



5. Edit Customer

Welcome Back !
ADMIN

CUSTOMER AREA
EMPLOYEE AREA
PRODUCT AREA
BILL MANAGEMENT
ACCOUNT SETTING

ADD CUSTOMER EDIT CUSTOMER CUSTOMER DETAIL

ID	NAME	CONTACT	EMAIL	SHOP NAME	ADDRESS
1	Nadeem Palkhiwala	9067101621	nadeem@gmail.com	fusion	pbr
2	Usha	9432890323	ujfas@infod.com	ujfa	ujfa
3	Hkaidi	hkasjda	hkasd@hkasd.com	hkasd	hkasd
4	Aman	8934284370	dsujs@gs.com	jonesdf	jmmmsda

CUSTOMER ID
1

CUSTOMER NAME
Nadeem Palkhiwala

CONTACT NO
9067101621

EMAIL
nadeem@gmail.com

SHOP NAME
fusion

SHOP ADDRESS
pbr

EDIT REMOVE

6. Customer Detail

Welcome Back !
ADMIN

CUSTOMER AREA
EMPLOYEE AREA
PRODUCT AREA
BILL MANAGEMENT
ACCOUNT SETTING

ADD CUSTOMER EDIT CUSTOMER CUSTOMER DETAIL

SEARCH BY CUSTOMER NAME

ID	NAME	CONTACT	EMAIL	SHOP NAME	SHOP ADDRESS
1	Nadeem Palkhiwala	9067101621	nadeem@gmail.com	fusion	pbr
2	Usha	9432890323	ujfas@infod.com	ujfa	ujfa
3	Hkaidi	hkasjda	hkasd@hkasd.com	hkasd	hkasd
4	Aman	8934284370	dsujs@gs.com	jonesdf	jmmmsda

7. Add Employee

The screenshot displays a web application interface for adding a new employee. On the left, a sidebar menu for an 'ADMIN' user includes options like 'CUSTOMER AREA', 'EMPLOYEE AREA', 'PRODUCT AREA', 'BILL MANAGEMENT', and 'ACCOUNT SETTING'. The main content area features a tabbed interface with 'ADD EMPLOYEE', 'EDIT EMPLOYEE', and 'EMPLOYEE DETAIL' tabs. The 'ADD EMPLOYEE' tab is active, showing a form with fields for 'EMPLOYEE ID' (containing the number 4), 'EMPLOYEE NAME', 'USERNAME', 'PASSWORD' (with a toggle icon), 'CONTACT NO', 'EMAIL', 'SALARY', and 'ADDRESS'. An 'ADD EMPLOYEE' button is positioned at the bottom right of the form.

LOGO

Welcome Back !
ADMIN

CUSTOMER AREA

EMPLOYEE AREA

PRODUCT AREA

BILL MANAGEMENT

ACCOUNT SETTING

ADD EMPLOYEE

EDIT EMPLOYEE

EMPLOYEE DETAIL

EMPLOYEE ID
4

EMPLOYEE NAME

USERNAME

PASSWORD

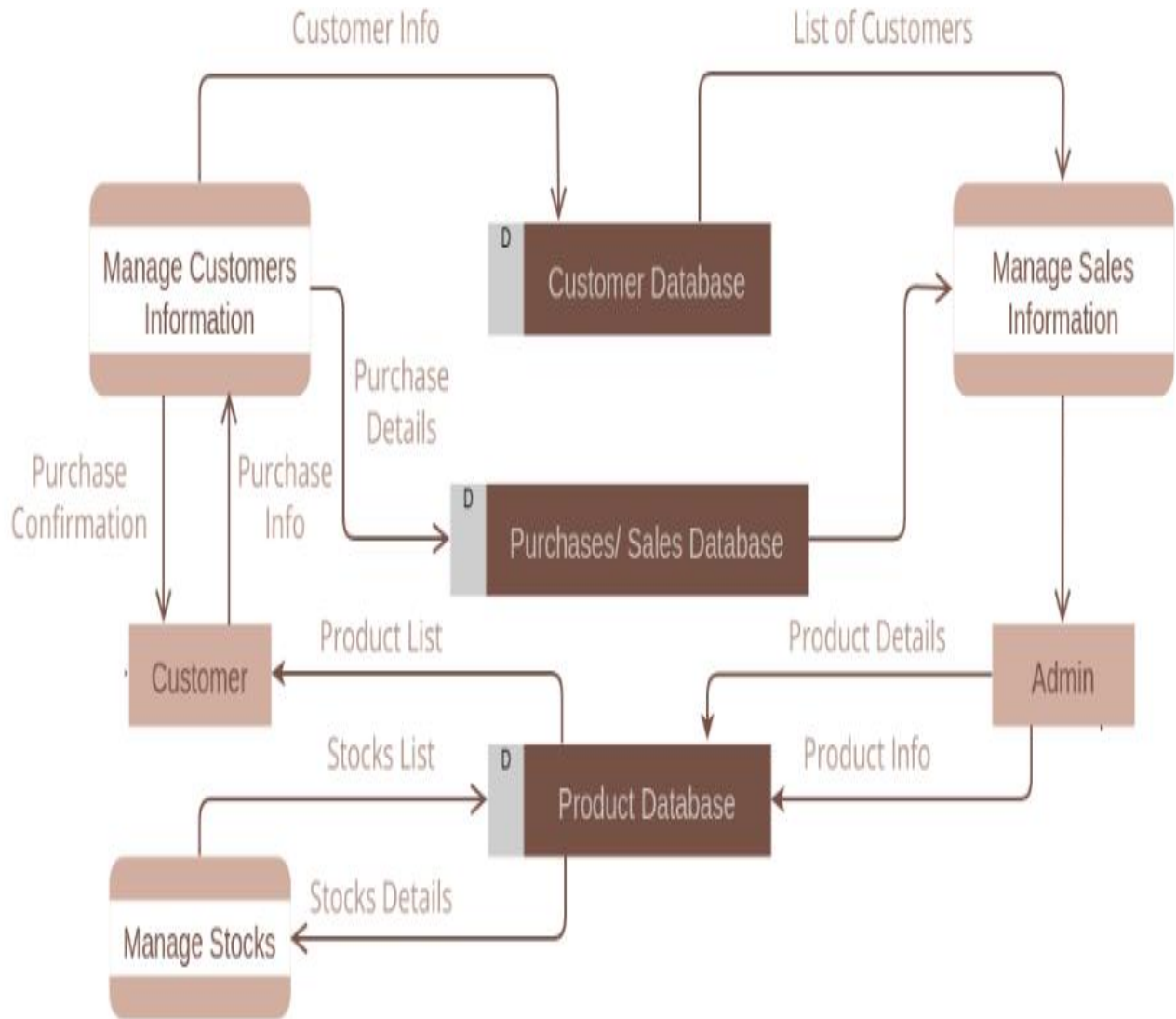
CONTACT NO

EMAIL

SALARY

ADDRESS

ADD EMPLOYEE



8. Edit Employee

9. Employee Detail



Logo

Welcome Back !
ADMIN

CUSTOMER AREA

EMPLOYEE AREA

PRODUCT AREA

BILL MANAGEMENT

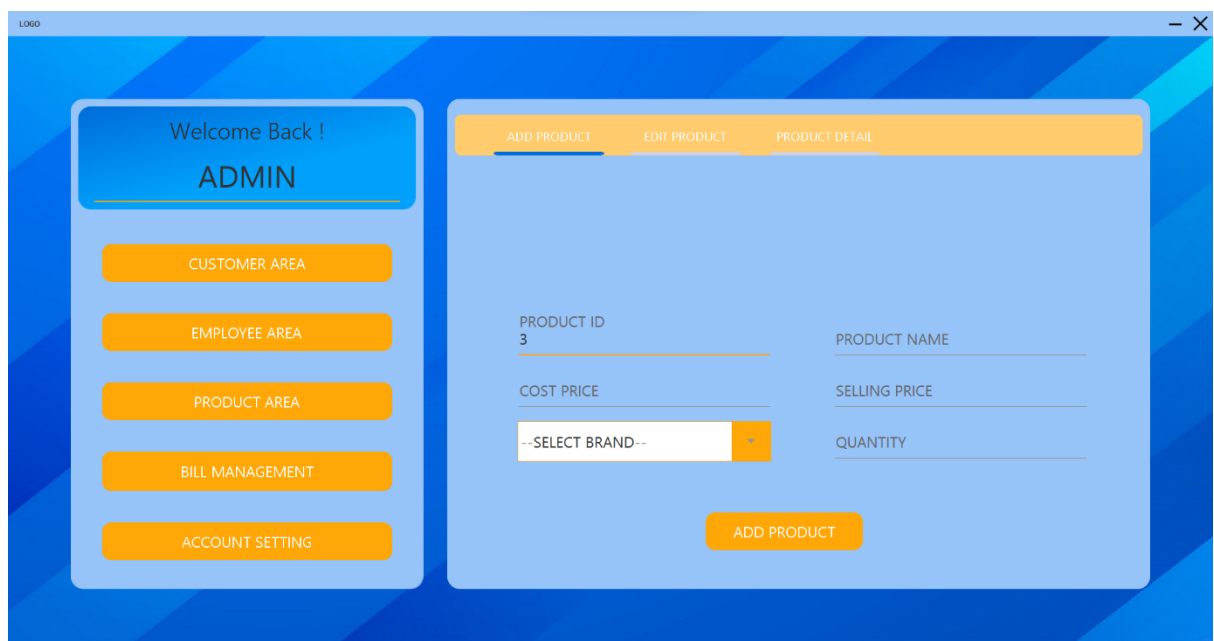
ACCOUNT SETTING

ADD EMPLOYEE EDIT EMPLOYEE **EMPLOYEE DETAIL**

SEARCH BY EMPLOYEE NAME

ID	NAME	CONTACT	EMAIL	SALARY	ADDRESS	PASSWORD
1	ADMIN	admin	admin123	9067101621	nadeempalshwala	120000
2	Dhruv Datta	dhravdatta	dhrav123	9408084884	dhrav@gmail.com	120000
3	Nadeem Palshwala	nadeem8	nadeem123	9067101621	nadeem@gmail.com	80000

10. Add Product



Logo

Welcome Back !
ADMIN

CUSTOMER AREA

EMPLOYEE AREA

PRODUCT AREA

BILL MANAGEMENT

ACCOUNT SETTING

ADD PRODUCT EDIT PRODUCT **PRODUCT DETAIL**

PRODUCT ID
3

PRODUCT NAME

COST PRICE

SELLING PRICE

--SELECT BRAND--

QUANTITY

ADD PRODUCT

11. Edit Product

The screenshot shows the 'Edit Product' interface. On the left, a sidebar for the 'ADMIN' user contains buttons for 'CUSTOMER AREA', 'EMPLOYEE AREA', 'PRODUCT AREA', 'BILL MANAGEMENT', and 'ACCOUNT SETTING'. The main area has a tabbed interface with 'ADD PRODUCT', 'EDIT PRODUCT' (active), and 'PRODUCT DETAIL'. Below the tabs is a table with columns: ID, NAME, COST PRICE, SELLING PRICE, BRAND, and QUANTITY. The table contains two rows: 1. Motherboard (Cost: 600, Selling: 783, Brand: --SELECT BRAND--, Qty: 3) and 2. Fan (Cost: 120, Selling: 150, Brand: ACER, Qty: 89). Below the table, the 'EDIT PRODUCT' form is displayed with fields for: PRODUCT ID (1), PRODUCT NAME (Motherboard), SELLING PRICE (783), COST PRICE (600), and QUANTITY (3). At the bottom are 'EDIT PRODUCT' and 'REMOVE PRODUCT' buttons.

ID	NAME	COST PRICE	SELLING PRICE	BRAND	QUANTITY
1	Motherboard	600	783	--SELECT BRAND--	3
2	Fan	120	150	ACER	89

PRODUCT ID: 1

PRODUCT NAME: Motherboard

SELLING PRICE: 783

COST PRICE: 600

QUANTITY: 3

Buttons: EDIT PRODUCT, REMOVE PRODUCT

12. Product Detail

The screenshot shows the 'Product Detail' interface. The sidebar and tabs are identical to the previous screen. The main area features a 'SEARCH BY PRODUCT NAME' input field above a table with the same columns and data as in the 'Edit Product' screen. Below the table, there are no additional form fields or buttons.

ID	NAME	COST PRICE	SELLING PRICE	BRAND	QUANTITY
1	Motherboard	600	783	--SELECT BRAND--	3
2	Fan	120	150	ACER	89

SEARCH BY PRODUCT NAME

13. Create Bill

The screenshot shows the 'Create Bill' interface. On the left is a sidebar with a 'Welcome Back! ADMIN' header and five menu items: 'CUSTOMER AREA', 'EMPLOYEE AREA', 'PRODUCT AREA', 'BILL MANAGEMENT', and 'ACCOUNT SETTING'. The main content area has a top bar with 'CREATE BILL' and 'MODIFY BILL' tabs. Below this, the 'BILL NO.' is 1. The 'CONSUMER DETAIL' section includes a dropdown for 'Datta', 'CUST ID' 2, 'CONTACT' 9432890323, and 'ADDRESS' klds. The 'PRODUCTS LISTING' section has a dropdown for 'Motherboard', 'PRODUCT ID' 1, 'PRICE' 783, 'QUANTITY' 1, 'DISCOUNT(%)' 0, and 'TAX(%)' 0. Below this is a table with columns: Costname, Contact, Product No., Price, Quantity, and Total. To the right of the table is the 'UNPAID AMOUNT' 0 and an 'ADD' button. At the bottom right are buttons for 'SAVE AS PDF' and a summary section with 'GROSS TOTAL : 0', 'DISCOUNT : 0', 'TAX : 0', 'UNPAID : 0', and 'NET TOTAL : 0'.

14. Modify Bill

The screenshot shows the 'Modify Bill' interface. The sidebar is identical to the 'Create Bill' screen. The main content area has a top bar with 'CREATE BILL' and 'MODIFY BILL' tabs, with 'MODIFY BILL' being the active tab. Below this is a 'BILL ID' dropdown. The 'CONSUMER DETAIL' section includes a dropdown for 'CUSTOMER NAME', 'CUST ID', 'CONTACT', and 'ADDRESS'. The 'PRODUCTS LISTING' section has a dropdown for 'PRODUCT NAME', 'PRODUCT ID', 'PRICE', 'QUANTITY' (set to 0), 'DISCOUNT(%)', and 'TAX(%)'. The bottom right section is empty, likely for updating totals.

15. Account Settings

LOGO

Welcome Back !
ADMIN

CUSTOMER AREA

EMPLOYEE AREA

PRODUCT AREA

BILL MANAGEMENT

ACCOUNT SETTING

SETTING TAB 2 TAB 3

1

EMPLOYEE NAME
ADMIN

USERNAME
admin

PASSWORD

CONTACT NO
9067101621

EMAIL
nadeempalkhiwala@gmail.com

SALARY
120000

ADDRESS
Porbandar

UPDATE

```
import com.k33ptoo.components.KButton;
import com.k33ptoo.components.KGradientPanel;
import combo_suggestion.ComboBoxSuggestion;
import connection.dbconnection;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Graphics;
import java.awt.Image;
import java.awt.Insets;
import java.awt.Toolkit;
import java.awt.event.KeyEvent;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.ImageIcon;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;
import textfield.PasswordField;
import textfield.TextField;

public class Main extends javax.swing.JFrame
{

    int total = 0;
    Connection con;
    Statement st;
    PreparedStatement pst;
    ResultSet result;
    DefaultTableModel tableModel;
    String query;

    public Main()
    {
        initComponents();

        Dimension size = Toolkit.getDefaultToolkit().getScreenSize();
        Insets scnmax = Toolkit.getDefaultToolkit().getScreenInsets(getGraphicsConfiguration());
        int taskbar = scnmax.bottom;

        int width = (int) size.getWidth();
        int height = (int) size.getHeight() - taskbar;

        this.setSize(width, height);
```

```
panel_bg.setSize(width, height);

System.out.println("Frame Size = " + this.getSize());
System.out.println("Panel BG = " + panel_bg.getSize());

cmbbox_login_usrnm.grabFocus();

panel_dashboard.setVisible(false);

con = dbconnection.getdbConnection();

fillCombobox("employee", "usrnm", cmbbox_login_usrnm);
}
private void lbl_closeMouseEntered(java.awt.event.MouseEvent evt)
{
    hover(pnl_close, Color.red);
}

private void lbl_closeMouseExited(java.awt.event.MouseEvent evt)
{
    hover(pnl_close, new Color(150, 195, 248));
}

private void lbl_minimizeMouseEntered(java.awt.event.MouseEvent evt)
{
    hover(pnl_minimize, new Color(153, 153, 153));
}

private void lbl_minimizeMouseExited(java.awt.event.MouseEvent evt)
{
    hover(pnl_minimize, new Color(150, 195, 248));
}

private void lbl_closeMouseClicked(java.awt.event.MouseEvent evt)
{
    System.exit(0);
}

private void lbl_minimizeMouseClicked(java.awt.event.MouseEvent evt)
{
    this.setExtendedState(NewJFrame.ICONIFIED);
}

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

private void lbl_forgot_pswdMouseEntered(java.awt.event.MouseEvent evt)
```

```
{
    lbl_forgot_pswd.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
}

private void btnCustomerMouseEntered(java.awt.event.MouseEvent evt)
{
    btnCustomer.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
}

private void btn_loginActionPerformed(java.awt.event.ActionEvent evt)
{
    panel_login.setVisible(false);
    panel_dashboard.setVisible(true);
    visibility(true, false, false, false, false, false, btnCustomer);

    try
    {
        con = dbconnection.getdbConnection();
        query = "select nm from employee where usnm like '" +
cmbbox_login_usnm.getSelectedItem().toString() + "'";
        st = con.createStatement();
        result = st.executeQuery(query);
        result.next();
        lbl_dash_usr.setText(result.getString("nm"));

        if (lbl_dash_usr.getText().equals("ADMIN"))
        {
            btnEmployee.setEnabled(true);
        }
        else
        {
            btnEmployee.setEnabled(false);
        }
    }
    catch (Exception e)
    {
        JOptionPane.showMessageDialog(null, e.getMessage());
    }
}

private void btnEmployeeMouseEntered(java.awt.event.MouseEvent evt)
{
    btnEmployee.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
}

private void btnProductMouseEntered(java.awt.event.MouseEvent evt)
{
    btnProduct.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
}
```

```
private void btnBillMouseEntered(java.awt.event.MouseEvent evt)
{
    btnBill.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
}

private void btnaccountMouseEntered(java.awt.event.MouseEvent evt)
{
    btnaccount.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
}

private void btn_loginFocusGained(java.awt.event.FocusEvent evt)
{
    hover(btn_login, new Color(1, 109, 218), Color.black);
}

private void btn_loginFocusLost(java.awt.event.FocusEvent evt)
{
    hover(btn_login, new Color(255, 167, 6), Color.white);
}

private void btnCustomerActionPerformed(java.awt.event.ActionEvent evt)
{
    visibility(false, true, false, false, false, false, btn_addcust_menu);
    visibility(pnl_add_cust, pnl_edit_cust, pnl_cust_det);
    onIndicator(lid_add_cust, lid_edit_cust, lid_cust_det);
    txt_addcust_id.setText(getMaxID("customer"));

    txt_addcust_id.grabFocus();
}

private void btn_loginMouseEntered(java.awt.event.MouseEvent evt)
{
    btn_login.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
}

private void btn_loginKeyPressed(java.awt.event.KeyEvent evt)
{
    int code = evt.getKeyCode();
    if (code == KeyEvent.VK_ENTER)
    {
        panel_login.setVisible(false);
        panel_dashboard.setVisible(true);
        lbl_dash_usr.setText(cmbbox_login_usrnm.getSelectedItem().toString());
        lbl_dash_welcome.grabFocus();
        visibility(true, false, false, false, false, false, btnCustomer);
    }
}
```



```
private void btnCustomerFocusGained(java.awt.event.FocusEvent evt)
{
    hover(btnCustomer, new Color(1, 109, 218), Color.black);
}
```

```
private void btnCustomerFocusLost(java.awt.event.FocusEvent evt)
{
    hover(btnCustomer, new Color(255, 167, 6), Color.white);
}
```

```
private void btnEmployeeFocusGained(java.awt.event.FocusEvent evt)
{
    hover(btnEmployee, new Color(1, 109, 218), Color.black);
}
```

```
private void btnEmployeeFocusLost(java.awt.event.FocusEvent evt)
{
    hover(btnEmployee, new Color(255, 167, 6), Color.white);
}
```

```
private void btnProductFocusGained(java.awt.event.FocusEvent evt)
{
    hover(btnProduct, new Color(1, 109, 218), Color.black);
}
```

```
private void btnProductFocusLost(java.awt.event.FocusEvent evt)
{
    hover(btnProduct, new Color(255, 167, 6), Color.white);
}
```

```
private void btnBillFocusGained(java.awt.event.FocusEvent evt)
{
    hover(btnBill, new Color(1, 109, 218), Color.black);
}
```

```
private void btnBillFocusLost(java.awt.event.FocusEvent evt)
{
    hover(btnBill, new Color(255, 167, 6), Color.white);
}
```

```
private void btnaccountFocusGained(java.awt.event.FocusEvent evt)
{
    hover(btnaccount, new Color(1, 109, 218), Color.black);
}
```

```
private void btnaccountFocusLost(java.awt.event.FocusEvent evt)
{
    hover(btnaccount, new Color(255, 167, 6), Color.white);
}
```

```
private void btnCustomerKeyPressed(java.awt.event.KeyEvent evt)
{
    if (evt.getKeyCode() == KeyEvent.VK_ENTER)
    {
        visibility(false, true, false, false, false, false, btn_addcust_menu);
        visibility(pnl_add_cust, pnl_edit_cust, pnl_cust_det);
        onIndicator(lid_add_cust, lid_edit_cust, lid_cust_det);
        txt_addcust_id.setText(getMaxID("customer"));
        txt_addcust_id.grabFocus();
    }
}

private void btnEmployeeKeyPressed(java.awt.event.KeyEvent evt)
{
    if (evt.getKeyCode() == KeyEvent.VK_ENTER)
    {
        visibility(false, false, true, false, false, false, btn_addemp_menu);
        visibility(pnl_add_emp, pnl_edit_emp, pnl_emp_det);
        onIndicator(lid_add_emp, lid_edit_emp, lid_emp_det);
        txt_addemp_id.setText(getMaxID("employee"));
    }
}

private void btnProductKeyPressed(java.awt.event.KeyEvent evt)
{
    if (evt.getKeyCode() == KeyEvent.VK_ENTER)
    {
        visibility(false, false, false, true, false, false, btn_addprod_menu);
        visibility(pnl_add_prod, pnl_edit_prod, pnl_prod_det);
        onIndicator(lid_addprod, lid_editprod, lid_proddet);
        txt_addprod_id.setText(getMaxID("product"));
        btn_addprod_menu.grabFocus();
    }
}

private void btnBillKeyPressed(java.awt.event.KeyEvent evt)
{
    if (evt.getKeyCode() == KeyEvent.VK_ENTER)
    {
        visibility(false, false, false, false, true, false, btn_createbill_menu);
        visibility(pnl_createbill, pnl_modifybill, pnl_bill_tab3);
        onIndicator(lid_createbill, lid_modifybill);
        fillCombobox("customer", "custnm", cmbbox_createbill_custnm);
        fillCombobox("product", "nm", cmbbox_createbill_prodnm);
    }
}

private void btnaccountKeyPressed(java.awt.event.KeyEvent evt)
```

```
{
    if (evt.getKeyCode() == KeyEvent.VK_ENTER)
    {
        visibility(false, false, false, false, false, true, btn_accsetting_menu);
        visibility(pnl_accsetting, pnl_acc_tab2, pnl_acc_tab3);
        onIndicator(lid_accsetting, lid_acc_menu2, lid_acc_menu3);
    }
}

private void btnEmployeeActionPerformed(java.awt.event.ActionEvent evt)
{
    visibility(false, false, true, false, false, false, btn_addemp_menu);
    visibility(pnl_add_emp, pnl_edit_emp, pnl_emp_det);
    onIndicator(lid_add_emp, lid_edit_emp, lid_emp_det);
    txt_addemp_id.setText(getMaxID("employee"));
    txt_addemp_id.grabFocus();
}

private void btnProductActionPerformed(java.awt.event.ActionEvent evt)
{
    visibility(false, false, false, true, false, false, btn_addprod_menu);
    visibility(pnl_add_prod, pnl_edit_prod, pnl_prod_det);
    onIndicator(lid_addprod, lid_editprod, lid_proddet);
    txt_addprod_id.setText(getMaxID("product"));
    btn_addprod_menu.grabFocus();
}

private void btnaccountActionPerformed(java.awt.event.ActionEvent evt)
{
    visibility(false, false, false, false, false, true, btn_accsetting_menu);
    visibility(pnl_accsetting, pnl_acc_tab2, pnl_acc_tab3);
    onIndicator(lid_accsetting, lid_acc_menu2, lid_acc_menu3);

    try
    {
        String usr = lbl_dash_usr.getText();
        con = dbconnection.getdbConnection();
        query = "select * from employee where nm = '" + usr + "'";
        st = con.createStatement();
        result = st.executeQuery(query);
        if (result.next())
        {
            String id = result.getString("id");
            String nm = result.getString("nm");
            String usrm = result.getString("usrm");
            String pswd = result.getString("pswd");
            String contact = result.getString("contact");
            String email = result.getString("email");
            String salary = result.getString("salary");
        }
    }
}
```

```
String address = result.getString("address");

lbl_accsetting_id.setText(id);
txt_accsetting_nm.setText(nm);
txt_accsetting_usrnm.setText(usrnm);
txt_accsetting_pswd.setText(pswd);
txt_accsetting_contact.setText(contact);
txt_accsetting_email.setText(email);
txt_accsetting_salary.setText(salary);
txt_accsetting_address.setText(address);
}
else
{
    JOptionPane.showMessageDialog(null, "No Id Found");
}

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

private void lbl_dash_usrMouseClicked(java.awt.event.MouseEvent evt)
{
    panel_dashboard.setVisible(false);
    panel_login.setVisible(true);
}

private void lbl_dash_welcomeMouseClicked(java.awt.event.MouseEvent evt)
{
    visibility(true, false, false, false, false, false, btn_login);
}

private void btn_addcust_menuMouseClicked(java.awt.event.MouseEvent evt)
{
    // lid1.setkStartColor(new Color(1, 109, 218));
    // lid1.setkEndColor(new Color(1, 109, 218));
    // lid1.setBackground(Color.black);

    btn_addcust_menu.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));

}

private void btn_addcust_menuMouseClicked(java.awt.event.MouseEvent evt)
{
    // lid1.setkStartColor(new Color(211, 211, 211));
    // lid1.setkEndColor(new Color(211, 211, 211));
    // lid1.setBackground(Color.white);
```

```
// kButton1.setkStartColor(new Color(255, 203, 109));
// kButton1.setkEndColor(new Color(255, 203, 109));
// kButton1.setBackground(Color.white);
}

private void btn_editcust_menuMouseEntered(java.awt.event.MouseEvent evt)
{
    btn_editcust_menu.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
}

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

private void btn_custdet_menuMouseEntered(java.awt.event.MouseEvent evt)
{
    btn_custdet_menu.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
}

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

private void btnCustomerMouseExited(java.awt.event.MouseEvent evt)
{
}

private void btn_addcust_menuFocusGained(java.awt.event.FocusEvent evt)
{
    hover(btn_addcust_menu, new Color(1, 109, 218), Color.white);
}

private void btn_addcust_menuFocusLost(java.awt.event.FocusEvent evt)
{
    hover(btn_addcust_menu, new Color(255, 203, 109), Color.black);
}

private void btn_editcust_menuFocusGained(java.awt.event.FocusEvent evt)
{
    hover(btn_editcust_menu, new Color(1, 109, 218), Color.white);
}

private void btn_editcust_menuFocusLost(java.awt.event.FocusEvent evt)
{
    hover(btn_editcust_menu, new Color(255, 203, 109), Color.black);
}
```

```
}

private void btn_custdet_menuFocusGained(java.awt.event.FocusEvent evt)
{
    hover(btn_custdet_menu, new Color(1, 109, 218), Color.white);
}

private void btn_custdet_menuFocusLost(java.awt.event.FocusEvent evt)
{
    hover(btn_custdet_menu, new Color(255, 203, 109), Color.black);
}

private void btn_addcust_menuKeyPressed(java.awt.event.KeyEvent evt)
{
    if (evt.getKeyCode() == KeyEvent.VK_ESCAPE)
    {
        btnCustomer.grabFocus();
    }
    if (evt.getKeyCode() == KeyEvent.VK_ENTER)
    {
        visibility(pnl_add_cust, pnl_edit_cust, pnl_cust_det);
        onIndicator(lid_add_cust, lid_edit_cust, lid_cust_det);
        txt_addcust_id.setText(getMaxID("customer"));
        txt_addcust_id.grabFocus();
    }
}

private void btn_editcust_menuKeyPressed(java.awt.event.KeyEvent evt)
{
    if (evt.getKeyCode() == KeyEvent.VK_ESCAPE)
    {
        btnCustomer.grabFocus();
    }
    if (evt.getKeyCode() == KeyEvent.VK_ENTER)
    {
        visibility(pnl_edit_cust, pnl_add_cust, pnl_cust_det);
        onIndicator(lid_edit_cust, lid_add_cust, lid_cust_det);
        bindCustomerTableData(table_editcust);
        txt_editcust_id.grabFocus();
    }
}

private void btn_custdet_menuKeyPressed(java.awt.event.KeyEvent evt)
{
    if (evt.getKeyCode() == KeyEvent.VK_ESCAPE)
    {
        btnCustomer.grabFocus();
    }
    if (evt.getKeyCode() == KeyEvent.VK_ENTER)
```

```
{
    visibility(pnl_cust_det, pnl_add_cust, pnl_edit_cust);
    onIndicator(lid_cust_det, lid_add_cust, lid_edit_cust);
    bindCustomerTableData(table_custdet);
    txt_custdet_custnm.grabFocus();
}
}

private void btn_addemp_menuFocusGained(java.awt.event.FocusEvent evt)
{
    hover(btn_addemp_menu, new Color(1, 109, 218), Color.white);
}

private void btn_addemp_menuFocusLost(java.awt.event.FocusEvent evt)
{
    hover(btn_addemp_menu, new Color(255, 203, 109), Color.black);
}

private void btn_addemp_menuMouseEntered(java.awt.event.MouseEvent evt)
{
    btn_addemp_menu.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
}

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

private void btn_addemp_menuKeyPressed(java.awt.event.KeyEvent evt)
{
    if (evt.getKeyCode() == KeyEvent.VK_ESCAPE)
    {
        btnEmployee.grabFocus();
    }
    if (evt.getKeyCode() == KeyEvent.VK_ENTER)
    {
        visibility(pnl_add_emp, pnl_edit_emp, pnl_emp_det);
        onIndicator(lid_add_emp, lid_edit_emp, lid_emp_det);
        txt_addemp_id.setText(getMaxID("employee"));
        txt_addemp_id.grabFocus();
    }
}

private void btn_editemp_menuFocusGained(java.awt.event.FocusEvent evt)
{
    hover(btn_editemp_menu, new Color(1, 109, 218), Color.white);
}

private void btn_editemp_menuFocusLost(java.awt.event.FocusEvent evt)
```

```
{
    hover(btn_editemp_menu, new Color(255, 203, 109), Color.black);
}

private void btn_editemp_menuMouseEntered(java.awt.event.MouseEvent evt)
{
    btn_editemp_menu.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
}

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

private void btn_editemp_menuKeyPressed(java.awt.event.KeyEvent evt)
{
    if (evt.getKeyCode() == KeyEvent.VK_ESCAPE)
    {
        btnEmployee.grabFocus();
    }
    if (evt.getKeyCode() == KeyEvent.VK_ENTER)
    {
        visibility(pnl_edit_emp, pnl_add_emp, pnl_emp_det);
        onIndicator(lid_edit_emp, lid_add_emp, lid_emp_det);
        bindEmployeeTableData(table_editemp);

        txt_editemp_nm.grabFocus();
    }
}

private void btn_empdet_menuFocusGained(java.awt.event.FocusEvent evt)
{
    hover(btn_empdet_menu, new Color(1, 109, 218), Color.white);
}

private void btn_empdet_menuFocusLost(java.awt.event.FocusEvent evt)
{
    hover(btn_empdet_menu, new Color(255, 203, 109), Color.black);
}

private void btn_empdet_menuMouseEntered(java.awt.event.MouseEvent evt)
{
    btn_empdet_menu.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
}

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



```
private void btn_empdet_menuKeyPressed(java.awt.event.KeyEvent evt)
{
    if (evt.getKeyCode() == KeyEvent.VK_ESCAPE)
    {
        btnEmployee.grabFocus();
    }
    if (evt.getKeyCode() == KeyEvent.VK_ENTER)
    {
        visibility(pnl_emp_det, pnl_add_emp, pnl_edit_emp);
        onIndicator(lid_emp_det, lid_add_emp, lid_edit_emp);
        txt_empdet_nm.grabFocus();
        bindEmployeeTableData(table_empdet);
    }
}

private void btn_addprod_menuFocusGained(java.awt.event.FocusEvent evt)
{
    hover(btn_addprod_menu, new Color(1, 109, 218), Color.white);
}

private void btn_addprod_menuFocusLost(java.awt.event.FocusEvent evt)
{
    hover(btn_addprod_menu, new Color(255, 203, 109), Color.black);
}

private void btn_addprod_menuMouseEntered(java.awt.event.MouseEvent evt)
{
    btn_addprod_menu.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
}

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

private void btn_addprod_menuKeyPressed(java.awt.event.KeyEvent evt)
{
    if (evt.getKeyCode() == KeyEvent.VK_ESCAPE)
    {
        btnProduct.grabFocus();
    }
    if (evt.getKeyCode() == KeyEvent.VK_ENTER)
    {
        visibility(pnl_add_prod, pnl_edit_prod, pnl_prod_det);
        onIndicator(lid_addprod, lid_editprod, lid_proddet);
        txt_addprod_id.grabFocus();
    }
}
```

```
private void btn_editprod_menuFocusGained(java.awt.event.FocusEvent evt)
{
    hover(btn_editprod_menu, new Color(1, 109, 218), Color.white);
}

private void btn_editprod_menuFocusLost(java.awt.event.FocusEvent evt)
{
    hover(btn_editprod_menu, new Color(255, 203, 109), Color.black);
}

private void btn_editprod_menuMouseEntered(java.awt.event.MouseEvent evt)
{
    btn_editprod_menu.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
}

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

private void btn_editprod_menuKeyPressed(java.awt.event.KeyEvent evt)
{
    if (evt.getKeyCode() == KeyEvent.VK_ESCAPE)
    {
        btnProduct.grabFocus();
    }
    if (evt.getKeyCode() == KeyEvent.VK_ENTER)
    {
        visibility(pnl_edit_prod, pnl_add_prod, pnl_prod_det);
        onIndicator(lid_editprod, lid_addprod, lid_proddet);
        bindProductTableData(table_editprod);

        txt_editprod_id.grabFocus();
    }
}

private void btn_proddet_menuFocusGained(java.awt.event.FocusEvent evt)
{
    hover(btn_proddet_menu, new Color(1, 109, 218), Color.white);
}

private void btn_proddet_menuFocusLost(java.awt.event.FocusEvent evt)
{
    hover(btn_proddet_menu, new Color(255, 203, 109), Color.black);
}

private void btn_proddet_menuMouseEntered(java.awt.event.MouseEvent evt)
{

```

```
        btn_proddet_menu.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
    }

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

    private void btn_proddet_menuKeyPressed(java.awt.event.KeyEvent evt)
    {
        if (evt.getKeyCode() == KeyEvent.VK_ESCAPE)
        {
            btnProduct.grabFocus();
        }
        if (evt.getKeyCode() == KeyEvent.VK_ENTER)
        {
            visibility(pnl_prod_det, pnl_add_prod, pnl_edit_prod);
            onIndicator(lid_proddet, lid_addprod, lid_editprod);
            bindProductTableData(table_proddet);

            txt_proddet_nm.grabFocus();
        }
    }

    private void btn_accsetting_menuFocusGained(java.awt.event.FocusEvent evt)
    {
        hover(btn_accsetting_menu, new Color(1, 109, 218), Color.white);
    }

    private void btn_accsetting_menuFocusLost(java.awt.event.FocusEvent evt)
    {
        hover(btn_accsetting_menu, new Color(255, 203, 109), Color.black);
    }

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

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

    private void btn_accsetting_menuKeyPressed(java.awt.event.KeyEvent evt)
    {
        if (evt.getKeyCode() == KeyEvent.VK_ESCAPE)
        {
            btnBill.grabFocus();
        }
    }
}
```

```
}
if (evt.getKeyCode() == KeyEvent.VK_ENTER)
{
    visibility(pnl_accsetting, pnl_acc_tab2, pnl_acc_tab3);
    onIndicator(lid_accsetting, lid_acc_menu2, lid_acc_menu3);
}
}

private void btn_acc_menu2FocusGained(java.awt.event.FocusEvent evt)
{
    hover(btn_acc_menu2, new Color(1, 109, 218), Color.white);
}

private void btn_acc_menu2FocusLost(java.awt.event.FocusEvent evt)
{
    hover(btn_acc_menu2, new Color(255, 203, 109), Color.black);
}

private void btn_acc_menu2MouseEntered(java.awt.event.MouseEvent evt)
{
    // TODO add your handling code here:
}

private void btn_acc_menu2MouseExited(java.awt.event.MouseEvent evt)
{
    // TODO add your handling code here:
}

private void btn_acc_menu2KeyPressed(java.awt.event.KeyEvent evt)
{
    if (evt.getKeyCode() == KeyEvent.VK_ESCAPE)
    {
        btnBill.grabFocus();
    }
    if (evt.getKeyCode() == KeyEvent.VK_ENTER)
    {
        visibility(pnl_acc_tab2, pnl_accsetting, pnl_acc_tab3);
        onIndicator(lid_acc_menu2, lid_accsetting, lid_acc_menu3);
    }
}

private void btn_acc_menu3FocusGained(java.awt.event.FocusEvent evt)
{
    hover(btn_acc_menu3, new Color(1, 109, 218), Color.white);
}

private void btn_acc_menu3FocusLost(java.awt.event.FocusEvent evt)
{
    hover(btn_acc_menu3, new Color(255, 203, 109), Color.black);
}
```

```
}

private void btn_acc_menu3MouseEntered(java.awt.event.MouseEvent evt)
{
    // TODO add your handling code here:
}

private void btn_acc_menu3MouseExited(java.awt.event.MouseEvent evt)
{
    // TODO add your handling code here:
}

private void btn_acc_menu3KeyPressed(java.awt.event.KeyEvent evt)
{
    if (evt.getKeyCode() == KeyEvent.VK_ESCAPE)
    {
        btnBill.grabFocus();
    }
    if (evt.getKeyCode() == KeyEvent.VK_ENTER)
    {
        visibility(pnl_acc_tab3, pnl_accsetting, pnl_acc_tab2);
        onIndicator(lid_acc_menu3, lid_accsetting, lid_acc_menu2);
    }
}

private void btn_createbill_menuFocusGained(java.awt.event.FocusEvent evt)
{
    hover(btn_createbill_menu, new Color(1, 109, 218), Color.white);
}

private void btn_createbill_menuFocusLost(java.awt.event.FocusEvent evt)
{
    hover(btn_createbill_menu, new Color(255, 203, 109), Color.black);
}

private void btn_createbill_menuMouseEntered(java.awt.event.MouseEvent evt)
{
    btn_createbill_menu.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
}

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

private void btn_createbill_menuKeyPressed(java.awt.event.KeyEvent evt)
{
    if (evt.getKeyCode() == KeyEvent.VK_ESCAPE)
    {
```

```
        btnaccount.grabFocus();
    }
    if (evt.getKeyCode() == KeyEvent.VK_ENTER)
    {
        visibility(pnl_createbill, pnl_modifybill, pnl_bill_tab3);
        onIndicator(lid_createbill, lid_modifybill);
        lbl_billno_no.setText(getMaxID("bill"));
        fillCombobox("customer", "custnm", cmbbox_createbill_custnm);
        cmbbox_createbill_custnm.grabFocus();
    }
}

private void btn_modifybill_menuFocusGained(java.awt.event.FocusEvent evt)
{
    hover(btn_modifybill_menu, new Color(1, 109, 218), Color.white);
}

private void btn_modifybill_menuFocusLost(java.awt.event.FocusEvent evt)
{
    hover(btn_modifybill_menu, new Color(255, 203, 109), Color.black);
}

private void btn_modifybill_menuMouseEntered(java.awt.event.MouseEvent evt)
{
    btn_modifybill_menu.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
}

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

private void btn_modifybill_menuKeyPressed(java.awt.event.KeyEvent evt)
{
    if (evt.getKeyCode() == KeyEvent.VK_ESCAPE)
    {
        btnaccount.grabFocus();
    }
    if (evt.getKeyCode() == KeyEvent.VK_ENTER)
    {
        visibility(pnl_modifybill, pnl_createbill, pnl_bill_tab3);
        onIndicator(lid_modifybill, lid_createbill);
    }
}

private void btn_modifybill_menuActionPerformed(java.awt.event.ActionEvent evt)
{
    visibility(pnl_modifybill, pnl_createbill, pnl_bill_tab3);
    onIndicator(lid_modifybill, lid_createbill);
}
```

```
}

private void btn_addcust_menuActionPerformed(java.awt.event.ActionEvent evt)
{
    visibility(pnl_add_cust, pnl_edit_cust, pnl_cust_det);
    onIndicator(lid_add_cust, lid_edit_cust, lid_cust_det);
    txt_editcust_id.setText(getMaxID("customer"));
    txt_addcust_id.grabFocus();
}

private void btn_editcust_menuActionPerformed(java.awt.event.ActionEvent evt)
{
    visibility(pnl_edit_cust, pnl_add_cust, pnl_cust_det);
    onIndicator(lid_edit_cust, lid_add_cust, lid_cust_det);
    bindCustomerTableData(table_editcust);
    txt_editcust_id.grabFocus();
}

private void btn_custdet_menuActionPerformed(java.awt.event.ActionEvent evt)
{
    visibility(pnl_cust_det, pnl_add_cust, pnl_edit_cust);
    onIndicator(lid_cust_det, lid_add_cust, lid_edit_cust);
    bindCustomerTableData(table_custdet);
    txt_custdet_custnm.grabFocus();
}

private void btn_addemp_menuActionPerformed(java.awt.event.ActionEvent evt)
{
    visibility(pnl_add_emp, pnl_edit_emp, pnl_emp_det);
    onIndicator(lid_add_emp, lid_edit_emp, lid_emp_det);
    txt_addemp_id.setText(getMaxID("employee"));
    txt_addemp_id.grabFocus();
}

private void btn_editemp_menuActionPerformed(java.awt.event.ActionEvent evt)
{
    visibility(pnl_edit_emp, pnl_add_emp, pnl_emp_det);
    onIndicator(lid_edit_emp, lid_add_emp, lid_emp_det);
    bindEmployeeTableData(table_editemp);

    txt_editemp_nm.grabFocus();
}

private void btn_empdet_menuActionPerformed(java.awt.event.ActionEvent evt)
{
    visibility(pnl_emp_det, pnl_add_emp, pnl_edit_emp);
    onIndicator(lid_emp_det, lid_add_emp, lid_edit_emp);
    txt_empdet_nm.grabFocus();
    bindEmployeeTableData(table_empdet);
}
```

```
}

private void btn_addprod_menuActionPerformed(java.awt.event.ActionEvent evt)
{
    visibility(pnl_add_prod, pnl_edit_prod, pnl_prod_det);
    onIndicator(lid_addprod, lid_editprod, lid_proddet);
    txt_addprod_id.grabFocus();
}

private void btn_editprod_menuActionPerformed(java.awt.event.ActionEvent evt)
{
    visibility(pnl_edit_prod, pnl_add_prod, pnl_prod_det);
    onIndicator(lid_editprod, lid_addprod, lid_proddet);
    bindProductTableData(table_editprod);
    txt_editprod_id.grabFocus();
}

private void btn_proddet_menuActionPerformed(java.awt.event.ActionEvent evt)
{
    visibility(pnl_prod_det, pnl_add_prod, pnl_edit_prod);
    onIndicator(lid_proddet, lid_addprod, lid_editprod);
    bindProductTableData(table_proddet);

    txt_proddet_nm.grabFocus();
}

private void btn_accsetting_menuActionPerformed(java.awt.event.ActionEvent evt)
{
    visibility(pnl_accsetting, pnl_acc_tab2, pnl_acc_tab3);
    onIndicator(lid_accsetting, lid_acc_menu2, lid_acc_menu3);
}

private void btn_acc_menu2ActionPerformed(java.awt.event.ActionEvent evt)
{
    visibility(pnl_acc_tab2, pnl_accsetting, pnl_acc_tab3);
    onIndicator(lid_acc_menu2, lid_accsetting, lid_acc_menu3);
}

private void btn_acc_menu3ActionPerformed(java.awt.event.ActionEvent evt)
{
    visibility(pnl_acc_tab3, pnl_accsetting, pnl_acc_tab2);
    onIndicator(lid_acc_menu3, lid_accsetting, lid_acc_menu2);
}

private void btn_createbill_menuActionPerformed(java.awt.event.ActionEvent evt)
{
    visibility(pnl_createbill, pnl_modifybill, pnl_bill_tab3);
    onIndicator(lid_createbill, lid_modifybill);
    lbl_billno_no.setText(getMaxID("bill"));
}
```



```
fillCombobox("customer", "custnm", cmbbox_createbill_custnm);
cmbbox_createbill_custnm.grabFocus();
}

private void btnBillActionPerformed(java.awt.event.ActionEvent evt)
{
    visibility(false, false, false, false, true, false, btn_createbill_menu);
    visibility(pnl_createbill, pnl_modifybill, pnl_bill_tab3);
    onIndicator(lid_createbill, lid_modifybill);
    lbl_billno_no.setText(getMaxID("bill"));
    fillCombobox("customer", "custnm", cmbbox_createbill_custnm);
    fillCombobox("product", "nm", cmbbox_createbill_prodnm);
}

private void btn_editcust_editFocusGained(java.awt.event.FocusEvent evt)
{
    hover(btn_editcust_edit, new Color(1, 109, 218), Color.BLACK);
}

private void btn_editcust_editFocusLost(java.awt.event.FocusEvent evt)
{
    hover(btn_editcust_edit, new Color(255, 167, 6), Color.WHITE);
}

private void btn_editcust_editMouseEntered(java.awt.event.MouseEvent evt)
{
    btn_editcust_edit.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
}

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

private void btn_editcust_editActionPerformed(java.awt.event.ActionEvent evt)
{
    if (isEmpty(txt_editcust_id, txt_editcust_nm, txt_editcust_contact, txt_editcust_email,
txt_editcust_shopnm, txt_editcust_shopaddr))
    {
        if (isIdExist(txt_editcust_id.getText(), "customer"))
        {
            try
            {
                con = dbconnection.getdbConnection();
                query = "update customer set custnm = ?, contact = ?, email = ?, shopnm = ?, address = ?
where id = ?";

                pst = con.prepareStatement(query);
                pst.setString(1, txt_editcust_nm.getText());
```

```
pst.setString(2, txt_editcust_contact.getText());
pst.setString(3, txt_editcust_email.getText());
pst.setString(4, txt_editcust_shopnm.getText());
pst.setString(5, txt_editcust_shopaddr.getText());
pst.setString(6, txt_editcust_id.getText());

pst.executeUpdate();
bindCustomerTableData(table_editcust);

JOptionPane.showMessageDialog(null, "Customer Details Updated Successfully");

clearTextFields(txt_editcust_id, txt_editcust_nm, txt_addcust_contact,
txt_editcust_email, txt_editcust_shopnm, txt_editcust_shopaddr);
txt_editcust_id.grabFocus();
}
catch (Exception e)
{
    JOptionPane.showMessageDialog(null, e.getMessage());
}
}
else
{
    JOptionPane.showMessageDialog(null, "No ID Found");
    txt_editcust_id.grabFocus();
}
}
}

private void btn_editcust_editKeyPressed(java.awt.event.KeyEvent evt)
{

}

private void btn_editcust_rmvFocusLost(java.awt.event.FocusEvent evt)
{
    hover(btn_editcust_rmv, new Color(255, 167, 6), Color.WHITE);
}

private void btn_editcust_rmvMouseEntered(java.awt.event.MouseEvent evt)
{
    btn_editcust_rmv.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
}

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

private void btn_editcust_rmvKeyPressed(java.awt.event.KeyEvent evt)
```

```
{
    // TODO add your handling code here:
}

private void btn_addcust_addFocusGained(java.awt.event.FocusEvent evt)
{
    hover(btn_addcust_add, new Color(1, 109, 218), Color.BLACK);
}

private void btn_addcust_addFocusLost(java.awt.event.FocusEvent evt)
{
    hover(btn_addcust_add, new Color(255, 167, 6), Color.WHITE);
}

private void btn_addcust_addActionPerformed(java.awt.event.ActionEvent evt)
{
    if (isEmpty(txt_addcust_id, txt_addcust_nm, txt_addcust_contact, txt_addcust_email,
txt_addcust_shopnm, txt_addcust_shopaddr))
    {
        if (isIdExist(txt_addcust_id.getText(), "customer"))
        {
            JOptionPane.showMessageDialog(null, "ID already exist");
            txt_addcust_id.grabFocus();
        }
        else
        {
            try
            {
                con = dbconnection.getdbConnection();
                query = "insert into customer values(?, ?, ?, ?, ?, ?)";
                pst = con.prepareStatement(query);

                pst.setString(1, txt_addcust_id.getText());
                pst.setString(2, txt_addcust_nm.getText());
                pst.setString(3, txt_addcust_contact.getText());
                pst.setString(4, txt_addcust_email.getText());
                pst.setString(5, txt_addcust_shopnm.getText());
                pst.setString(6, txt_addcust_shopaddr.getText());

                pst.executeUpdate();

                JOptionPane.showMessageDialog(null, "Record Inserted Successfully !");
                clearTextFields(txt_addcust_id, txt_addcust_nm, txt_addcust_contact,
txt_addcust_email, txt_addcust_shopnm, txt_addcust_shopaddr);
                txt_addcust_id.setText(getMaxID("customer"));
                txt_addcust_id.grabFocus();
            }
            catch (Exception e)
            {
            }
        }
    }
}
```

```
JOptionPane.showMessageDialog(null, e.getMessage());
    }
}
}

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

private void btn_addemp_addFocusGained(java.awt.event.FocusEvent evt)
{
    hover(btn_addemp_add, new Color(1, 109, 218), Color.BLACK);
}

private void btn_addemp_addFocusLost(java.awt.event.FocusEvent evt)
{
    hover(btn_addemp_add, new Color(255, 167, 6), Color.WHITE);
}

private void btn_addemp_addMouseEntered(java.awt.event.MouseEvent evt)
{
    btn_addemp_add.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
}

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

private void btn_addemp_addActionPerformed(java.awt.event.ActionEvent evt)
{
    if (isEmpty(txt_addemp_id, txt_addemp_nm, txt_addemp_usnm, txt_addemp_contact,
txt_addemp_email, txt_addemp_salary, txt_addemp_addr))
    {
        if (isIdExist(txt_addemp_id.getText(), "employee"))
        {
            JOptionPane.showMessageDialog(null, "ID already exist!");
            txt_addemp_id.grabFocus();
        }
        else
        {
            try
            {
                con = dbconnection.getdbConnection();
                query = "insert into employee (id, nm, usnm, pswd, contact, email, salary, address)
values(?, ?, ?, ?, ?, ?, ?, ?)";
```

```
pst = con.prepareStatement(query);
pst.setString(1, txt_addemp_id.getText());
pst.setString(2, txt_addemp_nm.getText());
pst.setString(3, txt_addemp_usrnm.getText());
pst.setString(4, txt_addemp_pswd.getText());
pst.setString(5, txt_addemp_contact.getText());
pst.setString(6, txt_addemp_email.getText());
pst.setString(7, txt_addemp_salary.getText());
pst.setString(8, txt_addemp_addr.getText());

pst.executeUpdate();

JOptionPane.showMessageDialog(null, "Employee Added Successfully!");
clearTextFields(txt_addemp_id, txt_addemp_nm, txt_addemp_usrnm,
txt_addemp_pswd, txt_addemp_contact, txt_addemp_email, txt_addemp_salary,
txt_addemp_addr);
txt_addemp_id.setText(getMaxID("employee"));
txt_addemp_id.grabFocus();
}
catch (Exception e)
{
    JOptionPane.showMessageDialog(null, e.getMessage());
}
}
}

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

private void btn_addprodFocusGained(java.awt.event.FocusEvent evt)
{
    hover(btn_addprod, new Color(1, 109, 218), Color.BLACK);
}

private void btn_addprodFocusLost(java.awt.event.FocusEvent evt)
{
    hover(btn_addprod, new Color(255, 167, 6), Color.WHITE);
}

private void btn_addprodMouseEntered(java.awt.event.MouseEvent evt)
{
    btn_addprod.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
}

private void btn_addprodMouseExited(java.awt.event.MouseEvent evt)
{

```

```
// TODO add your handling code here:
}

private void btn_addprodActionPerformed(java.awt.event.ActionEvent evt)
{
    if (isEmpty(txt_addprod_id, txt_addprod_nm, txt_addprod_costprice,
txt_addprod_sellingprice, txt_addprod_quantity) && isCmbboxDefaultSelected("--SELECT BRAND--",
cmbbox_addprod_brand) == false)
    {
        if (isIdExist(txt_addprod_id.getText(), "product"))
        {
            JOptionPane.showMessageDialog(null, "ID already exist");
            txt_addprod_id.requestFocus();
        }
        else
        {
            try
            {
                con = dbconnection.getdbConnection();
                query = "insert into product (id, nm, costprice, sellingprice, brand, quantity) values(?, ?, ?,
?, ?, ?)";

                pst = con.prepareStatement(query);
                pst.setString(1, txt_addprod_id.getText());
                pst.setString(2, txt_addprod_nm.getText());
                pst.setString(3, txt_addprod_costprice.getText());
                pst.setString(4, txt_addprod_sellingprice.getText());
                pst.setString(5, cmbbox_addprod_brand.getSelectedItem().toString());
                pst.setString(6, txt_addprod_quantity.getText());

                pst.executeUpdate();
                JOptionPane.showMessageDialog(null, "Record inserted Successfull");
                clearTextFields(txt_addprod_id, txt_addprod_nm, txt_addprod_costprice,
txt_addprod_sellingprice, cmbbox_addprod_brand, txt_addprod_quantity);
                txt_addprod_id.setText(getMaxID("product"));
            }
            catch (Exception e)
            {
                JOptionPane.showMessageDialog(null, e.getMessage());
            }
        }
    }
}

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

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

private void btn_addcust_addMouseEntered(java.awt.event.MouseEvent evt)
{
    btn_addcust_add.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
}

private void btn_editcust_rmvActionPerformed(java.awt.event.ActionEvent evt)
{
    if (txt_editcust_id.getText().equals(""))
    {
        txt_editcust_id.grabFocus();
    }
    else
    {
        if (isIdExist(txt_editcust_id.getText(), "customer"))
        {
            int confirmMsg = JOptionPane.showConfirmDialog(null, "Confirm Deletion ?");
            if (confirmMsg == 0)
            {
                try
                {
                    con = dbconnection.getdbConnection();
                    query = "delete from customer where id = ?";

                    pst = con.prepareStatement(query);
                    pst.setString(1, txt_editcust_id.getText());

                    pst.executeUpdate();
                    bindCustomerTableData(table_editcust);
                    JOptionPane.showMessageDialog(null, "Record Deleted Successfully !");

                    clearTextFields(txt_editcust_id, txt_editcust_nm, txt_addcust_contact,
txt_editcust_email, txt_editcust_shopnm, txt_editcust_shopaddr);
                    txt_editcust_id.grabFocus();
                }
                catch (Exception e)
                {
                    JOptionPane.showMessageDialog(null, e.getMessage());
                }
            }
        }
    }
}

private void btn_editcust_rmvFocusGained(java.awt.event.FocusEvent evt)
```

```
{
    hover(btn_editcust_rmv, new Color(1, 109, 218), Color.BLACK);
}

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

private void btn_editemp_rmvActionPerformed(java.awt.event.ActionEvent evt)
{
    if (txt_editemp_id.getText().equals(""))
    {
        txt_editemp_nm.grabFocus();
    }
    else
    {
        if (isIdExist(txt_editemp_id.getText(), "employee"))
        {
            try
            {
                con = dbconnection.getdbConnection();
                String empnm = txt_editemp_nm.getText();
                query = "delete from employee where nm = '" + empnm + "'";

                pst = con.prepareStatement(query);

                pst.executeUpdate();
                JOptionPane.showMessageDialog(null, "Employee Removed Successfully !");
                bindEmployeeTableData(table_editemp);
                clearTextFields(txt_editemp_nm, txt_editemp_usrnm, txt_editemp_contact,
txt_editemp_email, txt_editemp_salary, txt_editemp_addr);
                txt_editemp_pswd.setText("");
            }
            catch (Exception e)
            {
                JOptionPane.showMessageDialog(null, e.getMessage());
            }
        }
        else
        {
            JOptionPane.showMessageDialog(null, "ID not found");
        }
    }
}

private void btn_editemp_rmvMouseExited(java.awt.event.MouseEvent evt)
{
}
```



```
// TODO add your handling code here:
}

private void btn_editemp_rmvMouseEntered(java.awt.event.MouseEvent evt)
{
    btn_editemp_rmv.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
}

private void btn_editemp_rmvFocusLost(java.awt.event.FocusEvent evt)
{
    hover(btn_editemp_rmv, new Color(255, 167, 6), Color.WHITE);
}

private void btn_editemp_rmvFocusGained(java.awt.event.FocusEvent evt)
{
    hover(btn_editemp_rmv, new Color(1, 109, 218), Color.BLACK);
}

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

private void btn_editemp_editActionPerformed(java.awt.event.ActionEvent evt)
{
    if (isEmpty(txt_editemp_id, txt_editemp_nm, txt_editemp_usnm, txt_editemp_contact,
txt_editemp_email, txt_editemp_salary, txt_editemp_addr))
    {
        if (isEmpty(txt_editemp_id.getText(), "employee"))
        {
            try
            {
                con = dbconnection.getdbConnection();
                query = "update employee set nm = ?, usnm = ?, pswd = ?, contact = ?, email = ?, salary =
?, address = ? where usnm = ? ";

                pst = con.prepareStatement(query);
                pst.setString(1, txt_editemp_nm.getText());
                pst.setString(2, txt_editemp_usnm.getText());
                pst.setString(3, txt_editemp_pswd.getText());
                pst.setString(4, txt_editemp_contact.getText());
                pst.setString(5, txt_editemp_email.getText());
                pst.setString(6, txt_editemp_salary.getText());
                pst.setString(7, txt_editemp_addr.getText());
                pst.setString(8, txt_editemp_usnm.getText());

                pst.executeUpdate();
                JOptionPane.showMessageDialog(null, "Employee Details Updated Successfully!");
                bindEmployeeTableData(table_editemp);
            }
            catch (Exception ex)
            {
                JOptionPane.showMessageDialog(null, ex.getMessage());
            }
        }
    }
}
```

```
        clearTextFields(txt_editemp_nm, txt_editemp_usrnm, txt_editemp_contact,
txt_editemp_email, txt_editemp_salary, txt_editemp_addr);
        txt_editemp_pswd.setText("");
    }
    catch (Exception e)
    {
        JOptionPane.showMessageDialog(null, e.getMessage());
    }
}
else
{
    JOptionPane.showMessageDialog(null, "No ID Found");
    txt_editemp_id.grabFocus();
}
}
}

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

private void btn_editemp_editMouseEntered(java.awt.event.MouseEvent evt)
{
    btn_editemp_edit.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
}

private void btn_editemp_editFocusLost(java.awt.event.FocusEvent evt)
{
    hover(btn_editemp_edit, new Color(255, 167, 6), Color.WHITE);
}

private void btn_editemp_editFocusGained(java.awt.event.FocusEvent evt)
{
    hover(btn_editemp_edit, new Color(1, 109, 218), Color.BLACK);
}

private void txt_custdet_custnmKeyReleased(java.awt.event.KeyEvent evt)
{
    if (txt_custdet_custnm.getText().equals(""))
    {
        bindCustomerTableData(table_custdet);
    }
    else
    {
        String custnm = txt_custdet_custnm.getText();
        try
        {

```

```
con = dbconnection.getdbConnection();
query = "select * from customer where custnm like '" + custnm + "%'";

tableModel = (DefaultTableModel) table_custdet.getModel();
tableModel.setRowCount(0);

pst = con.prepareStatement(query);
result = pst.executeQuery();

String id, nm, contact, email, shpnm, shpaddr;

while (result.next())
{
    id = result.getString(1);
    nm = result.getString(2);
    contact = result.getString(3);
    email = result.getString(4);
    shpnm = result.getString(5);
    shpaddr = result.getString(6);

    String[] rows =
    {
        id, nm, contact, email, shpnm, shpaddr
    };

    tableModel.addRow(rows);
}
catch (Exception e)
{
    JOptionPane.showMessageDialog(null, e.getMessage());
}
}

private void txt_empdet_nmKeyReleased(java.awt.event.KeyEvent evt)
{
    if (txt_empdet_nm.getText().equals(""))
    {
        bindEmployeeTableData(table_empdet);
    }
    else
    {
        String empnm = txt_empdet_nm.getText();
        try
        {
            con = dbconnection.getdbConnection();
            query = "select * from employee where nm like '" + empnm + "%'";
```

```
tableModel = (DefaultTableModel) table_empdet.getModel();
tableModel.setRowCount(0);

pst = con.prepareStatement(query);
result = pst.executeQuery();

String id, nm, contact, email, salary, addr, pswd;

while (result.next())
{
    id = result.getString(1);
    nm = result.getString(2);
    contact = result.getString(3);
    email = result.getString(4);
    salary = result.getString(5);
    addr = result.getString(6);
    pswd = result.getString(7);

    String[] rows =
    {
        id, nm, contact, email, salary, addr, pswd
    };

    tableModel.addRow(rows);
}
catch (Exception e)
{
    JOptionPane.showMessageDialog(null, e.getMessage());
}
}

private void table_editcustMouseClicked(java.awt.event.MouseEvent evt)
{
    table_editcust = (JTable) evt.getSource();

    // Database table column index starts from "1", while Jtable column index starts from "0".
    if (evt.getClickCount() == 1 && table_editcust.getSelectedRow() != -1)
    {
        int i = table_editcust.getSelectedRow();
        tableModel = (DefaultTableModel) table_editcust.getModel();
        txt_editcust_id.setText(tableModel.getValueAt(i, 0).toString());
        txt_editcust_nm.setText(tableModel.getValueAt(i, 1).toString());
        txt_editcust_contact.setText(tableModel.getValueAt(i, 2).toString());
        txt_editcust_email.setText(tableModel.getValueAt(i, 3).toString());
        txt_editcust_shopnm.setText(tableModel.getValueAt(i, 4).toString());
        txt_editcust_shopaddr.setText(tableModel.getValueAt(i, 5).toString());
    }
}
```

```
}

private void table_editempMouseClicked(java.awt.event.MouseEvent evt)
{
    table_editemp = (JTable) evt.getSource();

    // Database table column index starts from "1", while Jtable column index starts from "0".
    if (evt.getClickCount() == 1 && table_editemp.getSelectedRow() != -1)
    {
        int i = table_editemp.getSelectedRow();
        tableModel = (DefaultTableModel) table_editemp.getModel();

        txt_editemp_nm.setText(tableModel.getValueAt(i, 1).toString());
        txt_editemp_usrnm.setText(tableModel.getValueAt(i, 2).toString());
        txt_editemp_pswd.setText(tableModel.getValueAt(i, 3).toString());
        txt_editemp_contact.setText(tableModel.getValueAt(i, 4).toString());
        txt_editemp_email.setText(tableModel.getValueAt(i, 5).toString());
        txt_editemp_salary.setText(tableModel.getValueAt(i, 6).toString());
        txt_editemp_addr.setText(tableModel.getValueAt(i, 7).toString());
    }
}

private void table_editprodMouseClicked(java.awt.event.MouseEvent evt)
{
    table_editprod = (JTable) evt.getSource();

    // Database table column index starts from "1", while Jtable column index starts from "0".
    if (evt.getClickCount() == 1 && table_editprod.getSelectedRow() != -1)
    {
        int i = table_editprod.getSelectedRow();
        tableModel = (DefaultTableModel) table_editprod.getModel();

        txt_editprod_id.setText(tableModel.getValueAt(i, 0).toString());
        txt_editprod_nm.setText(tableModel.getValueAt(i, 1).toString());
        txt_editprod_costprice.setText(tableModel.getValueAt(i, 2).toString());
        txt_editprod_sellingprice.setText(tableModel.getValueAt(i, 3).toString());
        cmbbox_editprod_brand.setSelectedItem(tableModel.getValueAt(i, 4).toString());
        txt_editprod_quantity.setText(tableModel.getValueAt(i, 5).toString());
    }
}

private void btn_editprod_editFocusGained(java.awt.event.FocusEvent evt)
{
    hover(btn_editprod_edit, new Color(1, 109, 218), Color.BLACK);
}

private void btn_editprod_editFocusLost(java.awt.event.FocusEvent evt)
{
    hover(btn_editprod_edit, new Color(255, 167, 6), Color.WHITE);
}
```

```
}

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

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

private void btn_editprod_editActionPerformed(java.awt.event.ActionEvent evt)
{
    if (isEmpty(txt_editprod_id, txt_editprod_nm, txt_editprod_costprice,
txt_editprod_sellingprice, txt_editprod_quantity) && isCmbboxDefaultSelected("--SELECT BRAND--",
cmbbox_editprod_brand) == false)
    {
        if (isIdExist(txt_editprod_id.getText(), "product"))
        {
            try
            {
                con = dbconnection.getdbConnection();
                query = "update product set nm = ?, costprice = ?, sellingprice = ?, brand = ?, quantity = ?
where id = ? ";

                pst = con.prepareStatement(query);
                pst.setString(1, txt_editprod_nm.getText());
                pst.setString(2, txt_editprod_costprice.getText());
                pst.setString(3, txt_editprod_sellingprice.getText());
                pst.setString(4, cmbbox_editprod_brand.getSelectedItem().toString());
                pst.setString(5, txt_editprod_quantity.getText());
                pst.setString(6, txt_editprod_id.getText());
                pst.executeUpdate();
                JOptionPane.showMessageDialog(null, "Product Updated Successfully!");
                bindProductTableData(table_editprod);
            }
            catch (Exception e)
            {
                JOptionPane.showMessageDialog(null, e.getMessage());
            }
        }
        else
        {
            JOptionPane.showMessageDialog(null, "ID does not exist");
            txt_editprod_id.grabFocus();
        }
    }
}
```

```
}

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

private void btn_editprod_rmvFocusGained(java.awt.event.FocusEvent evt)
{
    hover(btn_editprod_rmv, new Color(1, 109, 218), Color.BLACK);
}

private void btn_editprod_rmvFocusLost(java.awt.event.FocusEvent evt)
{
    hover(btn_editprod_rmv, new Color(255, 167, 6), Color.WHITE);
}

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

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

private void btn_editprod_rmvActionPerformed(java.awt.event.ActionEvent evt)
{
    if (txt_editprod_id.getText().equals(""))
    {
        txt_editprod_id.grabFocus();
    }
    else
    {
        if (isIdExist(txt_editprod_id.getText(), "product"))
        {
            try
            {
                con = dbconnection.getdbConnection();
                query = "delete from product where id = ?";

                pst = con.prepareStatement(query);
                pst.setString(1, txt_editprod_id.getText());

                pst.executeUpdate();
                JOptionPane.showMessageDialog(null, "Product Deleted Successfully !");
                bindProductTableData(table_editprod);
            }
        }
    }
}
```

```
        catch (Exception e)
        {
            JOptionPane.showMessageDialog(null, e.getMessage());
        }
    }
    else
    {
        JOptionPane.showMessageDialog(null, "ID does not exist");
    }
}

}

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

private void txt_proddet_nmKeyReleased(java.awt.event.KeyEvent evt)
{
    if (txt_proddet_nm.getText().equals(""))
    {
        bindProductTableData(table_proddet);
    }
    else
    {
        String prodnm = txt_proddet_nm.getText();
        try
        {
            con = dbconnection.getdbConnection();
            query = "select * from product where nm like '" + prodnm + "%'";

            tableModel = (DefaultTableModel) table_proddet.getModel();
            tableModel.setRowCount(0);

            pst = con.prepareStatement(query);
            result = pst.executeQuery();

            String id, nm, costrprice, sellingprice, brand, quantity;

            while (result.next())
            {
                id = result.getString(1);
                nm = result.getString(2);
                costrprice = result.getString(3);
                sellingprice = result.getString(4);
                brand = result.getString(5);
                quantity = result.getString(6);
            }
        }
        catch (Exception e)
        {
            JOptionPane.showMessageDialog(null, e.getMessage());
        }
    }
}
```



```
        String[] rows =
        {
            id, nm, costrprice, sellingprice, brand, quantity
        };

        tableModel.addRow(rows);
    }
}
catch (Exception e)
{
    JOptionPane.showMessageDialog(null, e.getMessage());
}
}

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

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

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

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

private void btn_createbill_addActionPerformed(java.awt.event.ActionEvent evt)
{
    tableModel = (DefaultTableModel) table_createbill.getModel();

    String custnm, custid, contact, address, prodnm, prodid;
    int price = 0, quantity = 0, gtototal = 0, discount = 0, tax = 0, netttotal = 0, unpaid = 0;

    custnm = cmbbox_createbill_custnm.getSelectedItem().toString();
    custid = txt_createbill_custid.getText();
    contact = txt_createbill_contact.getText();
    address = txt_createbill_addr.getText();
    prodnm = cmbbox_createbill_prodnm.getSelectedItem().toString();
    prodid = txt_createbill_prodid.getText();
}
```

```
price = Integer.parseInt(txt_createbill_prodprice.getText());
quantity = Integer.parseInt(spin_createbill_quantity.getValue().toString());
discount = Integer.parseInt(txt_createbill_discount.getText());
tax = Integer.parseInt(txt_createbill_tax.getText());
unpaid = Integer.parseInt(txt_createbill_unpaid.getText());

gtotal = price * quantity;
lbl_grosstotal.setText(Integer.toString(Integer.parseInt(lbl_grosstotal.getText()) + gtotal));

discount = (gtotal * discount) / 100;
lbl_discount.setText(Integer.toString(Integer.parseInt(lbl_discount.getText()) + discount));

tax = (gtotal * tax) / 100;
lbl_tax.setText(Integer.toString(Integer.parseInt(lbl_tax.getText()) + tax));

unpaid = Integer.parseInt(txt_createbill_unpaid.getText());
lbl_unpaid.setText(Integer.toString(Integer.parseInt(lbl_unpaid.getText()) + unpaid));

nettotal = gtotal - discount + tax;
lbl_nettotal.setText(Integer.toString(Integer.parseInt(lbl_nettotal.getText()) + nettotal));

String[] rows =
{
    custnm, contact, prodnm, Integer.toString(price), Integer.toString(quantity),
Integer.toString(nettotal)
};

tableModel.addRow(rows);

// lbl_discount.setText(Integer.toString(Integer.parseInt(lbl_discount.getText()) + discount));
// lbl_grosstotal.setText(Integer.toString(Integer.parseInt(lbl_grosstotal.getText()) + gtotal));
// lbl_tax.setText(Integer.toString(Integer.parseInt(lbl_tax.getText()) + tax));
// lbl_unpaid.setText(Integer.toString(Integer.parseInt(lbl_unpaid.getText()) + unpaid));
// lbl_nettotal.setText(Integer.toString(Integer.parseInt(lbl_nettotal.getText()) + nettotal));
}

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

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

private void btn_createbill_saveFocusLost(java.awt.event.FocusEvent evt)
{
    // TODO add your handling code here:
}
```

```

    }

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

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

    private void btn_createbill_saveActionPerformed(java.awt.event.ActionEvent evt)
    {
        if (isIdExist(lbl_billno_no.getText(), "bill"))
        {
            JOptionPane.showMessageDialog(null, "ID already exist");
        }
        else
        {
            try
            {
                con = dbconnection.getdbConnection();
                query = "insert into bill values(?, ?, ?, ?, ?, ?, ?, ?, ?)";
                pst = con.prepareStatement(query);
                pst.setString(1, lbl_billno_no.getText());
                pst.setString(2, txt_createbill_custid.getText());
                pst.setString(3, cmbbox_createbill_custnm.getSelectedItem().toString());
                pst.setString(4, txt_createbill_contact.getText());
                pst.setString(5, txt_createbill_addr.getText());
                pst.setString(6, lbl_tax.getText());
                pst.setString(7, lbl_discount.getText());
                pst.setString(8, lbl_grosstotal.getText());
                pst.setString(9, lbl_unpaid.getText());
                pst.setString(10, java.time.LocalDate.now().toString());

                pst.executeUpdate();

                JOptionPane.showMessageDialog(null, "Bill Added Successfully");

                txtarea_format.append("\t\t Billing Application \n"
                    + "\n*****\n"
                    + "*****\n"
                    + " Bill No. :          " + lbl_billno_no.getText() + "\n"
                    + " CUSTOMER NAME :      " +
                    cmbbox_createbill_custnm.getSelectedItem().toString() + "\n"
                    + " CONTACT :           " + txt_createbill_contact.getText() + "\n"
                    + " ADDRESS :           " + txt_createbill_addr.getText() + "\n"
                    + "===== \n"

```

```
+ " Tax :                " + lbl_tax.getText() + "\n"
+ " Discount :          " + lbl_discount.getText() + "\n"
+ " Total Amount :      " + lbl_nettotal.getText() + "\n"
+ "=====\\n"
+ " UNPAID :            " + lbl_unpaid.getText() + "\n"
+ "***** Thank You for Shopping *****\\n");

    txtarea_format.print();
}
catch (Exception e)
{
    JOptionPane.showMessageDialog(null, e.getMessage());
}
}

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

private void cmbbox_createbill_custnmItemStateChanged(java.awt.event.ItemEvent evt)
{
    String custnm = cmbbox_createbill_custnm.getSelectedItem().toString();

    if (custnm.equals("CUSTOMER NAME"))
    {
        txt_createbill_custid.setText("");
        txt_createbill_contact.setText("");
        txt_createbill_addr.setText("");
    }
    else
    {
        try
        {
            query = "select * from customer where custnm = '" + custnm + "'";
            con = dbconnection.getdbConnection();
            st = con.createStatement();
            result = st.executeQuery(query);

            if (result.next())
            {
                txt_createbill_custid.setText(result.getString("id"));
                txt_createbill_contact.setText(result.getString("contact"));
                txt_createbill_addr.setText(result.getString("address"));
            }
        }
        catch (Exception e)
        {
        }
```

```
        JOptionPane.showMessageDialog(null, e.getMessage());
    }
}

private void cmbbox_createbill_prodnmItemStateChanged(java.awt.event.ItemEvent evt)
{
    if (cmbbox_createbill_prodnm.getSelectedItem().equals("PRODUCT NAME"))
    {
        txt_createbill_prodid.setText("");
        txt_createbill_prodprice.setText("");
        spin_createbill_quantity.setValue(1);
    }
    else
    {
        String prodnm = cmbbox_createbill_prodnm.getSelectedItem().toString();

        try
        {
            query = "select * from product where nm like '" + prodnm + "'";
            con = dbconnection.getdbConnection();
            st = con.createStatement();
            result = st.executeQuery(query);

            if (result.next())
            {
                txt_createbill_prodid.setText(result.getString("id"));
                txt_createbill_prodprice.setText(result.getString("sellingprice"));
                txt_createbill_discount.setText("0");
                txt_createbill_tax.setText("0");
                txt_createbill_unpaid.setText("0");
                spin_createbill_quantity.setValue(1);
            }
        }
        catch (Exception e)
        {
            JOptionPane.showMessageDialog(null, e.getMessage());
        }
    }
}

private void txt_addcust_contactKeyTyped(java.awt.event.KeyEvent evt)
{
    isDigit(evt, txt_addcust_contact);
}

private void txt_editcust_contactKeyTyped(java.awt.event.KeyEvent evt)
```

```
{
    isDigit(evt, txt_editcust_contact);
}

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

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

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

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

private void btn_accsetting_updateActionPerformed(java.awt.event.ActionEvent evt)
{
    if (isEmpty(txt_accsetting_nm, txt_accsetting_usrnm, txt_accsetting_contact,
txt_accsetting_email, txt_accsetting_salary, txt_accsetting_address))
    {
        try
        {
            String nm = lbl_dash_usr.getText();
            query = "update employee set nm = ?, usrnm = ?, pswd = ?, contact = ?, email = ?, salary = ?,
address = ? where nm = ?";

            con = dbconnection.getdbConnection();
            pst = con.prepareStatement(query);
            pst.setString(1, txt_accsetting_nm.getText());
            pst.setString(2, txt_accsetting_usrnm.getText());
            pst.setString(3, txt_accsetting_pswd.getLabelText());
            pst.setString(4, txt_accsetting_contact.getText());
            pst.setString(5, txt_accsetting_email.getText());
            pst.setString(6, txt_accsetting_salary.getText());
            pst.setString(7, txt_accsetting_address.getText());
            pst.setString(8, nm);
            pst.executeUpdate();
            JOptionPane.showMessageDialog(null, "Employee Details Updated Successfully");
        }
        catch (Exception e)
```

```
        {
            JOptionPane.showMessageDialog(null, e.getMessage());
        }
    }
}

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

private void txt_createbill_discountFocusLost(java.awt.event.FocusEvent evt)
{
    if (txt_createbill_discount.getText().equals(""))
    {
        txt_createbill_discount.setText("0");
    }
}

private void txt_createbill_taxFocusLost(java.awt.event.FocusEvent evt)
{
    if (txt_createbill_tax.getText().equals(""))
    {
        txt_createbill_tax.setText("0");
    }
}

private void txt_createbill_unpaidFocusLost(java.awt.event.FocusEvent evt)
{
    if (txt_createbill_unpaid.getText().equals(""))
    {
        txt_createbill_unpaid.setText("0");
    }
}

private void cmbbox_createbill_custnm1ItemStateChanged(java.awt.event.ItemEvent evt)
{
    // TODO add your handling code here:
}

private void cmbbox_createbill_custnm1ActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:
}

private void txt_createbill_tax1FocusLost(java.awt.event.FocusEvent evt)
{
    // TODO add your handling code here:
}
```

```
private void txt_createbill_tax1KeyReleased(java.awt.event.KeyEvent evt)
{
    // TODO add your handling code here:
}

private void txt_createbill_discount1FocusLost(java.awt.event.FocusEvent evt)
{
    // TODO add your handling code here:
}

private void txt_createbill_discount1KeyReleased(java.awt.event.KeyEvent evt)
{
    // TODO add your handling code here:
}

private void spin_createbill_quantity1StateChanged(javax.swing.event.ChangeEvent evt)
{
    // TODO add your handling code here:
}

private void cmbbox_createbill_prodnm1ItemStateChanged(java.awt.event.ItemEvent evt)
{
    // TODO add your handling code here:
}

private void cmbbox_createbill_prodnm1ActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:
}

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

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

private void txt_addemp_contactKeyTyped(java.awt.event.KeyEvent evt)
{
    isDigit(evt, txt_addemp_contact);
}

private void txt_addemp_salaryKeyTyped(java.awt.event.KeyEvent evt)
{
    isDigit(evt, txt_addemp_salary);
}
```



```
}

private void txt_editemp_contactKeyTyped(java.awt.event.KeyEvent evt)
{
    isDigit(evt, txt_editemp_contact);
}

private void txt_editemp_salaryKeyTyped(java.awt.event.KeyEvent evt)
{
    isDigit(evt, txt_editemp_salary);
}

private void txt_addprod_costpriceKeyTyped(java.awt.event.KeyEvent evt)
{
    isDigit(evt, txt_addprod_costprice);
}

private void txt_addprod_sellingpriceKeyTyped(java.awt.event.KeyEvent evt)
{
    isDigit(evt, txt_addprod_sellingprice);
}

private void txt_addprod_quantityKeyTyped(java.awt.event.KeyEvent evt)
{
    isDigit(evt, txt_addprod_quantity);
}

private void txt_editprod_sellingpriceKeyTyped(java.awt.event.KeyEvent evt)
{
    isDigit(evt, txt_editprod_sellingprice);
}

private void txt_editprod_costpriceKeyTyped(java.awt.event.KeyEvent evt)
{
    isDigit(evt, txt_editprod_costprice);
}

private void txt_editprod_quantityKeyTyped(java.awt.event.KeyEvent evt)
{
    isDigit(evt, txt_editprod_quantity);
}

private void txt_editprod_idKeyTyped(java.awt.event.KeyEvent evt)
{
    isDigit(evt, txt_editprod_id);
}

private void txt_createbill_discountKeyTyped(java.awt.event.KeyEvent evt)
{

```

```
        isDigit(evt, txt_createbill_discount);
    }

    private void txt_createbill_taxKeyTyped(java.awt.event.KeyEvent evt)
    {
        isDigit(evt, txt_createbill_tax);
    }

    private void txt_createbill_unpaidKeyTyped(java.awt.event.KeyEvent evt)
    {
        isDigit(evt, txt_createbill_unpaid);
    }

    private void txt_editemp_idKeyTyped(java.awt.event.KeyEvent evt)
    {
        isDigit(evt, txt_editemp_id);
    }

    private void txt_editcust_idKeyTyped(java.awt.event.KeyEvent evt)
    {
        isDigit(evt, txt_editcust_id);
    }

    private void txt_addemp_idKeyTyped(java.awt.event.KeyEvent evt)
    {
        isDigit(evt, txt_addemp_id);
    }

    private void txt_addprod_idKeyTyped(java.awt.event.KeyEvent evt)
    {
        isDigit(evt, txt_addprod_id);
    }

    private void txt_accsetting_contactKeyTyped(java.awt.event.KeyEvent evt)
    {
        isDigit(evt, txt_accsetting_contact);
    }

    private void txt_accsetting_salaryKeyTyped(java.awt.event.KeyEvent evt)
    {
        isDigit(evt, txt_accsetting_salary);
    }

    public static void main(String args[])
    {

        java.awt.EventQueue.invokeLater(new Runnable()
        {
            public void run()
```

```
        {
            new Main().setVisible(true);
        }
    });
}

public void hover(JPanel panel, Color color)
{
    panel.setBackground(color);
}

public void hover(KButton button, Color color, Color bg)
{
    button.setkStartColor(color);
    button.setkEndColor(color);
    button.setBackground(bg);
}

public void visibility(boolean pnl_home, boolean pnl1, boolean pnl2, boolean pnl3, boolean pnl4,
boolean pnl5, KButton btn, int keycode)
{
    if (keycode == KeyEvent.VK_ENTER)
    {
        panel_home.setVisible(pnl_home);
        panel_customer.setVisible(pnl1);
        panel_employee.setVisible(pnl2);
        panel_product.setVisible(pnl3);
        panel_bill.setVisible(pnl5);
        panel_account.setVisible(pnl4);
        btn.grabFocus();
    }
}

public void visibility(boolean pnl_home, boolean pnl1, boolean pnl2, boolean pnl3, boolean pnl4,
boolean pnl5, KButton btn)
{
    panel_home.setVisible(pnl_home);
    panel_customer.setVisible(pnl1);
    panel_employee.setVisible(pnl2);
    panel_product.setVisible(pnl3);
    panel_bill.setVisible(pnl4);
    panel_account.setVisible(pnl5);
    btn.grabFocus();
}

public void visibility(KGradientPanel tabtoshow, KGradientPanel tab2, KGradientPanel tab3)
{
    tabtoshow.setVisible(true);
    tab2.setVisible(false);
}
```

```
        tab3.setVisible(false);
    }

    public void visibility(KGradientPanel tab1, boolean tab1bool, KGradientPanel tab2, boolean
tab2bool, KGradientPanel tab3, boolean tab3bool)
    {
        tab1.setVisible(tab1bool);
        tab2.setVisible(tab2bool);
        tab3.setVisible(tab3bool);
    }

    public void onIndicator(KGradientPanel lid1, KGradientPanel lid2, KGradientPanel lid3)
    {
        lid1.setkStartColor(new Color(1, 109, 218));
        lid1.setkEndColor(new Color(1, 109, 218));
        lid1.setBackground(Color.white);

        lid2.setkStartColor(new Color(211, 211, 211));
        lid2.setkEndColor(new Color(211, 211, 211));
        lid2.setBackground(Color.black);

        lid3.setkStartColor(new Color(211, 211, 211));
        lid3.setkEndColor(new Color(211, 211, 211));
        lid3.setBackground(Color.black);
    }

    public void onIndicator(KGradientPanel lid1, KGradientPanel lid2)
    {
        lid1.setkStartColor(new Color(1, 109, 218));
        lid1.setkEndColor(new Color(1, 109, 218));
        lid1.setBackground(Color.white);

        lid2.setkStartColor(new Color(211, 211, 211));
        lid2.setkEndColor(new Color(211, 211, 211));
        lid2.setBackground(Color.black);
    }

    public void offIndicator(KGradientPanel lid1, Color color1, KGradientPanel lid2, Color color2)
    {
        lid1.setkStartColor(color1);
        lid1.setkEndColor(color1);
        lid1.setBackground(Color.black);

        lid2.setkStartColor(color2);
        lid2.setkEndColor(color2);
        lid2.setBackground(Color.black);
    }

    public String getMaxID(String tablenm)
```

```

{
    String id;
    try
    {
        con = dbconnection.getdbConnection();
        query = "select max(id) from " + tablename;
        pst = con.prepareStatement(query);
        result = pst.executeQuery();

        if (result.next())
        {
            if (Integer.parseInt(result.getString(1)) > 0)
            {
                return Integer.toString(Integer.parseInt(result.getString(1)) + 1);
            }
        }
    }

    catch (NumberFormatException e)
    {
        /*
            If no ID is found by ResultSet, will cause NumberFormatException
            and needed to keep catch block blank to avoid intereption.
        */
    }
    catch (Exception e)
    {
        JOptionPane.showMessageDialog(null, e);
    }
    return "1";
}

public void bindCustomerTableData(JTable table)
{
    try
    {
        tableModel = (DefaultTableModel) table.getModel();
        tableModel.setRowCount(0);

        con = dbconnection.getdbConnection();
        query = "select * from customer";
        pst = con.prepareStatement(query);

        result = pst.executeQuery();

        String id, nm, contact, email, shpnm, addr;
        while (result.next())
        {
            id = result.getString(1);

```

```
        nm = result.getString(2);
        contact = result.getString(3);
        email = result.getString(4);
        shpnm = result.getString(5);
        addr = result.getString(6);

        String[] rows =
        {
            id, nm, contact, email, shpnm, addr
        };

        tableModel.addRow(rows);
    }
}
catch (Exception e)
{
    JOptionPane.showMessageDialog(null, e.getMessage());
}
}

public void bindEmployeeTableData(JTable table)
{
    try
    {
        tableModel = (DefaultTableModel) table.getModel();
        tableModel.setRowCount(0);

        con = dbconnection.getdbConnection();
        query = "select * from employee";
        pst = con.prepareStatement(query);

        result = pst.executeQuery();

        String id, nm, usnm, pswd, contact, email, salary, addr;
        while (result.next())
        {
            id = result.getString(1);
            nm = result.getString(2);
            usnm = result.getString(3);
            pswd = result.getString(4);
            contact = result.getString(5);
            email = result.getString(6);
            salary = result.getString(7);
            addr = result.getString(8);

            String[] rows =
            {
                id, nm, usnm, pswd, contact, email, salary, addr
            };
        }
    }
}
```

```

        tableModel.addRow(rows);
    }
}
catch (Exception e)
{
    JOptionPane.showMessageDialog(null, e.getMessage());
}
}

public void bindProductTableData(JTable table)
{
    try
    {
        tableModel = (DefaultTableModel) table.getModel();
        tableModel.setRowCount(0);

        con = dbconnection.getdbConnection();
        query = "select * from product";
        pst = con.prepareStatement(query);

        result = pst.executeQuery();

        String id, nm, costprice, sellingprice, brand, quantity;
        while (result.next())
        {
            id = result.getString(1);
            nm = result.getString(2);
            costprice = result.getString(3);
            sellingprice = result.getString(4);
            brand = result.getString(5);
            quantity = result.getString(6);

            String[] rows =
            {
                id, nm, costprice, sellingprice, brand, quantity
            };

            tableModel.addRow(rows);
        }
    }
    catch (Exception e)
    {
        JOptionPane.showMessageDialog(null, e.getMessage());
    }
}

public boolean isIdExist(String txtid, String tblnm)
{

```

```
try
{
    con = dbconnection.getdbConnection();
    query = "select id from " + tblnm + " where id = ?";

    pst = con.prepareStatement(query);
    pst.setString(1, txtid);

    result = pst.executeQuery();

    if (result.next())
    {
        return true;
    }
    else
    {
        return false;
    }
}
catch (Exception e)
{
    JOptionPane.showMessageDialog(null, e.getMessage());
    return false;
}
}

public void clearTextFields(TextField txt1, TextField txt2, TextField txt3, TextField txt4, TextField
txt5, TextField txt6)
{
    txt1.setText("");
    txt2.setText("");
    txt3.setText("");
    txt4.setText("");
    txt5.setText("");
    txt6.setText("");
}

public void clearTextFields(TextField txt1, TextField txt2, TextField txt3, TextField txt4, TextField
txt5, TextField txt6, TextField txt7)
{
    txt1.setText("");
    txt2.setText("");
    txt3.setText("");
    txt4.setText("");
    txt5.setText("");
    txt6.setText("");
    txt7.setText("");
}
```



```
public void clearTextFields(TextField txt1, TextField txt2, TextField txt3, PasswordField txt4,
TextField txt5, TextField txt6, TextField txt7, TextField txt8)
```

```
{
    txt1.setText("");
    txt2.setText("");
    txt3.setText("");
    txt4.setText("");
    txt5.setText("");
    txt6.setText("");
    txt7.setText("");
    txt8.setText("");
}
```

```
public void clearTextFields(TextField txt1, TextField txt2, TextField txt3, TextField txt4,
ComboBoxSuggestion cmbbox, TextField txt6)
```

```
{
    txt1.setText("");
    txt2.setText("");
    txt3.setText("");
    txt4.setText("");
    cmbbox.setSelectedIndex(0);
    txt6.setText("");
}
```

```
public void fillCombobox(String tablename, String columnname, ComboBoxSuggestion cmbbox)
```

```
{
    try
    {
        con = dbconnection.getdbConnection();
        query = "select * from " + tablename;
        st = con.createStatement();
        result = st.executeQuery(query);
        while (result.next())
        {
            cmbbox.addItem(result.getString(columnname));
        }
    }
    catch (Exception e)
    {
        JOptionPane.showMessageDialog(null, e.getMessage());
    }
}
```

```
public boolean isFieldEmpty(TextField txt1, TextField txt2, TextField txt3, TextField txt4, TextField
txt5, TextField txt6)
```

```
{
    if (txt1.getText().equals(""))
    {
```

```
        txt1.grabFocus();
        return false;
    }
    else if (txt2.getText().equals(""))
    {
        txt2.grabFocus();
        return false;
    }
    else if (txt3.getText().equals(""))
    {
        txt3.grabFocus();
        return false;
    }
    else if (txt4.getText().equals(""))
    {
        txt4.grabFocus();
        return false;
    }
    else if (txt5.getText().equals(""))
    {
        txt5.grabFocus();
        return false;
    }
    else if (txt6.getText().equals(""))
    {
        txt6.grabFocus();
        return false;
    }
    else
    {
        return true;
    }
}
```

```
public boolean isEmpty(TextField txt1, TextField txt2, TextField txt3, TextField txt4, TextField
txt5, TextField txt6, TextField txt7)
{
    if (txt1.getText().equals(""))
    {
        txt1.grabFocus();
        return false;
    }
    else if (txt2.getText().equals(""))
    {
        txt2.grabFocus();
        return false;
    }
}
```

```

    }
    else if (txt3.getText().equals(""))
    {
        txt3.grabFocus();
        return false;

    }
    else if (txt4.getText().equals(""))
    {
        txt4.grabFocus();
        return false;

    }
    else if (txt5.getText().equals(""))
    {
        txt5.grabFocus();
        return false;

    }
    else if (txt6.getText().equals(""))
    {
        txt6.grabFocus();
        return false;

    }
    else if (txt7.getText().equals(""))
    {
        txt7.grabFocus();
        return false;

    }
    else
    {
        return true;
    }
}

```

```

public boolean isEmpty(TextField txt1, TextField txt2, TextField txt3, TextField txt4, TextField
txt5)
{
    if (txt1.getText().equals(""))
    {
        txt1.grabFocus();
        return false;
    }
    else if (txt2.getText().equals(""))
    {
        txt2.grabFocus();
        return false;
    }
    else if (txt3.getText().equals(""))

```

```
{
    txt3.grabFocus();
    return false;

}
else if (txt4.getText().equals(""))
{
    txt4.grabFocus();
    return false;

}
else if (txt5.getText().equals(""))
{
    txt5.grabFocus();
    return false;

}
else
{
    return true;
}
}
```

```
public void isDigit(KeyEvent evt, TextField txt)
{
    if (evt.getKeyChar() < '0' || evt.getKeyChar() > '9' || txt.getText().length() == 10)
    {
        evt.consume();
    }
}
```

```
public boolean isCmbboxDefaultSelected(String str, combo_suggestion.ComboBoxSuggestion
cmbbox)
{
    if (cmbbox.getSelectedIndex() == 0)
    {
        return true;
    }
    else
    {
        return false;
    }
}
```

Limitation

- The following are the limitations of the project:
- This application supports only on **1920 x 1080** resolution devices.
- Any computer can crash or become affected by a virus at any time. So, it is imperative to take the backup of the data, but this Application does not provide this utility.
- No email facility available.
- Application doesn't provide any images like product image, employee image etc.

Conclusion

- It also maintains records of various products, clients and Admins.
- In fine we want to say we are not yet an experienced system analyst, but tried our best to fulfill this project.

Bibliography

www.youtube.com

www.google.com

9.1 References

We are really thankful to our guider Mr. Savan Raval to guide us and inspire us. We also Thankful to the whole staff of computer Department to gives us a huge support in our project.

Sites:

- www.stackoverflow.com
- www.tutorialspoint.com